

Rancang Bangun Aplikasi Pembelajaran Biologi untuk Sistem Pernafasan pada Hewan Berbasis Multimedia

Badani Haryono Putra (Danie_3193@yahoo.com)
Jurusan Teknik Informatika
STMIK GI MDP / AMIK GI MDP / STIE MDP

Abstrak : Tujuan penulisan skripsi ini adalah untuk mempermudah proses sistem belajar mengajar. Adapun permasalahan pada saat belajar adalah sulitnya menerima materi pelajaran, kurangnya minat anak didik dalam penyampaian materi bahan ajar yang memerlukan simulasi dan tidak adanya media pembelajaran yang mendukung proses belajar mengajar tersebut. Aplikasi pembelajaran berbasis multimedia nantinya diharapkan mampu untuk menyampaikan informasi yang lebih cepat dan akurat. Dalam permasalahan ini penulis menggunakan metode *RUP* dan merancang sebuah aplikasi pembelajaran dan nilai berbasis *multimedia*. Aplikasi pembelajaran *multimedia* ini diharapkan dapat diterapkan dan diimplentasikan guna membantu dalam mempermudah proses belajar mengajar.

Kata Kunci : Aplikasi Pembelajaran, *Multimedia*.

1 PENDAHULUAN

Perkembangan ilmu pengetahuan dan teknologi komputer semakin menunjukkan kemajuan yang sangat pesat. Pemanfaatannya sudah hampir meliputi berbagai hal dan lapisan. Salah satunya adalah ditandai dengan makin maraknya instansi-instansi yang menerapkan teknologi komputer. Teknologi informasi merupakan salah satu bentuk pemanfaatan dari kemajuan teknologi komputer yang sangat mendasar. Dengan perkembangan teknologi komputer yang demikian pesat dapat mengatasi kesulitan masyarakat dalam penyediaan informasi secara tepat dan aktual. Multimedia sebagai bagian dari sistem komputerisasi dapat menjawab informasi yang bersifat interaktif dan menampilkan suara, tulisan, gambar serta keterangan tentang informasi yang lebih efektif dan efisien.

Kecanggihan dari aplikasi multimedia ini dapat dengan cepat menarik perhatian dan rasa ingin tahu seseorang ketika memanfaatkan teknologi tersebut serta

memudahkan untuk memahami informasi yang disampaikan.

Di dalam proses belajar mengajar tidak luput dari penyampaian informasi yang di sampaikan oleh pengajar yang sedapat mungkin menyampaikan materi dengan baik, tapi dalam beberapa materi pelajaran siswa sering tidak memahami materi yang disampaikan oleh pengajar akibat kurangnya minat siswa terhadap pelajaran tersebut, karena kurangnya informasi yang interaktif.

Disini penulis ingin memberikan solusi terhadap materi ajar yang kurang dimengerti oleh siswa dengan memberikan materi bahan ajar yang lebih informatif, interaktif dan mudah di pahami dengan system pembelajaran berbasis multimedia, menampilkan simulasi materi dengan tampilan yang menarik agar siswa dapat menangkap materi dengan baik.

Berdasarkan latar belakang masalah di atas, maka diambil judul skripsi ” RANCANG BANGUN APLIKASI PEMBELAJARAN BIOLOGI UNTUK SISTEM PERNAFASAN PADA HEWAN

2 LANDASAN TEORI

2.1 Sistem Informasi

Sistem informasi adalah suatu sistem dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan informasi yang diperlukan untuk pengambilan keputusan.

Perkembangan sistem informasi telah menyebabkan terjadinya perubahan yang cukup signifikan dalam pola pengambilan keputusan yang dilakukan oleh manajemen baik pada tingkat operasional. Perkembangan ini juga telah Input Proses Output Umpan Balik Umpan Balik Output menyebabkan perubahan-perubahan peran dari para manajer dalam pengambilan keputusan, mereka dituntut untuk selalu dapat memperoleh informasi yang paling akurat dan terkini.

Meningkatnya penggunaan teknologi informasi, khususnya internet, telah membawa setiap orang dapat melaksanakan berbagai aktivitas dengan lebih akurat, berkualitas, dan tepat waktu. Setiap organisasi dapat memanfaatkan internet dan jaringan teknologi informasi untuk menjalankan berbagai aktivitasnya secara elektronik.

Sistem informasi dalam suatu organisasi dapat dikatakan sebagai suatu sistem yang menyediakan informasi bagi semua tingkatan dalam organisasi tersebut kapan saja diperlukan. Sistem ini menyimpan, mengambil, mengubah, mengolah dan mengkomunikasikan informasi yang diterima dengan menggunakan sistem informasi atau peralatan sistem lainnya.

Definisi sistem informasi dalam bukunya Abdul Kadir yang berjudul Pengenalan Sistem Informasi, yaitu: “system informasi adalah kerangka kerja yang mengkoordinasikan sumber daya (manusia, komputer) untuk mengubah masukan (input) menjadi keluaran (informasi), guna mencapai sasaran-sasaran perusahaan”. (Kadir, 2003,h:11)

2.2 Metode RUP

Metodologi yang digunakan dalam pengembangan aplikasi ini adalah *Rational Unified Process* (RUP). Dimulai dari hal analisa kebutuhan sistem, tahap perancangan sampai dengan tahap implementasi semuanya menggunakan metodologi ini. RUP membagi tahapan pengembangan perangkat lunaknya ke dalam 4 tahap sebagai berikut:

1. *Inception* (Analisa sistem)

Pada tahapan analisis, penulis melakukan analisis sistem, meliputi analisis masalah aplikasi pembelajaran dan analisis kebutuhan fungsional yang digambarkan melalui diagram *use case* seperti pada tabel 3.2 dan analisis kebutuhan non fungsional, analisis pemecahan masalah dan *logika procedural sistem* dengan *activity diagram*.

2. *Elaboration* (Desain sistem)

Melalui hasil analisis pada tahap *inception* (tahap analisis), maka pada tahapan berikutnya penulis mulai merancang sistem secara lengkap, mulai dari rancangan *class diagram*, deskripsi *sequence diagram* dan merancang tampilan antar muka..

3. *Construction* (Pembuatan sistem)

Pada tahap ini awalnya penulis memeriksa kembali hasil dari analisis dan desain aplikasi pada tahap *inception* dan *elaboration* apakah telah sesuai dengan analisis yang dilakukan, setelah selesai maka penulis mulai mengimplementasikan pembuatan aplikasi pada perangkat keras..

4. *Transition* (Tahap Pengembangan)

Setelah aplikasi selesai diimplementasikan maka penulis menyerahkan aplikasi kepada pengguna yang merupakan target dari pengembangan aplikasi ini. Penulis juga meminta respon dari para pengguna tentang aplikasi ini dengan menyerahkan kuisisioner untuk melihat tingkat kepuasan dari pengguna.

2.3 Adobe Flash

Sejak diakuisisi oleh perusahaan rekasasa adobe, maka *Software* multimedia Macromedia Flash berubah nama menjadi Adobe Flash. Akuisisi inipun bisa jadi merupakan pertanda bahwa prospek pembuatan animasi menggunakan Flash akan semakin berkembang.

Flash sudah dipakai luas sejak puluhan tahun yang lalu. Sebagian kalangan menggunakannya untuk membuat animasi untuk halaman website, profil perusahaan, cd interaktif, game dan lain-lain. Sekarang mulai berkembang penggunaan Flash untuk pembuatan game di mobile device seperti handphone, PDA, dll.

Setiap *software* memiliki kelebihan dan kekurangan. Adobe photoshop memiliki fitur untuk menggambar yang luar biasa, tetapi tidak menganimasikan. Adobe After Effect memiliki kemampuan animasi yang luar biasa, tapi tidak untuk menggambar objek. Objek-objek yang digunakan adobe after effect adalah import dari output software. Selain memiliki kemampuan menggambar Flash juga bias sekaligus menganimasikannya. Memang efek-efek gambarnya tidak secanggih dan seberagam adobe photoshop, tapi sudah cukup untuk menggambar objek-objek terlihat cantik dan menarik.

Di dalam Flash, kita bias memasukkan rumus fisika, matematika atau rumus-rumus lainnya dengan bentuk Action Script. Sehingga kita bias menyimulasikan mobil yang bergerak dengan kecepatan dan percepatan tertentu, Sebuah peluru yang dilontarkan dengan sudut elevasi tertentu atau grafik sebuah persamaan matematika. Semuanya menjadi mungkin dan mudah dengan Flash.

3. Analisis Kebutuhan

“Analisis sistem (*System Analysis*) merupakan sebuah teknik pemecahan masalah yang menguraikan sebuah sistem menjadi bagian-bagian komponen dengan tujuan mempelajari seberapa bagus bagian-bagian komponen tersebut bekerja dan berinteraksi untuk meraih tujuan mereka” (Jeffrey L. Whitten 2006, h. 9).

Analisis sistem dilakukan sebelum tahap desain sistem (*system design*) dan merupakan tahap-tahap kritis dan sangat penting, karena kesalahan di dalam tahap ini akan menyebabkan juga kesalahan di tahap selanjutnya.

3.1 Analisis Kebutuhan Sistem

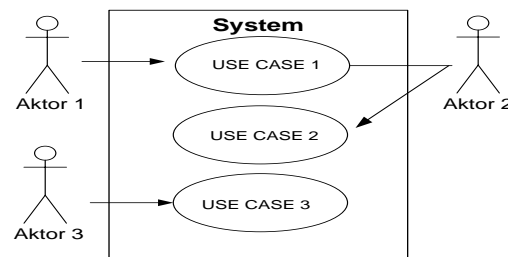
Kebutuhan sistem terbagi menjadi dua yaitu kebutuhan fungsional dan kebutuhan non fungsional.

3.2 Kebutuhan Fungsional

Kebutuhan fungsional adalah deskripsi layanan sistem yang harus disediakan, bagaimana sistem beraksi pada input tertentu dan bagaimana perilaku sistem pada situasi tertentu. Di dalam analisis kebutuhan fungsional, pemodelan yang akan digunakan adalah pemodelan dengan menggunakan *use case*. Pemodelan *use-case* adalah proses pemodelan fungsi fungsi sistem dalam kontak peristiwa-peristiwa bisnis, siapa yang mengawalnya, dan bagaimana sistem itu merespon hal tersebut.

1. Notasi Use Case

Diagram *Use Case* menunjukkan 3 aspek dari sistem yaitu *actor*, *use case* dan *system/sub system boundary*. Aktor mewakili peran orang, Sistem yang lain atau alat ketika berkomunikasi dengan *use case*. Gambar 2.5 mengilustrasikan aktor, *use case* dan *system*.



Sumber : *Permodelan Visual dengan UML*, Munawar.

Gambar 2.23 Use Case Model

2. Identifikasi Aktor

Untuk mengidentifikasi *actor*, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks *target system*. Aktor adalah abstraksi dari orang dan sistem yang lain yang

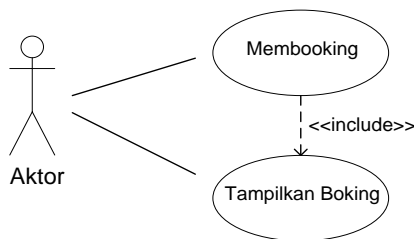
mengaktifkan fungsi dan target system. Aktor berinteraksi dengan *use case* tetapi tidak memiliki kontrol atas *use case*.

3. Deskripsi Use Case

Use Case adalah abstraksi interaksi antara sistem dan actor. *Use Case* dibuat berdasarkan keperluan actor. *Use Case* harus merupakan ‘apa’ yang dikerjakan *software* aplikasi, bukan ‘bagaimana’ *software* aplikasi mengerjakannya. Setiap *Use Case* harus diberi nama yang menyatakan apa hal yang dicapai dari hasil interaksinya dengan actor. Nama *Use Case* boleh terdiri dari beberapa kata dan tidak boleh ada dua *use case* yang memiliki nama yang sama.

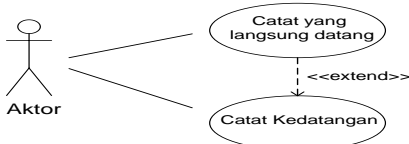
4. Stereotype

Stereotype adalah model khusus yang terbatas untuk kondisi tertentu. Untuk menunjukkan *stereotype* digunakan symbol “<< diawalnya dan ditutup >>” diakhirnya. <<extend>> digunakan untuk menunjukan satu *use case* merupakan tambahan fungsional dari *use case* yang lain jika kondisi atau syarat tertentu terpenuhi. Sedangkan <<include>> digunakan untuk menggambarkan suatu *use case* sepenuhnya merupakan fungsional dari *use case* lainnya. Contoh penggunaan <<extend>> dan <<include>> dapat dilihat pada Gambar 2.24 dan 2.25



Sumber : Permodelan Visual dengan UML, Munawar

Gambar 2.24 Use Case Inclusion



Sumber : Permodelan Visual dengan UML, Munawar

Gambar 2.25 Use Case Extension

3.3 Kebutuhan Non Fungsional

Kebutuhan non fungsional merupakan fitur-fitur pelengkap yang menunjang kerja sebuah sistem dan mempunyai pengaruh yang tidak langsung.

Kebutuhan non fungsional menentukan kualitas layanan. Terdapat beberapa patokan yang digunakan untuk menentukan kebutuhan non fungsional. Kebutuhan kualitas, sebagai salah satu faktor pada kebutuhan non fungsional, yang dapat dinyatakan secara rinci dan konkret dalam sebuah pernyataan.

Dalam menentukan kebutuhan non fungsional, faktor kebutuhan kinerja dapat juga mengacu pada kebutuhankeseluruhan sistem namun tidak dalam kerangka pola penggunaan, skalabilitas maupun dampak pemanfaatan sistem.

3.4 Logika Prosedural Sistem

Logika prosedural sistem dapat digambarkan dengan activity diagram. *Activity Diagram* adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity Diagram* mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah *activity diagram* bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa.

Berikut ini simbol-simbol yang digunakan pada saat pembuatan *activity diagram*, tampak pada tabel 2.4.

Tabel 2.2 Simbol-Simbol yang Sering Dipakai pada Activity Diagram

Simbol	Keterangan
	Titik awal
	Titik akhir
	Activity
	Pilihan untuk pengambilan keputusan
	Fork, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
	Tanda pengiriman
	Tanda penerimaan
	Tanda waktu
	Aliran akhir (Flow Final)

Sumber : Permodelan Visual dengan UML, Munawar

Node pada *activity diagram* disebut dengan *action* bukan *activity*. *Activity* menunjuk ke urutan *action*, sehingga diagram tersebut menunjuk *activity* yang membangun *action*. Perilaku bersyarat ditunjukkan dengan *decision* dan *merge decision* hanya mempunyai satu aliran masuk dan beberapa *guard* untuk aliran keluar. Setiap aliran keluar mempunyai *guard* yaitu *boolean* yang ditempatkan pada kurung kotak. Setiap kali mencapai *decision* hanya bisa mengambil satu keputusan, sehingga *guard* harus *mutually exclusive*. Pengguna [else] sebagai *guard* menunjuk bahwa *guard* yang lain salah.

3.5 Perancangan Sistem

Rancangan sistem terdiri dari rancangan diagram kelas dan rancangan diagram *sequence*.

3.5.1 Rancangan Kelas

Class diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. *Class diagram* juga menunjukkan property dan operasi sebuah class dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut.



Sumber : *Permodelan Visual dengan UML, Munawar.*

Gambar 2.26 Class Diagram

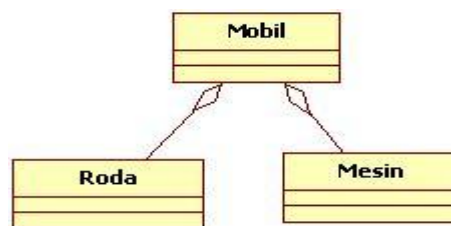
Hubungan Keterkaitan antar Berbagai Objek atau Kelas :

1. Aggregation

Jika dalam basis data dikenal konsep *one-to-many*, maka di dalam perancangan berbasis objek dikenal sebagai *aggregation*, di mana kedua konsep tersebut sebenarnya memiliki konsep yang sama. *Aggregation* adalah hubungan yang dapat terjadi antar objek yang bersifat *whole-part*.

Maksudnya adalah suatu objek dapat berisi atau memiliki objek-objek yang lainnya, di mana objek-objek tersebut dapat memiliki *atribut* dan *method*-nya masing-masing.

Misalnya, sebuah objek mobil dapat terdiri dari beberapa objek lain seperti roda, mesin, spion. Di mana pengujian dari hubungan semacam ini dapat dilakukan dengan kata “*HAS A*” (misal mobil *HAS A* mesin). Hubungan *aggregation* (mobil-mesin-roda) di atas dapat dinyatakan dalam bentuk *class diagram* semacam ini.

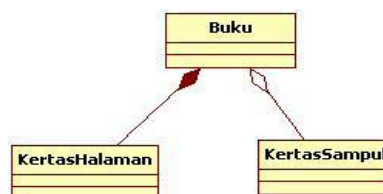


Gambar 2.27 Contoh Hubungan Aggregation

2. Composition

Composition sebenarnya hanyalah merupakan bentuk khusus dari *aggregation*. Di mana yang membedakan adalah dalam suatu bagian dari *class* tidak dapat berdiri sendiri tanpa adanya *class* lain.

Misalnya, sebuah objek buku harus terdiri dari objek seperti: *kertas_halaman*, sebuah objek buku dapat terdiri dari: *kertas_sampul* dan *tali_pembatas_buku* (lihat perbedaannya antara dapat dan harus). Hubungan *composition* (*buku-kertasHalaman-kertasSampul*) di atas dapat dinyatakan dalam bentuk *class diagram* semacam ini.



Gambar 2.10 Contoh Hubungan Composition

3. Polymorphism

Polymorphism merupakan penggunaan lebih dari 1 *method* dengan nama yang sama (pembedanya adalah berdasarkan jumlah dan jenis parameter yang didefinisikannya). *Polymorphism* bisa dibangun dengan menerapkan:

1. *Overriding* mendefinisikan *method* dengan nama sama pada *class* yang berbeda.
2. *Overloading* mendefinisikan *method* dengan nama sama dalam satu *class*.

3.5.2 Rancangan Diagram Sequence

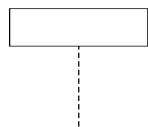
Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah *scenario*. Diagram ini menunjukkan sejumlah contoh obyek dan *message* (pesan) yang diletakkan diantara objek-objek ini di dalam *use case*.

Komponen utama *sequence diagram* terdiri atas objek yang dituliskan dengan kotak segiempat bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*.

1. Objek/Participant

Objek diletakkan di dekat bagian atas diagram dengan urutan dari kiri ke kanan. Pengertian objek hanya ada di UML 1 sedangkan di UML 2 istilah objek diganti dengan *participant*.

Setiap *participant* terhubung dengan garis titik-titik yang disebut *lifeline*. Sepanjang *lifeline* ada kotak yang disebut *activation*. *Activation* mewakili sebuah eksekusi operasi dari *participant*. Panjang kotak ini berbanding lurus dengan durasi *activation*.



Sumber : *Permodelan Visual dengan UML, Munawar*

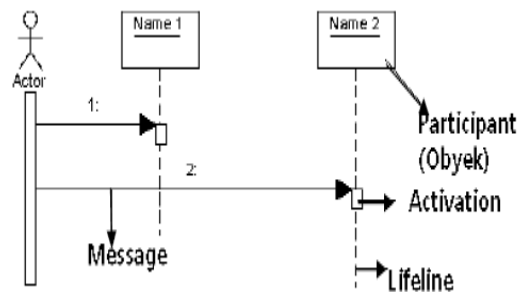
Gambar 2.29 Participant pada Sebuah Sequence Diagram

2. Message

Sebuah *Message* bergerak dari satu *participant* ke *participant* yang lain. Dan dari satu *lifeline* ke *lifeline* yang lain. Sebuah *participant* bisa mengirim *message* kepada dirinya sendiri.

3. Time

Time adalah diagram yang mewakili waktu pada arah vertikal. Waktu dimulai dari atas ke bawah. *Message* yang lebih dekat dari akan dijalankan terlebih dahulu dibanding *message* yang lebih dekat ke bawah. Simbol-simbol yang terdapat pada *Sequence diagram* dapat dilihat pada gambar 2.13.



Sumber : *Permodelan Visual dengan UML, Munawar*

Gambar 2.30 Simbol-simbol yang Ada pada Sequence Diagram

Gambar ini menunjukkan esensi simbol dari *sequence diagram* dan simbol kerjanya secara bersama-sama. *Participant* terletak di sebelah atas. Setiap *lifeline* menggunakan garis putus-putus yang menurun dari *participant*. Garis yang solid dengan tanda panah menghubungkan antara satu *lifeline* dengan *lifeline* yang lain dan mewakili sebuah *message* dari satu *participant* ke *participant* yang lain. Dari gambar tersebut terlihat seorang aktor bukan bagian dari *sequence diagram*.

4 Prosedur Uji Coba Program

Pada prosedur uji coba program akan dijelaskan tentang cara pengoperasian program serta langkah-langkah yang harus dilakukan untuk menjalankan aplikasi.

4.1 Cara Pengoperasian Program

Langkah-langkah yang dilakukan untuk dapat mengoperasikan aplikasi pembelajaran biologi untuk system pernafasan pada hewan berbasis multimedia ini adalah sebagai berikut:

1. Menginstal aplikasi pembelajaran biologi untuk system pernafasan hewan berbasis multimedia ke dalam PC/gadget yaitu dengan cara menginstall aplikasi.exe, jika PC/gadget telah terpasang aplikasi flash player maka aplikasi otomatis dapat di jalankan tanpa harus di install terlebih dahulu yaitu dengan mencari file aplikasi flash.
2. Pilih aplikasi yang berjudul aplikasi.flash yang telah diinstall pada PC/gadget untuk menjalankan aplikasi, selanjutnya *user* dapat memulai menggunakan aplikasi pada PC/gadget.

4.1.1 Langkah-langkah Menjalankan Program

Langkah awal yang harus dilakukan untuk menjalankan aplikasi pembelajaran biologi untuk system pernafasan hewan berbasis multimedia ini, yaitu membuka aplikasi yang berjudul aplikasi, kemudian mulai menjalankan aplikasi dengan melakukan pilihan yang ada pada tampilan awal menu.

4.1.1.1 Menjalankan Program Tampilan Menu Awal

Halaman tampilan Menu adalah halaman awal tampilan aplikasi, dimana Menu ini akan ditampilkan secara otomatis ketika pengguna mulai membuka aplikasi pembelajaran biologi untuk system pernafasan hewan berbasis multimedia. Pengguna dapat memilih menu-menu yang akan dipilih oleh *user* dan inilah tampilan menu awal aplikasi seperti pada gambar 4.1.



Gambar 4.1 Tampilan Halaman Menu Awal

4.1.1.2 Menjalankan Program Tombol Aplikasi

Halaman tampilan tombol aplikasi merupakan tampilan halaman pilihan ketika pengguna ingin melihat daftar aplikasi system pernafasan yang terdiri dari beberapa pilihan yaitu : aplikasi pernafasan burung, ikan, amphibi, reptil, serangga dan mamalia. Jika pengguna mengklik tombol burung maka akan ke tampilan menu aplikasi burung, Jika pengguna mengklik tombol ikan maka akan ke tampilan menu aplikasi ikan, Jika pengguna mengklik tombol serangga maka akan ke tampilan menu aplikasi serangga, Jika pengguna mengklik tombol ampibi maka akan ke tampilan menu aplikasi ampibi, Jika pengguna mengklik tombol reptil maka akan ke tampilan menu aplikasi reptil, Jika pengguna mengklik tombol mamalia maka akan ke tampilan menu aplikasi mamalia. Tampilan halaman aplikasi dapat dilihat pada gambar 4.2.



Gambar 4.2 Tampilan Aplikasi

4.1.1.3 Menjalankan Program Aplikasi Burung

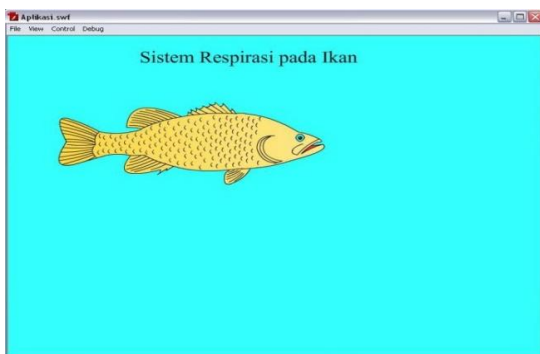
Halaman Aplikasi burung merupakan tampilan halaman pilihan ketika pengguna ingin melihat aplikasi respirasi burung setelah pengguna memilih tombol dengan tampilan “burung”, dan terdapat pilihan kembali, pause, play dan stop. Tampilan halaman aplikasi burung dapat dilihat pada gambar 4.3.



Gambar 4.3 Tampilan Halaman Aplikasi Burung

4.1.1.4 Menjalankan Program Aplikasi Ikan

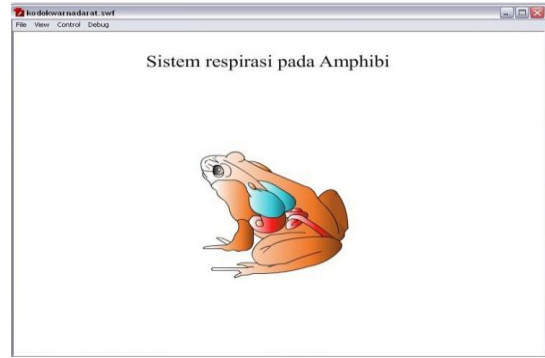
Halaman Aplikasi ikan merupakan tampilan halaman pilihan ketika pengguna ingin melihat aplikasi respirasi burung setelah pengguna memilih tombol dengan tampilan “ikan”, dan terdapat pilihan kembali, pause, play dan stop. Tampilan halaman aplikasi burung dapat dilihat pada gambar 4.4



Gambar 4.4 Tampilan Halaman Aplikasi Ikan

4.1.1.5 Menjalankan Program Aplikasi Amphibi

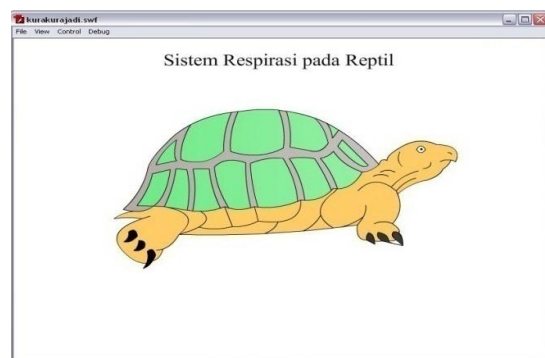
Halaman Aplikasi amphibi merupakan tampilan halaman pilihan ketika pengguna ingin melihat aplikasi respirasi burung setelah pengguna memilih tombol dengan tampilan “amphibi”, dan terdapat pilihan kembali, pause, play dan stop. Tampilan halaman aplikasi burung dapat dilihat pada gambar 4.5.



Gambar 4.5 Tampilan Halaman Aplikasi Amphibi

4.1.1.6 Menjalankan Program Aplikasi Reptil

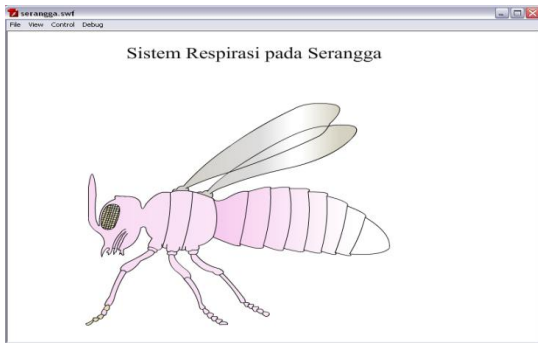
Halaman Aplikasi reptil merupakan tampilan halaman pilihan ketika pengguna ingin melihat aplikasi respirasi reptil setelah pengguna memilih tombol dengan tampilan “reptil”, dan dalam tampilan terdapat pilihan kembali, pause, play dan stop. Tampilan halaman aplikasi reptil dapat dilihat pada gambar 4.6.



Gambar 4.6 Tampilan Halaman Aplikasi Reptil

4.1.1.7 Menjalankan Program Aplikasi Serangga

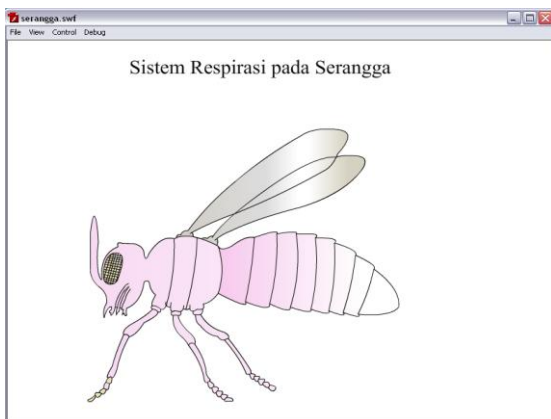
Halaman Aplikasi serangga merupakan tampilan halaman pilihan ketika pengguna ingin melihat aplikasi respirasi serangga setelah pengguna memilih tombol dengan tampilan “serangga”, dan dalam tampilan terdapat pilihan kembali, pause, play dan stop. Tampilan halaman aplikasi serangga dapat dilihat pada gambar 4.7.



Gambar 4.7 Tampilan Halaman Aplikasi Serangga

4.1.1.8 Menjalankan Program Aplikasi Mamalia

Halaman Aplikasi mamalia merupakan tampilan halaman pilihan ketika pengguna ingin melihat aplikasi respirasi mamalia setelah pengguna memilih tombol dengan tampilan “mamalia”, dan dalam tampilan terdapat pilihan kembali, pause, play dan stop. Tampilan halaman aplikasi serangga dapat dilihat pada gambar 4.8.

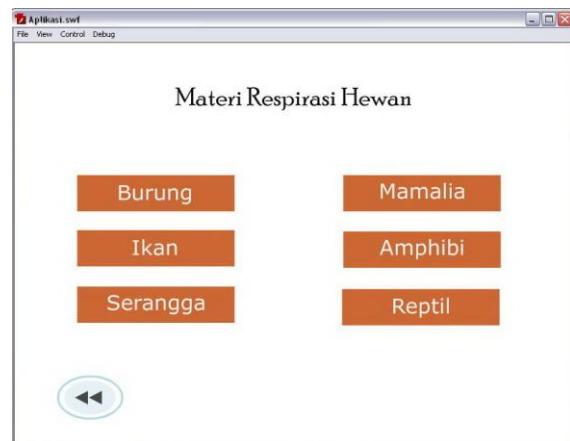


Gambar 4.8 Tampilan Halaman Aplikasi Mamalia

4.1.1.9 Menjalankan Program Tampilan Materi

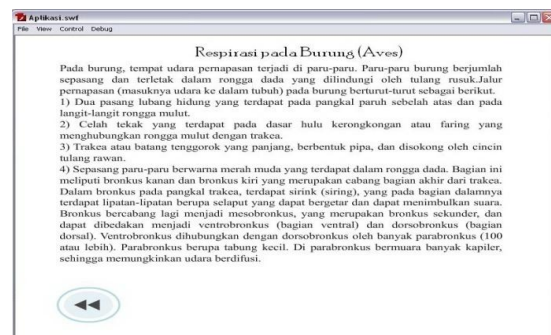
Halaman tampilan tombol materi merupakan tampilan halaman pilihan ketika pengguna ingin melihat daftar materi system pernafasan yang terdiri dari beberapa pilihan yaitu : materi pernafasan burung, ikan, amphibi, reptil, serangga dan mamalia. Jika pengguna mengklik tombol burung maka akan ke tampilan menu materi burung, Jika

pengguna mengklik tombol ikan maka akan ke tampilan menu materi ikan, Jika pengguna mengklik tombol serangga maka akan ke tampilan menu materi serangga, Jika pengguna mengklik tombol ampibi maka akan ke tampilan menu materi ampibi, Jika pengguna mengklik tombol reptil maka akan ke tampilan menu materi reptil, Jika pengguna mengklik tombol mamalia maka akan ke tampilan menu materi mamalia. Tampilan halaman aplikasi dapat dilihat pada gambar 4.9.



Gambar 4.9 Tampilan Halaman Materi 4.1.1.10 Menjalankan Program Materi Burung

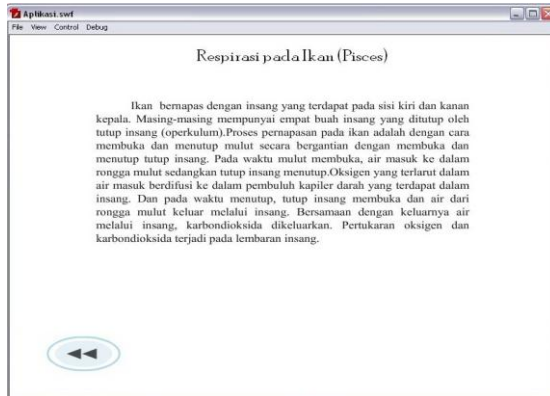
Halaman materi burung merupakan tampilan halaman pilihan ketika pengguna ingin melihat materi burung setelah pengguna memilih tombol dengan tampilan “burung”, dan dalam tampilan terdapat pilihan kembali. Tampilan halaman materi burung dapat dilihat pada gambar 4.10.



Gambar 4.10 Tampilan Halaman Materi Burung

4.1.1.11 Menjalankan Program Materi Ikan

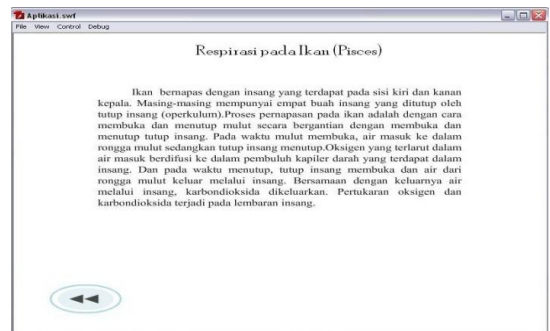
Halaman materi ikan merupakan tampilan halaman pilihan ketika pengguna ingin melihat materi ikan setelah pengguna memilih tombol dengan tampilan “ikan”, dan dalam tampilan terdapat pilihan kembali. Tampilan halaman materi ikan dapat dilihat pada gambar 4.11.



Gambar 4.11 Tampilan Halaman Materi Ikan

4.1.1.12 Menjalankan Program Materi Serangga

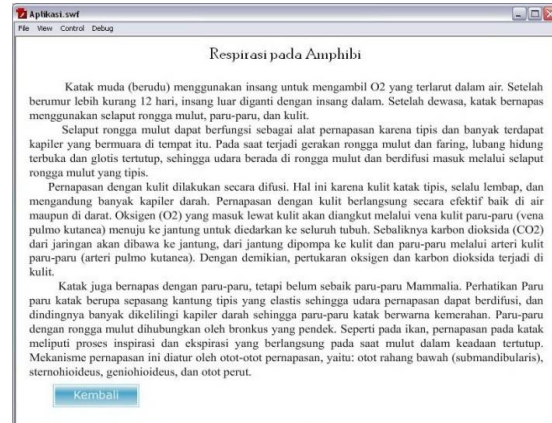
Halaman materi serangga merupakan tampilan halaman pilihan ketika pengguna ingin melihat materi serangga setelah pengguna memilih tombol dengan tampilan “serangga”, dan dalam tampilan terdapat pilihan kembali. Tampilan halaman materi serangga dapat dilihat pada gambar 4.12.



Gambar 4.12 Tampilan Halaman Materi Serangga

4.1.1.13 Menjalankan Program Materi Amphibi

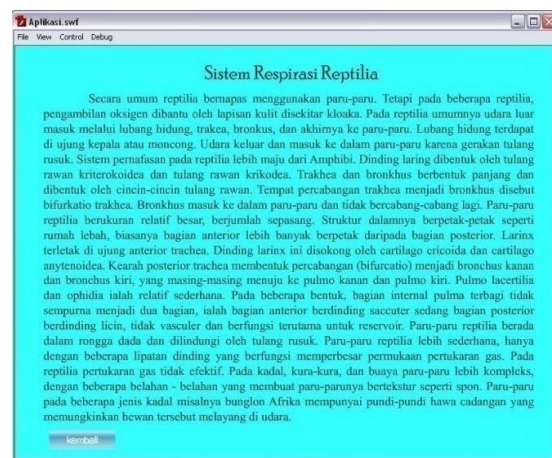
Halaman materi amphibi merupakan tampilan halaman pilihan ketika pengguna ingin melihat materi amphibi setelah pengguna memilih tombol dengan tampilan “amphibi”, dan dalam tampilan terdapat pilihan kembali. Tampilan halaman materi amphibi dapat dilihat pada gambar 4.13.



Gambar 4.13 Tampilan Halaman Materi Amphibi

4.1.1.14 Menjalankan Program Materi Reptil

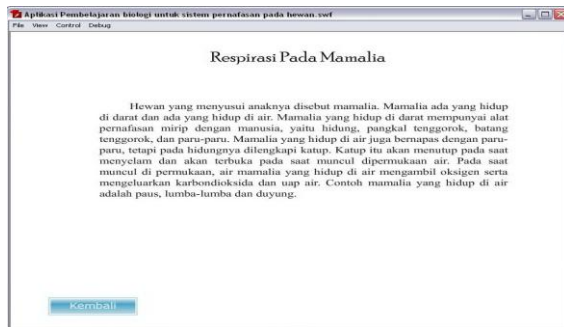
Halaman materi reptil merupakan tampilan halaman pilihan ketika pengguna ingin melihat materi reptil setelah pengguna memilih tombol dengan tampilan “reptil”, dan dalam tampilan terdapat pilihan kembali. Tampilan halaman materi reptil dapat dilihat pada gambar 4.14.



Gambar 4.14 Tampilan Halaman Materi Reptil

4.1.1.15 Menjalankan Program Materi Mamalia

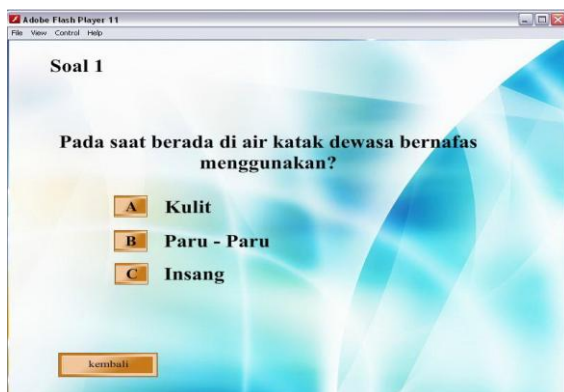
Halaman materi mamalia merupakan tampilan halaman pilihan ketika pengguna ingin melihat materi mamalia setelah pengguna memilih tombol dengan tampilan “mamalia”, dan dalam tampilan terdapat pilihan kembali. Tampilan halaman materi reptil dapat dilihat pada gambar 4.15.



Gambar 4.15 Tampilan Halaman Materi Mamalia

4.1.1.16 Menjalankan Program Latihan

Halaman tampilan latihan merupakan halaman untuk pengguna dapat menguji kemampuannya tentang respirasi pada hewan dengan menjawab pertanyaan dari halaman latihan dengan menyediakan pertanyaan 10 soal dan pada akhir soal akan muncul skor yang telah di raih dari latihan tersebut dengan menampilkan kebenaran dan kesalahan pada latihan.



Gambar 4.16 Tampilan Halaman Latihan

4.1.1.17 Menjalankan Program Tentang

Halaman tampilan tentang aplikasi merupakan tampilan halaman pilihan ketika pengguna ingin melihat informasi mengenai

aplikasi pembelajaran biologi untuk system pernafasan hewan berbasis multimedia. Setelah tampilan halaman tentang terdapat pilihan *command* kembali untuk kembali ke halaman awal.



Gambar 4.17 Tampilan Halaman Tentang

DAFTAR PUSTAKA

- [1] Sutopo, Ariesto Hadi. (2003). *Multimedia Interaktif dengan Flash*. Graha Ilmu. Yogyakarta.
- [2] Shalahuddin, Muhammad dan Rosa A.S. 2006. *Pemrograman J2ME Belajar Cepat Pemrograman Perangkat Telekomunikasi Mobile*. Informatika, Bandung.
- [3] Fathansyah, 2004, *Basis Data*. Andi Offset, Yogyakarta.
- [4] Jeffery L. Whitten, 2006, *Metode Desain dan Analisis Sistem*. Andi Offset, Yogyakarta.