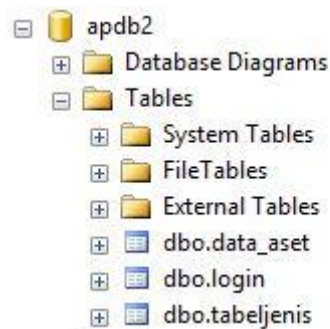


BAB IV HASIL DAN PEMBAHASAN

4.1 Pembuatan Basis Data

Cara kerja aplikasi data aset Angkasa Pura I Yogyakarta adalah dengan memasukkan data kedalam *database* yang sudah terhubung sebelumnya. *Database* yang digunakan adalah perangkat lunak *SQL Server Management Studio 2016*. Gambar 4.1 menjelaskan tentang *database* yang dipakai pada aplikasi data aset.



Gambar 4. 1 Pembuatan Basis Data

4.1.1 Definisi Basis Data

Berikut adalah tabel-tabel *database* aplikasi *desktop* data aset Angkasa Pura I Yogyakarta.

Tabel 4. 1 Basis Data Admin

Nama Kolom	Tipe Data	Null	Panjang Data	Keterangan
<i>Username</i>	Varchar	No	50	<i>Primary Key</i>
<i>Password</i>	Varchar	No	50	-

Tabel 4.1 menggambarkan tabel *login* admin. Dimana admin hanya memasukkan *username* dan *password* yang benar, maka sistem otomatis akan masuk kedalam tampilan selanjutnya. Dengan begitu, admin dapat menggunakan aplikasi sesuai fungsi.

Tabel 4. 2 Data Aset

Nama Kolom	Tipe Data	Null	Panjang Data	Keterangan
Nomor	Varchar	No	50	<i>Primary key</i>
Nama Peralatan	Varchar	No	50	-
Jenis	Varchar	No	50	-
Serial Number	Integer	No	-	-
Pabrik Pembuat	Varchar	No	50	-
Merk dan Tipe	Varchar	No	50	-
Tahun	Varchar	No	50	-
Daya	Varchar	No	50	-
Kondisi	Varchar	No	50	-
Penempatan	Varchar	No	50	-
Status	Varchar	No	50	-
Keterangan	Varchar	No	50	-

Tabel 4.2 menunjukkan tabel data aset. Tabel ini berfungsi untuk menyimpan data-data yang ada di Angkasa Pura I Yogyakarta. Tabel ini hanya bisa diisi oleh admin data aset, karena harus melalui proses *login* terlebih dahulu yang dimaksudkan untuk menghindari penyalahgunaan data yang akan tersimpan didalam tabel *database* aset.

Tabel 4. 3 Data Jenis

Nama Kolom	Tipe Data	Null	Panjang Data	Keterangan
Id	Integer	No	-	-
Jenis	Varchar	No	50	-

Tabel 4.3 menggambarkan tabel jenis. Dimana data jenis dengan kolom id dan jenis dimasukkan untuk menyimpan data yang akan ditampilkan didalam data aset.

4.2 Implementasi GUI (Graphic User Interface)

Rancangan antar muka ini diubah kedalam bahasa pemrograman terlebih dahulu sebelum implementasi *user interface*. *Software* yang digunakan dalam melakukan pemrograman dengan bahasa *Java* adalah *NetBeans IDE 8.2*.

4.2.1 Halaman *Login*

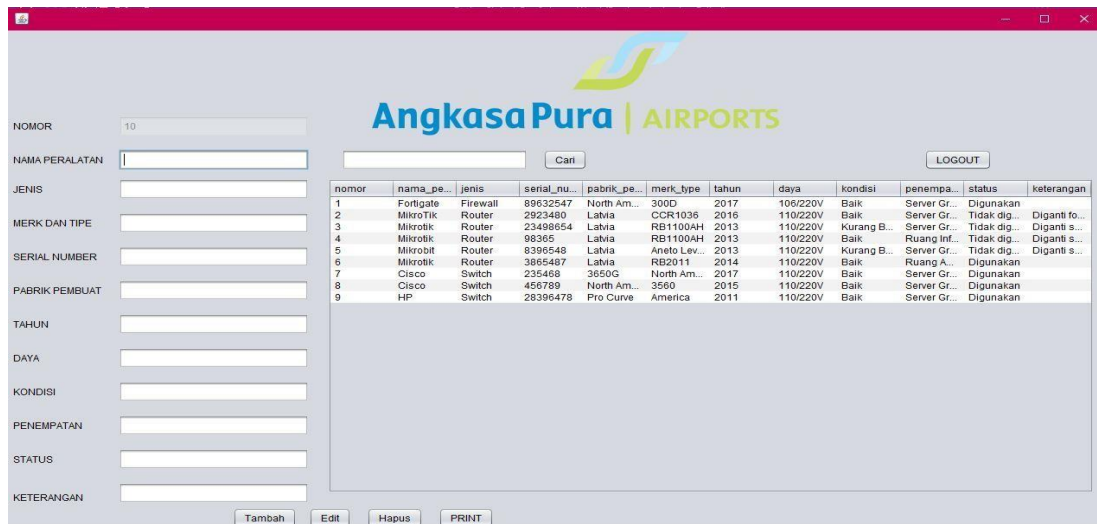
Dalam mengoperasikan sistem data aset ini, admin harus melakukan proses *login* terlebih dahulu dengan cara memasukkan *username* dan *password* pada *textbox* yang sudah disediakan dengan benar seperti pada gambar 4.2.



Gambar 4. 2 Tampilan Login

4.2.2 Halaman Data Aset

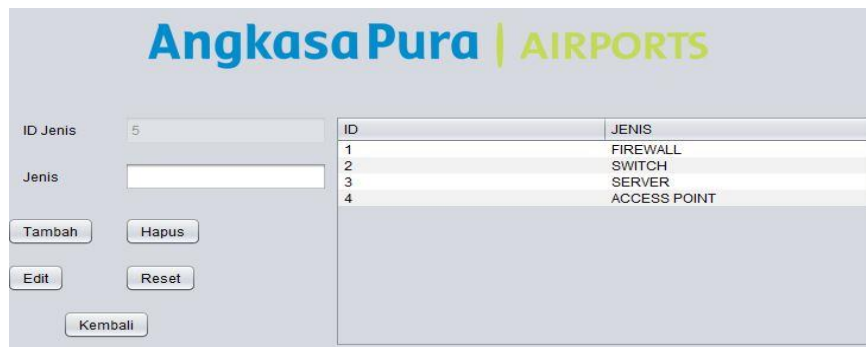
Setelah admin berhasil *login*, halaman otomatis menuju ke tampilan data aset yang berisi daftar *textbox* untuk data yang akan dimasukkan kedalam tabel data aset. Admin dapat memasukkan satu persatu data kedalam *textbox* sesuai nama data yang tertera untuk kemudian dimasukkan kedalam tabel, seperti yang terlihat pada gambar 4.3.



Gambar 4. 3 Halaman Data Aset

4.2.3 Halaman Tabel Jenis

Halaman selanjutnya adalah halaman jenis, dimana ketika sistem butuh tambaha jenis pada halaman data aset, maka admin bisa masuk kedalam tabel jenis untuk menambah jenis yang sebelumnya belum terdata. Seperti yang terlihat pada gambar 4.4.



Gambar 4. 4 Halaman Data Aset

4.3 Pengujian

4.3.1 Pengujian Fungsional

Dalam tahapan ini, hasil dari pembangunan sistem diuji terlebih dahulu berdasarkan rancangan dan kebutuhan yang ada pada tahapan-tahapan sebelumnya guna memastikan apakah sudah sesuai dengan analisa kebutuhan. Dengan tujuan untuk memastikan bahwa aplikasi sudah berjalan sesuai fungsi atau sesuai keinginan serta berjalan dengan baik tanpa teridentifikasi *error* atau *bug*.

Hasil pengujian ini terlihat pada tabel 4.3.

Tabel 4. 4 Tabel Pengujian Fungsional

No	Kasus Uji	Hasil Yang Diharapkan	Hasil Yang Didapat	Status
1	Tampilan <i>login</i>	Admin berhasil masuk ke aplikasi	Admin berhasil masuk dan form <i>login</i> terlihat.	Berhasil
2	<i>Button login</i> dengan <i>username</i> dan <i>password</i>	Keluar notifikasi	Adanya notifikasi bahwa <i>username</i> dan <i>password</i> salah.	Berhasil
3	<i>Button Log out</i>	Admin kembali ke menu utama (<i>login</i>)	Tampilan kembali ke menu <i>login</i> dan dapat <i>login</i> ulang.	Berhasil
4	<i>Button</i> tambah	Menambah data	Data masuk ke <i>table database</i> .	Berhasil.
5	<i>Button</i> edit	Edit data yang sudah tersedia	Data yang tampil adalah data yang sudah di edit.	Berhasil.

No	Kasus Uji	Hasil Yang Diharapkan	Hasil Yang Didapat	Status
6	<i>Button hapus</i>	Hapus data yang ada.	Data yang sebelumnya ada sudah tidak terdaftar dalam <i>table database</i> data aset.	Berhasil.
7	<i>Textbox Search</i>	Menemukan data	Tabel akan menampilkan data yang di <i>search</i> .	Berhasil.
8	Fungsi <i>button print</i>	Mencetak data	Data tercetak sesuai dengan isi <i>table</i> .	Berhasil.

4.4 Pembahasan

Aplikasi data aset Angkasa Pura hanya dapat di akses oleh admin. Dalam analisa kebutuhannya, jalan keluar yang ditawarkan dengan mengadakan fitur *login* pada sistem. *Login* memiliki proses memasukkan *username* dan *password* yang sesuai, maka sebagai otoritas, hanya admin yang berhak mengetahui *username* dan *password* yang sesuai. Tentu dengan memberikan hak akses sepenuhnya kepada admin, maka kekeliruan dalam pemalsuan data terhindari. Berdasarkan tabel 4.4 no. 1, admin berhasil *login* dan masuk kedalam aplikasi, dan no. 2, ketika login di masukkan dengan *username* dan *password* yang tidak sesuai maka akan keluar notifikasi bahwa *username* atau *password* yang dimasukkan salah.

Terdapat kekeliruan saat mengelola data. Seperti yang umum ditemukan adalah kesalahan penulisan, kertas yang hilang, kertas terselip atau tumpahnya cairan . Hal ini menjadi pemicu untuk menambahkan 3 tombol yang berfungsi untuk mengelola data. Pertama adalah tombol tambah. Tombol tambah digunakan oleh admin sebagai sarana untuk menambahkan data yang belum terdaftar, dengan cara memasukkan satu persatu data yang

sesuai kedalam *textbox* yang sudah disediakan. Seperti tercantum pada tabel 4.4 no. 4 tentang *button* tambah yang berhasil menambah data masuk ke dalam tabel *database*.

Kedua adalah tombol edit, karena bisa saja sewaktu-waktu data yang dimasukkan kurang tepat, maka proses edit data sangat berfungsi. Dengan sekali klik pada data yang ingin di edit pada tabel, secara otomatis data akan masuk kedalam *textbox*. Kemudian admin bisa langsung memperbaiki. Setelah itu klik tombol edit dan data akan tampil pada tabel. Terlihat pada tabel 4.4 no. 5 tentang *button* edit, berhasil mengedit data.

Yang ketiga adalah tombol hapus, dimana tombol hapus ini berfungsi untuk menghapus data yang salah dimasukkan atau asetnya sudah tidak ada lagi. Caranya dengan klik pada data apa yang akan dihapus, maka akan keluar pemberitahuan bahwa data akan dihapus, kemudian ikuti langkah selanjutnya agar data berhasil dihapus. Dalam ketiga proses tersebut, tentu terjadi proses pencarian data, maka untuk memudahkan pencarian data, maka disediakan tombol *search* pada tampilan sistem. Seperti yang terlihat pada tabel 4.4 no. 6 dimana data yang sebelumnya ada, sudah berhasil dihapus.

Dalam proses pendataan aset yang manual, admin yang bertugas dalam mengelola data aset harus membuat arsip data. Logikanya, admin harus bekerja dua kali untuk menduplikat data yang tujuannya adalah untuk arsip dan pegangannya sebagai bahan pertanggungjawaban. Sehingga dalam aplikasi ini terdapat fitur *print* atau cetak data. Agar meringankan kerja admin yang harus mendata dua kali. Dengan adanya fitur *print*, maka data yang tersedia pada sistem bisa dicetak demi keperluan arsip data. Seperti yang tercantum pada tabel 4.4 no. 8 dalam fungsi *button print* yang berhasil mencetak data.

Dari semua filter yang ditawarkan, pengujian membuktikan bahwa semua fitur bekerja dengan semestinya.