

2. LANDASAN TEORI

2.1. Macromedia Flash MX

Macromedia Flash merupakan standar profesional yang digunakan untuk membuat web lebih hidup dengan permainan animasi dan sinkronisasi suara yang mengagumkan. Bayangkan pada halaman web, dapat dibuat suatu animasi ataupun promosi suatu produk seperti iklan di *layer* televisi. Menampilkan informasi dengan permainan efek suara dan gerakan yang menarik dapat membuat orang lebih terfokus pada pesan yang akan disampaikan melalui Macromedia Flash. Kreatifitaslah yang paling utama untuk membuat situs menjadi menarik di mata pengunjung.

2.1.1. Konsep Dasar Dan Istilah

2.1.1.1. *Stage*

Stage pada lingkungan Flash terlihat berupa kotak putih besar. Semua *object* seperti gambar, tulisan, dan foto ditempatkan pada *stage*. Diluar area kotak putih tersebut, merupakan daerah yang berwarna abu-abu. Pada daerah yang berwarna abu-abu ini penonton tidak dapat melihat obyek apapun pada saat film dimainkan.

2.1.1.2. *Toolbox*

Toolbox berisi alat-alat kerja. Isi dari *toolbox* adalah *arrow tool*, *subselection tool*, *line tool*, *lasso tool*, *pen tool*, *text tool*, *oval tool*, *rectangle tool*, *pencil tool*, *brush tool*, *free transform tool*, *fill transform tool*, *ink bottle*, *paint bucket*, *eyedropper*, *eraser*, *hand*, *zoom*, *stroke color*, dan *fill color*. Berbagai simbol yang ada disana mempunyai fungsi-fungsi tersendiri.

2.1.1.3. *Timeline*

Semua alur cerita diatur disini, kapan *object* harus muncul dan hilang, kapan suara harus mulai dan berhenti.

2.1.1.4. *Frame*

Frame adalah sebuah gambar dari kumpulan banyak gambar. Jika *frame* ini kita tampilkan satu persatu secara berurutan di layar dengan kecepatan tertentu, maka kita akan melihat gambar yang bergerak. Itulah konsep animasi.

2.1.1.5. *Framerate*

Framerate adalah ukuran kecepatan *frame* yang ditampilkan untuk *user* dalam melihat gambar-gambar yang bergerak. Ukurannya adalah fps (*frame per second*). Pada lingkungan standar Flash, ukuran *framerate*-nya 12 fps. Ukuran ini dapat diubah sesuai dengan kebutuhan

2.1.1.6. *Scene*

Jika *movie* mempunyai jalan cerita yang panjang, tentunya perlu dibagi menjadi beberapa tema. Untuk itu perlu dibuat *scene* baru untuk tiap tema yang berbeda.

2.1.1.7. *Layer*

Layer dapat dibayangkan seperti gambaran kertas yang transparan yang berguna untuk membantu proses animasi dan penggambaran.

2.1.2. Cara Pembuatan Animasi

Ada dua cara untuk menganimasikan object pada Flash. Pertama dengan *frame by-frame animation* dan yang satu lagi dengan menggunakan *tweened animation*.

2.1.2.1. *Frame-by-frame animation*

Frame-by-frame animation cara kerjanya adalah dengan menggambar satu per satu gambar pada *frame* dan setelah semuanya selesai digambar baru dianimasikan. Biasanya *frame-by-frame animation* ini digunakan untuk membuat animasi yang rumit. Kelemahan *frame-by-frame animation* adalah ukuran file jauh lebih besar dibandingkan dengan *tweened animation*.

2.1.2.2. Tweened Animation

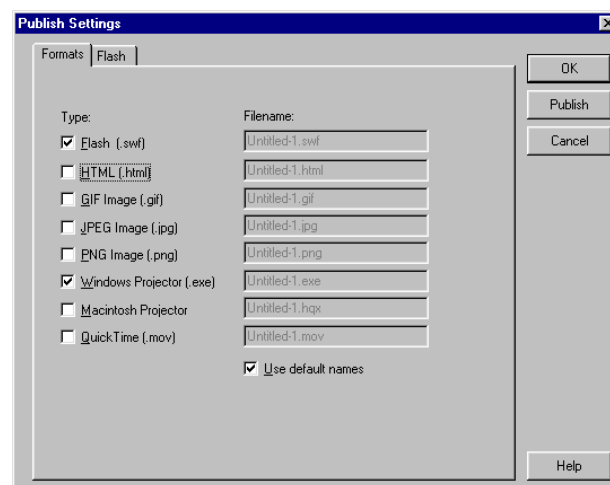
Tweened animation ada dua jenis:

- *Motion tweening*: Untuk membuat *motion tweening*, ditentukan dulu *keyframe* awal dan akhir. Secara otomatis Flash akan men-*tween frame-frame* yang ada diantaranya sehingga terbentuklah animasi. Pada animasi ini, dapat diubah *property* dari *instance*, *group* ataupun *text block*. Flash dapat men-*tween* posisi, ukuran, rotasi, dan memiringkan *instance*, *group* ataupun tulisan.
- *Shape tweening*: Dengan *shape tweening* kita dapat merubah bentuk dari suatu *object* ke *object* lain.

2.1.3. Publikasi Movie Flash

Agar pengguna dapat melihat hasil movie flash yang dipergunakan dalam program ini, diperlukan publikasi file dari format FLA menjadi format EXE. Secara *default*, publikasi file Flash menghasilkan file SWF dan HTML, untuk itu berikut ini langkah-langkah mempublikasi format file flash menjadi file EXE

- Buka File FLA
- Pilih File > Publish Settings. Pada kotak dialog Publish Settings, klik pada tab Format. Terdapat berbagai macam format seperti format Flash (SWF), HTML, GIF Image, JPEG Image, PNG Image, Windows Projector (EXE), Machintosh Projector, dan Quick Time.
- Tandai *check box* untuk Windows Projector, Tekan tombol Publish.



Gambar 2.1. Kotak Dialog Publish pada Flash

2.2. *ActionScript*

ActionScript merupakan bahasa pemrograman atau *scripting* di Flash. *ActionScript* pada Flash MX mempunyai banyak kemiripan dengan bahasa pemrograman *JavaScript*. European Computers Manufactures Association (ECMA) menulis dokumen yang disebut ECMA-262 dan diambil dari *JavaScript* sebagai standar internasional untuk bahasa *JavaScript*. *ActionScript* berdasar atas spesifikasi ECMA-262 tersebut.

Dengan *ActionScript*, dapat dibuat tombol-tombol navigasi dan elemen yang dapat berinteraksi dengan pengunjung. Pada animasi sebelumnya yaitu: animasi dengan menggunakan *frame-by-frame animation* maupun yang menggunakan *shape tweening*, Flash hanya menjalankan *frame-frame* secara berurutan sehingga hanya membentuk animasi dan berhenti pada *frame* terakhir. Animasi ini disebut dengan animasi linier. Sedangkan animasi dengan *ActionScript* animasi dapat dijalankan dengan menampilkan adegan-adegan tertentu saja tanpa harus menampilkan animasi secara keseluruhan. Animasi ini disebut dengan animasi non-linear, yaitu animasi yang tidak harus berjalan dari awal hingga akhir frame.

ActionScript dapat membuat movie lebih interaktif, karena *user* dapat menggunakan *mouse*, *keyboard*, atau keduanya untuk mengendalikan animasi, menggerakkan *object* yang ada, memasukkan form isian dan masih banyak lagi. Kumpulan perintah yang ditulis dalam bahasa *ActionScript* dijalankan saat kejadian (*event*) tertentu terjadi.

Ada dua jenis mode untuk mengedit ataupun menulis *ActionScript* ini:

1. *Normal mode*: Mode ini memberikan kemudahan bagi *programmer* untuk melakukan *script*-ing pada Flash, karena Flash telah menyediakan daftar *action*, sehingga *programmer* tinggal meng-*click* dan memberikan beberapa parameter tertentu saja, tanpa perlu banyak mengetik dan menghafalkan *ActionScript*.
2. *Expert mode*: dengan menggunakan mode ini *programmer* menulis dan mengetik *action* secara langsung pada kotak tulisan yang telah disediakan. Mode ini cocok digunakan untuk mereka yang telah mahir menulis program.

2.2.1. Kategori *ActionScript*

Di dalam *Action* Panel, action dibagi dalam berbagai kategori yaitu:

Tabel 2.1. Tabel Kategori *ActionScript*

No	Kategori	Keterangan
1	Action	Kategori ini menampung berbagai action yang sering digunakan untuk Flash Movie
2	Operators	Kategori ini berisi simbol yang digunakan misalnya untuk operasi logika, matematika, seperti tambah, kurang, kali, dan lain-lain
3	Function	Kategori ini meliputi berbagai <i>predefined function</i> yang sudah tersedia di Flash
4	Constants	Kategori ini memuat berbagai nilai constant
5	Properties	Kategori ini berisi property obyek yang dimodifikasi. Sebagian besar property ini digunakan untuk obyek <i>movie clip</i>
6	Objects	Flash memiliki berbagai kelas obyek yang sudah didefinisikan. Flash MX memiliki banyak obyek baru seperti button, stage, dan lain-lain

2.3. Adobe Photoshop

Adobe Photoshop 7.0 merupakan produk terbaru dari Adobe Systems Incorporated yang telah begitu dikenal di dalam produk aplikasi yang berbasis grafis. (Mulyana, 2002) *Software* ini untuk memanipulasi image yang mampu mengkonversikan berbagai macam file image dari satu format ke format lain, dengan menggunakan *layer-layer*. Banyak sekali inovasi yang muncul dari produk Adobe, termasuk Adobe Photoshop 7.0. Perubahan itu selalu dimunculkan dengan tujuan untuk kenyamanan pengguna secara umum. Fitur-fitur baru selalu dimunculkan merupakan pengembangan dan perbaikan dari *tool* sebelumnya dan semakin memperkaya sarana menuju ke kesempurnaan bagi penggunanya.

2.4. Borland Delphi 7.0

Borland Delphi 7.0 adalah bahasa pemrograman yang bekerja dalam lingkup *MS-Windows*. Kemampuan *Borland Delphi 7.0* secara umum adalah menyediakan komponen-komponen yang memungkinkan *programmer* membuat aplikasi yang sesuai dengan tampilan dan cara kerja *MS-Windows*, diperkuat dengan bahasa pemrograman terstruktur yaitu *Object Pascal*.

Pemrograman *database* pada *Borland Delphi 7.0* tersedia format *database Paradox* dan *dBase*, selain itu juga dapat menangani berbagai macam format *database*, antara lain *MS-Access*, *ODBC*, *SyBASE*, *Oracle*, *IB-Expert* dan lain-lain.

2.5. Firebird 1.5.

Firebird adalah relational *database* yang menawarkan banyak model dari ANSI SQL-92 yang dapat dijalankan pada LINUX, Windows, dan UNIX. Firebird menawarkan *excellent concurrency, high performance, and powerful language* yang mendukung penyimpanan *procedures and triggers*

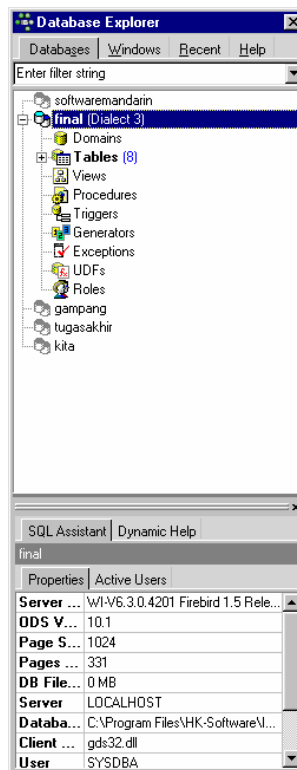
Sistem *database* Firebird pertama kali diluncurkan pada tanggal 25 Juli 2000, sejauh ini Firebird telah dikembangkan oleh sebuah tim independent yang terdiri dari kode dasar (source code) yang biasanya dipakai oleh Interbase. Firebird ini dikeluarkan oleh Borland dan berada dibawah Interbase Public Licence V 1.0.

Pengembangan dari kode-kode dasar Firebird yang kedua dimulai pada awal tahap pengembangan Firebird yang pertama. Kode-kodenya yang pertama berasal dari bahasa pemrograman C menjadi C++. Pada pengembangan ini juga dilakukan optimasi kode secara besar-besaran. Firebird 1.5. adalah release pertama yang dihasilkan pada saat pengembangan Firebird yang kedua. Walaupun hal tersebut merupakan tonggak sejarah yang penting bagi para developer dan bagi keseluruhan proyek Firebird, namun ini bukanlah akhir dari proyek Firebird. Ketika Firebird 1.5. diluncurkan, pengembangan proyek ini terus berjalan menuju pada tahap berikutnya dimana Firebird yang kedua diluncurkan.

2.6. IB Expert

IB Expert adalah salah satu *software* yang dikembangkan untuk membuat *database* dalam format Interbase dan Firebird. IB Expert memberikan kemudahan dalam pengembangan *database*, misalnya dengan Visual editor untuk semua objek *database*, editor SQL dan editor *Script*, pencari kesalahan untuk prosedur – prosedur SQL yang dibuat, pengembang Query dan editor untuk mendesain suatu *database*.

Dengan menggunakan editor visual yang dimiliki oleh IB Expert, pengguna akan dimudahkan untuk membentuk obyek–obyek dalam *database* tanpa menulis pengkodeannya. IB Expert juga memiliki *Database Explorer* untuk melihat semua obyek dalam *database*, dan dapat mengelompokkannya (*Grouping*) dalam satu direktori yang dapat didefinisikan sendiri. Contoh *Database Explorer* dapat dilihat pada gambar 2.2.



Gambar 2.2. *Database Explorer*

2.7. Interaksi Manusia Komputer

User adalah orang yang awam dengan teknologi dan komputer. Agar *user* dapat menggunakan aplikasi Sistem Informasi yang dibuat maka sistem harus dibuat sedemikian rupa agar dapat dengan mudah dipahami oleh *user*.

Interaksi Manusia dan Komputer merupakan ilmu yang membahas bagaimana sistem yang dibuat harus sederhana dan mudah digunakan oleh *user*. Di dalamnya diajarkan konsep-konsep bagaimana mengolah suatu sistem sehingga outputnya dapat mudah dipahami oleh *user* (*user friendly*).

User Interface yang baik haruslah memungkinkan *user* memahami domain permasalahannya untuk menggunakan aplikasi tanpa harus membaca manual atau menerima training terlebih dahulu.

Screen layout dan desain dibedakan menjadi 5 kategori yaitu:

1. *General layout*: dalam hal ini, hanya informasi yang penting saja yang ditampilkan, desain sesuai dengan standard dan mengikuti konsistensi dengan keseluruhan isi. Tombol yang ada umum dan mudah dipahami *user*.
2. *Text*: pesan yang ingin disampaikan ditulis dengan singkat dan jelas, tidak boleh bertele-tele, menghindarkan kalimat negatif jika ingin menyampaikan pesan, isi yang disampaikan disesuaikan dengan kebutuhan *user*.
3. *Number*: jika ada tampilan yang berupa angka sebaiknya dibuat rata kanan,urut secara decimal, hindari penggunaan angka 0 yang berlebihan jika tidak dibutuhkan.
4. *Coding*: jika ingin memberikan efek *blink*, jangan semua diberi *blink*, hanya pesan yang penting saja. Jika ingin memberi efek *bold*, lebih cocok untuk judul. Jika ingin menggunakan suara untuk membantu memperjelas pesan, suara disesuaikan dengan tampilan, dan pesan yang dimaksud.
5. *Color*: warna disesuaikan dengan tampilan dan pesan yang akan disampaikan, situasi ruangan pada saat itu, hindari penggunaan warna yang berlebihan yang menimbulkan kesan membingungkan.

2.7 Flowchart

Flowchart adalah bagan yang menggambarkan urutan instruksi untuk proses dengan komputer dan hubungan antara suatu proses dengan proses lainnya, dengan menggunakan simbol-simbol tertentu.

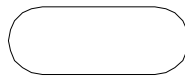
Program *flowchart* adalah suatu bagan dengan simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses dengan proses lainnya dalam suatu program.

Dalam membuat program *flowchart* tidak ada rumus atau patokan yang mutlak. Karena pada dasarnya pada dasarnya program *flowchart* adalah gambaran hasil pemikiran orang dalam menganalisa penyelesaian suatu masalah dengan komputer. Yang penting di dalam membuat program *flowchart* adalah bahwa bila langkah-langkah yang digambarkan dalam *flowchart*, dilaksanakan oleh komputer, akan menghasilkan hasil yang benar.

Bagan alur program (program *flowchart*) adalah suatu bagan yang menggambarkan arus logika dari data yang akan diproses dalam suatu program dari awal sampai akhir. Bagan alur program merupakan alat yang berguna *programmer* untuk mempersiapkan program yang rumit. Bagan alur terdiri dari simbol-simbol yang mewakili fungsi-fungsi langkah program dan garis alur (*flow lines*) menunjukkan urutan dari simbol-simbol yang akan dikerjakan.

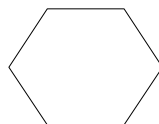
Berikut ini adalah simbol-simbol program *flowchart* menurut ANSI (American National Standard Institute):

1. Simbol terminal : digunakan untuk menunjukkan awal dan akhir dari program



Gambar 2.3. Simbol terminal

2. Simbol persiapan: digunakan untuk memberi nilai awal pada suatu variable



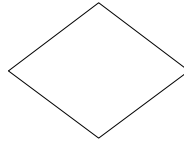
Gambar 2.4. Simbol persiapan

3. Simbol pengolahan: digunakan untuk pengolahan arithmatika dan pemindahan data



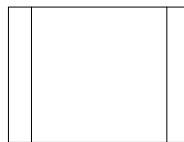
Gambar 2.5. Simbol pengolahan

4. Simbol keputusan: digunakan untuk mewakili operasi perbandingan logika



Gambar 2.6. Simbol keputusan

5. Simbol proses terdefinisi: digunakan untuk proses yang detilnya dijelaskan terpisah, misalnya dalam bentuk routine



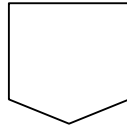
Gambar 2.7. Simbol proses terdefinisi

6. Simbol penghubung: digunakan untuk menunjukkan hubungan arus proses yang terputus masih dalam halaman yang sama



Gambar 2.8. Simbol penghubung

7. Simbol penghubung halaman lain: digunakan untuk menunjukkan hubungan arus proses yang terputus masih dalam halaman yang sama



Gambar 2.9. Simbol penghubung halaman lain

2.8. Desain Sistem Informasi

Desain Sistem Informasi merupakan suatu tahap yang sangat penting di dalam pengembangan Sistem Informasi. Implementasi program dibuat berdasarkan desain yang dibuat sebelumnya. Jika desain baik dan sesuai dengan kebutuhan *user* maka implementasi program juga akan baik dan memenuhi kebutuhan *user*. Proses desain dilakukan dengan menggunakan *software Power Designer*

2.8.1. User Requirement

Seringkali *programmer* memiliki istilah-istilah yang tidak umum dipakai di dalam menggambarkan sistemnya. Sehingga output yang dihasilkan tidak mudah dipahami oleh user (kurang *user friendly*). Untuk memperoleh gambaran yang jelas tentang kebutuhan *user* dilakukan wawancara dengan orang yang berkompeten di dalam sistem. Tahap awal di dalam mendesain Sistem Informasi adalah melakukan analisa *user requirement*. Hal ini penting karena Sistem Informasi yang dibuat harus dapat memenuhi kebutuhan *user*.

Pembangunan Sistem Informasi harus menggunakan istilah-istilah yang biasa dipakai *user* agar *user* mengerti dan memahami sistem yang dibuat.

2.8.2. Entity Relationship Diagram

Perancangan ERD sangat penting, karena ERD ini nantinya merupakan logical model dari *database* yang dibuat.

Entity Relationship Diagram (ERD) adalah diagram yang menggambarkan hubungan antar entitas di dalam sistem. Entitas merupakan tabel-tabel yang

nantinya akan dibuat pada sistem *database*, lengkap dengan field-fieldnya, tipe data dan key yang dimiliki. Hubungan antar tabel dibuat dengan menghubungkan key dengan foreign key antar tabel.

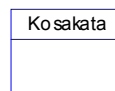
Ada 7 langkah yang harus dilakukan dalam membuat ERD:

- a. Mengidentifikasi entitas
- b. Mengidentifikasi hubungan
- c. Menyiapkan ERD kasar
- d. Memetakan elemen-elemen data pada entitas
- e. Membuat analisis data
- f. Menyiapkan ERD yang telah dimodifikasi
- g. Menelaah ERD bersama pemakai dan memperbaikinya

Dalam menggambar ERD, ada beberapa komponen yang perlu untuk diperhatikan, yaitu:

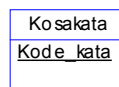
- *Entity*

Entity digambarkan dengan kotak segiempat. Tiap kotak diberi label. Contoh *entity* dapat dilihat pada gambar 2.10.



Gambar 2.10. *Entity*

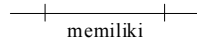
Identifikasi dan deskripsi dari *entity* dijelaskan oleh *attribute*-nya. Contoh *attribute* dari Kosakata adalah Kode Kata. *Attribute* yang mengidentifikasi *entity* disebut *identifier* (contohnya Kode Kata tidak boleh ada yang sama), sedangkan *attribute* lain yang mendeskripsikan *entity* disebut *descriptors*. *Identifier* biasa ditunjukkan dengan garis bawah. Contoh *attribute* dapat dilihat pada gambar 2.11.



Gambar 2.11. *Attribute*

- *Relationship*

Relationship adalah hubungan yang terjadi antar dua *entity*. Simbol *relationship* dapat dilihat pada gambar 2.12.



Gambar 2.12. *Relationship*

Relationship terdiri dari 3 macam yaitu: *one-to-one*, *one-to-many*, dan *many-to-many*. Contoh dapat dilihat pada gambar 2.13, gambar 2.14, dan gambar 2.15.



Gambar 2.13. *One to one*

Pada relasi *one to one* berarti tiap *record* pada *entity* 1 hanya dapat berhubungan dengan sebuah *record* pada *entity* 2.



Gambar 2.14. *One to many*

Pada relasi *one to many* berarti tiap *record* pada *entity* 1 dapat berhubungan lebih dari satu *record* pada *entity* 2.



Gambar 2.15. *Many to many*

Pada relasi *many to many* berarti banyak *record* pada *entity* 1 dapat berhubungan dengan lebih dari satu *record* pada *entity* 2.