BAB V IMPLEMENTASI

Bab ini membahas mengenai tahapan implementasi aplikasi Profil SMAN 2 Malang berdasarkan hasil yang telah didapatkan dari analisis dan perancangan. Pembahasan terdiri atas penjelasan tentang batasan - batasan dalam implementasi, implementasi basis data, implementasi tiap *class* pada *file* program, implementasi algoritma dan implementasi antarmuka.

5.1 Deskripsi Aplikasi

Pembuatan aplikasi ini didukung beberapa *software*, diantaranya Adobe Director, Adobe Flash, Adobe Photoshop, 3D Studio Max 6, dan Adobe After Effects.

Secara garis besar aplikasi ini mendeskripsikan profil SMAN 2 Malang, kemudian dimuat dalam bentuk *desktop*. Sehingga dengan melihat aplikasi ini pengunjung akan mendapatkan gambaran tentang SMAN 2 Malang.

5.2 Spesifikasi Kebutuhan *Hardware* dan *Software*

Dalam pembuatan profil SMAN 2 Malang ini, diperlukan perangkat lunak yang digunakan untuk desain tampilan dan *coding* program. Selain itu diperlukan juga piranti perangkat keras dimana sebagai media dalam pembuatan profil ini.

Berikut ini adalah beberapa *software* yang kami gunakan :

1. Sistem Operasi yang digunakan adalah Windows XP.
2. Adobe Director, digunakan untuk mendesain tampilan dan animasi dari aplikasi ini.
3. Adobe Photoshop, digunakan untuk editing foto atau gambar.
4. Adobe Flash, untuk desain animasi.
5. 3D Studio Max 6, digunakan untuk desain gambar 3D.

Dan untuk hardware :

1. Komputer PC 2.80 Hz
2. Monitor 14”
3. RAM min 1Gb
4. HDD min 40 Gb

5.3 Batasan –Batasan Implementasi

Beberapa batasan dalam mengimplementasikan perangkat lunak *Your Place Messenger* adalah sebagai berikut:

1. Perangkat lunak yang digunakan adalah *Adobe Director 11*.
2. Bahasa pemrograman yang digunakan adalah *Lingo*.
3. Manajemen basis data yang digunakan adalah *Arca Database Xtras*.

5.4 Implementasi Basis Data

Implementasi penyimpanan data dilakukan dengan *Arca Database Xtras*. Hasil implementasi penyimpanan data ini berupa *file .dat* dimana sebelumnya diletakkan *global script* pada *cast member* halaman yang membutuhkan olah data seperti *insert, update, maupun delete*. *Arca Database Xtras* adalah *plugin / xtras* yang telah disediakan oleh pihak *Adobe* yang bisa digunakan untuk memanggil fungsi-fungsi *database*.

5.5 Implementasi Menu dan Interface Pada File Program

Setiap menu yang telah dirancang pada proses perancangan direalisasikan pada sebuah *file* program dengan ekstensi **.dir*. Tabel 5.1 menjelaskan mengenai *file* program dan *interface* yang terkandung didalamnya yang digunakan untuk mengimplementasikannya.

Tabel 5.1 Implementasi *interface* pada kode program **.dir*

No.	Interface Terkandung	Nama File Program
1	<ul style="list-style-type: none"> • <i>Login</i> • <i>Insert, Update, Delete</i> 	home.dir
2	Halaman Denah 3Dimensi	3d.dir

	• <i>Walkthrough</i>	
3	Halaman Kontak	about2.dir
4	Halaman Album Foto • <i>Insert, Update, Delete</i>	album2.dir
5	Halaman Menu Album	album3.dir
6	Halaman Menu Denah	denah2.dir
7	Halaman Denah 2Dimensi • <i>Insert, Update, Delete</i>	denah.dir
8	Halaman Foto-foto Ekstrakurikuler • <i>Insert, Update, Delete</i>	ekstra1.dir
9	Halaman Menu Ekstrakurikuler	ekstra2.dir
10	Halaman Guru Agama dan Komputer • <i>Insert, Update, Delete</i>	g_agkom.dir
11	Halaman Profil Guru Bahasa • <i>Insert, Update, Delete</i>	g_bahasa.dir
12	Halaman Profil Guru Bimbingan Konseling, PKn, dan Olahraga • <i>Insert, Update, Delete</i>	g_bk.dir
13	Halaman Profil Guru IPS dan Kesenian • <i>Insert, Update, Delete</i>	g_ips.dir
14	Halaman Profil Guru Matematika dan IPA • <i>Insert, Update, Delete</i>	g_mipa.dir
15	Halaman Profil Karyawan	g_karyawan.dir

	• <i>Insert, Update, Delete</i>	
16	Halaman Menu Guru	guru3.dir
17	Halaman Menu Kegiatan	kegiatan2.dir
18	Halaman Struktur Aplikasi	map.dir
19	Halaman Mars dan Hymne SMAN 2 Malang	mars.dir
20	Halaman Daftar Prestasi	prestasi2.dir
21	Halaman Menu Profil Sekolah	profil_sekolah2.dir
22	Halaman Slogan <i>Studium Pro Patria</i> • <i>Update</i>	profil_studium.dir
23	Halaman Sejarah SMAN 2 Malang • <i>Update</i>	sejarah.dir
24	Halaman Kejuaraan Basket SMP	smandacup.dir
25	Halaman Struktur Organisasi • <i>Update</i>	struktur.dir
26	Halaman Visi dan Misi • <i>Update</i>	visi.dir

5.6 Implementasi Algoritma

Aplikasi Profil SMAN 2 Malang memiliki beberapa *method* yang ada pada beberapa *interface*. *Method* yang akan dijelaskan pada bagian ini hanya untuk algoritma dari beberapa proses saja sehingga tidak semua *method* akan dicantumkan. Implementasi algoritma ini akan direpresentasikan dalam bentuk *pseudocode* dari algoritma – algoritma tersebut.

5.6.1 Implementasi Algoritma untuk *Login*

Proses login berfungsi untuk melakukan seleksi kondisi pada waktu *user* melakukan proses *login* dengan menekan tombol *Login* yang ada pada Halaman *Home*. Implementasi algoritma untuk melakukan seleksi kondisi *login* dijelaskan pada Gambar 5.1.

Nama Algoritma : login

Deklarasi

- String → password

Deskripsi

- Masukan : password
- Proses
 1. Menekan tombol login pada halaman Home
 2. Memasukkan password pada field 'pass'
 3. Jika salah, maka tampilkan pesan "Password Salah!", set field = 0
 4. Jika benar, set field = 1
go adminhome of movie Home
 5. Menutup halaman home user
- Keluaran : halaman home admin

Gambar 5.1 Implementasi algoritma *Login*

5.6.2 Implementasi Algoritma untuk *BukaData*

Proses *BukaData* berfungsi untuk membuka data yang sudah atau akan dimasukkan ke dalam *database* yang akan dibuat. Implementasi algoritma untuk melakukan *BukaData* dijelaskan pada Gambar 5.2.

Nama Algoritma : bukadata

- Proses
 1. Buka file .dat dengan kata kuncinya adalah "katakunci"

2. Memeriksa apakah terjadi error karena file tidak ditemukan
 3. Jika ya, maka panggil fungsi **buatdatabase**
 4. Memeriksa apakah terjadi error karena Arca xtras tidak ditemukan.
 5. Setelah file database dibuat, tampilkan semua record.
- Keluaran : -

Gambar 5.2 Implementasi Algoritma BukaData

5.6.3 Implementasi Algoritma untuk BuatDatabase

Proses BuatDatabase berfungsi untuk membuat sebuah *file* berekstensi .dat yang berfungsi untuk menyimpan data masukan dari *stage* dan *xtras*. Implementasi algoritma untuk melakukan seleksi kondisi BuatDatabase dijelaskan pada Gambar 5.3.

Nama Algoritma : buatdatabase

- Proses
 1. Memeriksa apakah file data bisa dibuat?
Jika tidak maka tampilkan pesan "database tidak dapat dibuat"
 2. Membuat struktur tabel dengan nama Datamotive
 3. Mengeset nama dan field datamotive adalah:
id(INTEGER PRIMARY KEY), nomor(INTEGER),
nama(TEXT), dan gambar(BLOB).
- Keluaran : -

Gambar 5.3 Implementasi Algoritma BuatDatabase

5.6.4 Implementasi Algoritma untuk TambahRecordBaru

Proses TambahRecordBaru berfungsi untuk melakukan penambahan data baru ke dalam *file* .dat yang telah dibuat sebelumnya. Implementasi algoritma untuk melakukan TambahRecordBaru dijelaskan pada Gambar 5.4.

Nama Algoritma : tambahrecordbaru

Deklarasi

- String → id,nomor,nama,gambar

Deskripsi

- Masukan : id,nomor,nama,gambar
- Proses
 1. Menambahkan ruang record baru ke tabel **datamotive** dengan nilai masing-masing field adalah kosong (NULL).
 2. Memberi nilai pada variabel **id** dengan nomor posisi record terakhir.
 3. Menyeleksi record dari tabel **datamotive** berdasarkan **id**.
- Keluaran : record lama + record baru ditambahkan

Gambar 5.4 Implementasi Algoritma TambahRecordBaru

5.6.5 Implementasi Algoritma untuk SimpanPerubahanRecord

Proses SimpanPerubahanBaru berfungsi untuk melakukan penyimpanan data baru ke dalam *file* .dat yang telah dimasukkan dalam proses TambahRecordBaru. Implementasi algoritma untuk melakukan SimpanPerubahanRecord dijelaskan pada Gambar 5.5.

Nama Algoritma : simpanperubahanecord

Deklarasi

- String → id,nomor,nama,gambar

Deskripsi

- Masukan : id,nomor,nama,gambar

- Proses
 1. Memberi nilai pada variabel **id** dengan posisi record yang sedang diseleksi.
 2. Mengisi label datamotive pada
 3. masing-masing field dengan data yang terdapat pada obyek Filled dan Image.
 4. Menyeleksi semua record
- Keluaran : record lama + record baru ditambahkan

Gambar 5.5 Implementasi Algoritma SimpanPerubahanRecord

5.6.6 Implementasi Algoritma untuk HapusRecord

Proses HapusRecord berfungsi untuk melakukan penghapusan data baru di dalam *file* .dat yang telah disimpan sebelumnya. Implementasi algoritma untuk melakukan HapusRecord dijelaskan pada Gambar 5.6.

Nama Algoritma : hapusrecord

- Proses
 1. Memberi nilai pada variabel **id** dengan posisi record yang sedang diseleksi.
 2. Menghapus record dari tabel datamotive pada posisi record yang sesuai dengan nilai **id**.
 3. Tampilkan semua record.
- Keluaran : record setelah dilakukan penghapusan

Gambar 5.6 Implementasi Algoritma HapusRecord