

Wandah Wibawanto, S.Sn., M.Ds

Desain dan Pemrograman Multimedia Pembelajaran Interaktif



Desain dan Pemrograman Multimedia Pembelajaran Interaktif

Wandah Wibawanto, S.Sn., M.Ds.

Desain dan Pemrograman Multimedia Pembelajaran Interaktif

© All Rights Reserved

Hak cipta dilindungi undang-undang

Penulis:

Wandah Wibawanto, S.Sn. M.Ds.

Penyunting:

Dhega Febiharsa

Desain & Tata Letak:

Wandah Wibawanto, S.Sn. M.Ds.

Penerbit :

Penerbit Cerdas Ulet Kreatif

Jl. Manggis 72 RT 03 RW 04 Jember Lor - Patrang

Jember - Jawa Timur 68118

Telp. 0331-3013999

Faks. 0331-3013999

e-mail: publisher@cerdas.co.id

ISBN : 978-602-7534-26-1

Cetakan Pertama : 2017

Distributor:

Penerbit Cerdas Ulet Kreatif

Jl. Manggis 72 RT 03 RW 04 Jember Lor - Patrang

Jember - Jawa Timur 68118

Telp. 0331-3013999

Faks. 0331-3013999

e-mail: publisher@cerdas.co.id

Undang - Undang RI Nomor 19 Tahun 2002

Tentang Hak Cipta

Ketentuan Pidana

Pasal 72 (ayat 2)

Barang Siapa dengan sengaja menyiarkan, memamerkan, mengedarkan, atau menjual kepada umum suatu ciptaan atau barang hasil pelanggaran Hak Cipta atau hak terkait sebagaimana dimaksud pada ayat (1), dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp. 500.000.000,00 (lima ratus juta rupiah).

Prakata

Multimedia Pembelajaran Interaktif merupakan salah satu media yang dapat digunakan untuk menyampaikan materi pembelajaran kepada peserta didik dengan sangat efektif dan efisien. Keunggulan utama media pembelajaran interaktif yaitu interaktivitas itu sendiri membuka berbagai peluang interaksi antara pengguna dengan media. Namun demikian untuk membentuk interaktivitas yang baik diperlukan pengetahuan yang baik tentang desain antar muka dan teknik pemrograman, dua hal yang menjadi kelemahan bagi sebagian besar tenaga pengajar di Indonesia.

Minimnya literatur yang membahas baik secara teori maupun secara praktik tentang prinsip desain multimedia pembelajaran interaktif, sekaligus proses pemrograman untuk menampilkan interaktivitas pada media mengakibatkan kurang berkembangnya media-media pembelajaran interaktif yang berkualitas, yang pada akhirnya media pembelajaran hanyalah sebatas memindahkan materi statis dari media konvensional seperti buku, ke layar komputer.

Sebagai respon dari permasalahan tersebut, melalui buku ini penulis berusaha untuk membahas tentang proses desain, prinsip desain media pembelajaran sampai dengan proses pemrograman untuk membentuk interaktivitas suatu media. Pembahasan dalam bentuk teori dan praktik, dilengkapi dengan penjelasan ditiap-tiap program, dengan langkah-langkah yang paling mudah untuk diikuti oleh semua kalangan.

Semoga buku ini membawa kebermanfaatan bagi perkembangan media pembelajaran interaktif tanah air, dan mohon maaf sebesar-besarnya apabila banyak kekurangan yang ada dalam buku ini.

Penulis,

Daftar Isi

| | |
|---|-----|
| Prakata | iii |
| Daftar isi | iv |
| Bab I Pendahuluan | 1 |
| Bab II Media Pembelajaran..... | 5 |
| Manfaat Media Pembelajaran | 6 |
| Jenis Media Pembelajaran..... | 7 |
| Bab III Desain Media Pembelajaran..... | 11 |
| Design Treatment..... | 11 |
| Visual Development Phase | 19 |
| Programming Phase | 20 |
| Testing Phase | 21 |
| Prinsip Desain Media Pembelajaran | 22 |
| Bab IV Adobe Flash..... | 29 |
| Ruang kerja Adobe Flash..... | 30 |
| Bab V Membuat Animasi Sederhana..... | 33 |
| Animasi Frame by Frame..... | 36 |
| Animasi Frame by Frame dengan Gambar Siap Pakai | 40 |
| Animasi Motion Tween | 42 |
| Prinsip Dasar Animasi Tween Dalam Flash | 45 |
| Animasi Tween dengan Ease | 45 |
| Animasi Tween dengan Rotasi | 50 |
| Animasi Roda Gigi | 51 |
| Animasi Motion Guide | 54 |
| Memahami Movieclip..... | 57 |
| Animasi Motion Guide dengan Movieclip Kompleks | 59 |
| Animasi Menggunakan Metode Graphic | 64 |
| Memanfaatkan Layer dalam Animasi | 67 |
| Memahami Masking | 70 |
| Membuat Animasi Bumi Berotasi | 71 |
| Animasi Shape Tween | 74 |

| | |
|---|-----|
| Bab VI Dasar Pemrograman Action Script 3 | 75 |
| Mencoba Kode Actionscript | 75 |
| Syntax | 77 |
| Variable..... | 80 |
| Access control attributes..... | 81 |
| Mendeklarasikan Variabel | 81 |
| Constants..... | 81 |
| Data Types | 81 |
| Operators..... | 82 |
| Function | 83 |
| Loops | 84 |
| Break dan Continue | 85 |
| Conditional if..... | 86 |
| Conditional switch..... | 88 |
| Array | 88 |
| Mengendalikan Timeline dengan Actionscript..... | 88 |
| Bab VII Interaktifitas..... | 91 |
| Movieclip properties | 94 |
| Instance Name | 95 |
| Menambah objek dengan AddChild | 96 |
| EventListener | 98 |
| MouseEvent | 99 |
| Drag and Drop | 100 |
| Drag dan Snap..... | 101 |
| EnterFrame Event | 103 |
| Keyboard Event | 105 |
| Bab VIII Navigasi Halaman Interaktif Multimedia..... | 109 |
| Desain Halaman..... | 109 |
| Membuat tombol..... | 111 |
| Mengatur Halaman | 113 |
| Menambahkan label..... | 117 |
| Penambahan Instance Name | 118 |

| | |
|--|------------|
| Penambahan Actionscript untuk Navigasi | 118 |
| Membuat Transisi Halaman..... | 120 |
| Sub Menu (sub halaman) | 123 |
| Bab VIII Operasi Text..... | 127 |
| Scrolling Text..... | 127 |
| Scrolling Text menggunakan Component | 128 |
| Scrolling menggunakan Dynamic Text | 129 |
| Scrolling text dengan Gambar | 130 |
| Input Text..... | 133 |
| HTML Text Format | 135 |
| Bab IX Audio dan Video | 137 |
| Menambahkan Suara (Audio) | 137 |
| Memainkan File Suara di Luar Aplikasi | 139 |
| Menambahkan Video ke dalam Multimedia Pembelajaran | 141 |
| Menambahkan tombol Play alternatif | 144 |
| Bab X Aplikasi Kuis..... | 147 |
| Membuka data XML..... | 155 |
| Mengintegrasikan XML ke dalam aplikasi kuis | 156 |
| Bab XI Candy Eye (Pemanis Tampilan)..... | 161 |
| Mengubah kurSOR mouse..... | 161 |
| Efek Starburst (Percikan bintang) | 162 |
| Menampilkan Waktu..... | 164 |
| Membuka file SWF Eksternal..... | 167 |
| Bab XII Publishing | 169 |
| Mempublish Aplikasi untuk PC Desktop | 170 |
| Mempublish menjadi Aplikasi Mobile Android | 171 |
| Bab XIII Penutup..... | 175 |
| Daftar Pustaka | 176 |

Daftar Gambar

| | |
|--|----|
| Gambar 1. contoh media pembelajaran dengan aplikasi Power Point..... | 3 |
| Gambar 2. media pembelajaran berbasis komputer..... | 3 |
| Gambar 3. contoh media pembelajaran berbasis mobilephone | 10 |
| Gambar 4. Halaman depan Yummy Land “Dunia Buah dan Sayur” | 13 |
| Gambar 5. Pengenalan Buah dalam Taman Bermain | 13 |
| Gambar 6. Penjelasan audio dan video tentang buah jeruk | 14 |
| Gambar 7. Animasi kehidupan “buah-buahan” di dalam air | 14 |
| Gambar 8. Tampilan sayur dan buah dalam ekosistem danau | 15 |
| Gambar 9. Halaman permainan Interaktif | 16 |
| Gambar 10. Media pembelajaran “Mengenal pakaian dan rumah adat nusantara” .. | 16 |
| Gambar 11. Pulau interaktif | 17 |
| Gambar 12. Penjelasan baju dan rumah adat | 17 |
| Gambar 13. Foto pakaian dan rumah adat | 18 |
| Gambar 14. Permainan interaktif mencocokan pakaian adat | 18 |
| Gambar 15. Contoh flowchart multimedia pembelajaran interaktif | 20 |
| Gambar 16. Tahapan testing aplikasi | 21 |
| Gambar 17. Penggunaan proporsi tombol | 23 |
| Gambar 18. Keterbacaan yang kurang baik pada media pembelajaran | 23 |
| Gambar 19. Penekanan teks MENGAJI pada halaman judul | 24 |
| Gambar 20. Penekanan materi dengan cara membingkai materi | 24 |
| Gambar 21. Keseimbangan simetris pada desain media pembelajaran | 25 |
| Gambar 22. Keseimbangan asimetris pada desain media pembelajaran | 25 |
| Gambar 23. Unsur pengulangan membentuk ritme dan eyeflow | 26 |
| Gambar 24. Elemen yang selaras dari segi warna, gaya desain dan proporsi | 26 |
| Gambar 25. Kesatuan di tiap-tiap halaman media pembelajaran interaktif..... | 27 |
| Gambar 26. Pemanfaatan Flash dalam aplikasi Sekaran Ville | 30 |
| Gambar 27. Tampilan interface Adobe Flash CS5 | 30 |
| Gambar 28. Mengubah ke mode Classic | 31 |
| Gambar 29. Interface Adobe Flash dalam mode Classic | 31 |
| Gambar 30. Frame | 34 |
| Gambar 31. Keyframe | 34 |

| | |
|---|----|
| Gambar 32. Keyframe dan Blank Keyframe pada timeline | 35 |
| Gambar 33. Frame Per Second | 35 |
| Gambar 34. Onion Skin | 36 |
| Gambar 35. New File | 36 |
| Gambar 36. New Document AS3 | 37 |
| Gambar 37. Document Settings | 37 |
| Gambar 38. Brush tool..... | 38 |
| Gambar 39. Insert Blank Keyframe..... | 38 |
| Gambar 40. Onion skin dan hasilnya..... | 38 |
| Gambar 41. Hasil animasi frame by frame | 39 |
| Gambar 42. Menggeser Keyframe..... | 39 |
| Gambar 43. Insert Frame | 40 |
| Gambar 44. File gambar dengan penamaan berurutan | 41 |
| Gambar 45. Import to Stage | 41 |
| Gambar 46. Import sequence | 42 |
| Gambar 47. Animasi Frame by Frame hasil import | 42 |
| Gambar 48. Oval tool..... | 43 |
| Gambar 49. Convert to Symbol | 43 |
| Gambar 50. Insert Keyframe..... | 43 |
| Gambar 51. Peletakan objek di frame 2 | 44 |
| Gambar 52. Create Classic Tween..... | 44 |
| Gambar 53. Hasil animasi motion tween | 44 |
| Gambar 54. Animasi dengan ease..... | 45 |
| Gambar 55. Symbol “pendulum” | 46 |
| Gambar 56. New Layer..... | 46 |
| Gambar 57. Library..... | 47 |
| Gambar 58. Menambahkan symbol dari library ke stage | 47 |
| Gambar 59. Menggeser Pivot Point | 47 |
| Gambar 60. Posisi awal pendulum..... | 48 |
| Gambar 61. Insert Keyframe 40 – tanpa perubahan apapun..... | 48 |
| Gambar 62. Pengaturan Keyframe pendulum..... | 49 |
| Gambar 63. Menambahkan Classic Tween pada seluruh layer | 49 |
| Gambar 64. Pengaturan ease..... | 50 |

| | |
|---|----|
| Gambar 65. Perbedaan Animasi pendulum tanpa dan dengan ease | 50 |
| Gambar 66. Mengconvert Movieclip roda..... | 51 |
| Gambar 67. Pengaturan rotasi..... | 51 |
| Gambar 68. Hasil animasi rotasi..... | 51 |
| Gambar 69. Membuat gigi..... | 52 |
| Gambar 70. Drag titik pusat..... | 52 |
| Gambar 71. Scale and Rotate..... | 53 |
| Gambar 72. Tahapan membuat roda gigi | 53 |
| Gambar 73. Menyempurnakan roda gigi | 53 |
| Gambar 74. Pengaturan roda gigi | 54 |
| Gambar 75. Menambahkan animasi | 54 |
| Gambar 76. Mengimport gambar pesawat..... | 55 |
| Gambar 77. Mengconvert symbol pesawat | 55 |
| Gambar 78. Add Classic Motion Guide | 55 |
| Gambar 79. Membuat lintasan..... | 56 |
| Gambar 80. Drag ke awal dan ke akhir lintasan | 56 |
| Gambar 81. Animasi Motion Guide | 57 |
| Gambar 82. Penambahan Orient to Path..... | 57 |
| Gambar 83. Flip Horizontal..... | 58 |
| Gambar 84. Lintasan kupu-kupu | 58 |
| Gambar 85. Mode edit symbol “kupu-kupu” | 59 |
| Gambar 86. Animasi frame by frame kepakan kupu-kupu..... | 59 |
| Gambar 87. Import to Library | 60 |
| Gambar 88. Distribute to Layers | 60 |
| Gambar 89. Pengaturan Objek..... | 60 |
| Gambar 90. Mengconvert movieclip mobil..... | 61 |
| Gambar 91. Lintasan mobil di atas bukit..... | 61 |
| Gambar 92. Animasi mobil tanpa roda..... | 62 |
| Gambar 93. Movieclip animasi roda | 62 |
| Gambar 94. Membuat animasi roda..... | 62 |
| Gambar 95. Membuat rotasi roda | 63 |
| Gambar 96. Kembali ke mode edit mobil..... | 63 |
| Gambar 97. Pengaturan roda belakang | 63 |

| | |
|---|----|
| Gambar 98. Hasil animasi mobil | 63 |
| Gambar 99. Gerbong roller coaster..... | 65 |
| Gambar 100. Posisi dan convert to symbol graphic “animasi roller coaster” | 65 |
| Gambar 101. Mengconvert kembali gerbong menjadi movieclip | 65 |
| Gambar 102. Animasi motion guide roller coaster..... | 66 |
| Gambar 103. Pengaturan gerbong ke 2..... | 66 |
| Gambar 104. Hasil duplikasi graphic “animasi roller coaster”..... | 67 |
| Gambar 105. Penambahan rel (lintasan) | 67 |
| Gambar 106. Import to Stage..... | 68 |
| Gambar 107. Distribute to Layers..... | 68 |
| Gambar 108. Struktur layer..... | 68 |
| Gambar 109. Struktur layer..... | 69 |
| Gambar 110. Posisi bulan saat di depan dan di belakang bumi..... | 69 |
| Gambar 111. Masking..... | 70 |
| Gambar 112. Text Tool..... | 70 |
| Gambar 113. Posisi movieclip “kotak” terhadap teks | 71 |
| Gambar 114. Masking..... | 71 |
| Gambar 115. Hasil efek kilau dengan masking | 71 |
| Gambar 116. Peta dunia | 72 |
| Gambar 117. Pengaturan peta | 72 |
| Gambar 118. Bentuk dasar bumi | 72 |
| Gambar 119. Posisi map terhadap lingkaran | 73 |
| Gambar 120. Pengaturan warna gradasi | 73 |
| Gambar 121. PolyStar Tool | 74 |
| Gambar 122. Hasil animasi shape tween | 74 |
| Gambar 123. Panel Action..... | 76 |
| Gambar 124. Hasil pada output panel..... | 76 |
| Gambar 125. Tampilan Compiler Errors | 77 |
| Gambar 126. Peletakan kode stop();..... | 89 |
| Gambar 127. Interaktivitas sederhana..... | 93 |
| Gambar 128. Interaktivitas kompleks | 93 |
| Gambar 129. Brush tool..... | 94 |
| Gambar 130. Movieclip “animasi burung” | 94 |

| | |
|--|-----|
| Gambar 131. Instance name “burung” | 95 |
| Gambar 132. kordinat kartesian..... | 96 |
| Gambar 133. Export for Actionscript | 97 |
| Gambar 134. Linkage “bola_mc” pada library..... | 97 |
| Gambar 135. Pengaturan instance name..... | 99 |
| Gambar 136. Movieclip kepala dan badan panda..... | 100 |
| Gambar 137. Movieclip dengan instance name “burung” | 103 |
| Gambar 138. Movieclip mobil..... | 105 |
| Gambar 139. Rencana halaman interaktif multimedia | 109 |
| Gambar 140. Layout interface dengan sistem grid | 110 |
| Gambar 141. Contoh layout halaman depan dengan navigasi di atas | 110 |
| Gambar 142. Timeline symbol bertipe button | 111 |
| Gambar 143. Rounded rectangle | 111 |
| Gambar 144. Menambahkan teks pada tombol | 112 |
| Gambar 145. Pengaturan keyframe button “tombol” | 112 |
| Gambar 146. Properties dokumen | 113 |
| Gambar 147. Contoh halaman awal multimedia | 113 |
| Gambar 148. Position and size properties | 114 |
| Gambar 149. Titik registrasi movieclip konten | 114 |
| Gambar 150. Mode edit movieclip konten | 114 |
| Gambar 151. Susunan layer movieclip konten | 114 |
| Gambar 152. Tombol halaman depan dan posisinya terhadap background | 115 |
| Gambar 153. Duplicate Symbol | 115 |
| Gambar 154. Pengaturan posisi tombol..... | 115 |
| Gambar 155. Background di frame 10 | 116 |
| Gambar 156. Konten sementara untuk materi tata surya..... | 116 |
| Gambar 157. Frame label | 117 |
| Gambar 158. Label secara keseluruhan | 118 |
| Gambar 159. Instance name tombol_home | 118 |
| Gambar 160. Mengatur transisi | 120 |
| Gambar 161. Movieclip tutup atas dan tutup bawah | 123 |
| Gambar 162. Animasi tween pada movieclip transisi | 123 |
| Gambar 163. Posisi Movieclip materi dan tombol panah..... | 123 |

| | |
|--|-----|
| Gambar 164. Pengaturan materi dalam movieclip “materi” | 124 |
| Gambar 165. Menu components | 128 |
| Gambar 166. Componen parameter text area | 128 |
| Gambar 167. Pengaturan text properties..... | 129 |
| Gambar 168. Pengaturan teks dan tombol | 129 |
| Gambar 169. Movieclip konten scroll | 131 |
| Gambar 170. Mengconvert movieclip “info sepeda” | 131 |
| Gambar 171. Penambahan masking..... | 132 |
| Gambar 172. Pengaturan instance name | 132 |
| Gambar 173. Contoh penggunaan teks type | 134 |
| Gambar 174. Opsi Render Test as HTML | 135 |
| Gambar 175. Mengimport suara ke library | 138 |
| Gambar 176. Tombol play dan mute | 138 |
| Gambar 177. Mengatur linkage symbol suara | 138 |
| Gambar 178. Tombol pengatur suara..... | 139 |
| Gambar 179. Import video..... | 142 |
| Gambar 180. Panel Select video | 142 |
| Gambar 181. Skinning video | 142 |
| Gambar 182. Video yang berhasil di import ke dalam aplikasi..... | 143 |
| Gambar 183. Pengaturan properties component video | 143 |
| Gambar 184. Tombol play alternatif..... | 144 |
| Gambar 185. Mengatur tombol invisible | 144 |
| Gambar 186. Rencana layout kuis | 147 |
| Gambar 187. Pengaturan awal asset aplikasi kuis | 148 |
| Gambar 188. Pengaturan dynamic text..... | 148 |
| Gambar 189. Susunan movieclip “alternatif jawaban” | 149 |
| Gambar 190. Pengaturan instance name | 149 |
| Gambar 191. Pengaturan popup jawaban pada movieclip “hasil_mc” | 150 |
| Gambar 192. Struktur movieclip “hasil_mc”..... | 150 |
| Gambar 193. Contoh XML..... | 155 |
| Gambar 194. Animasi frame by frame movieclip tawon..... | 161 |
| Gambar 195. Movieclip “bintang_mc” | 163 |
| Gambar 196. Hasil efek percikan bintang..... | 164 |

| | |
|---|-----|
| Gambar 197. Tampilan waktu | 165 |
| Gambar 198. Pengaturan jarum jam | 165 |
| Gambar 199. Tampilan jarum jam..... | 166 |
| Gambar 200. Hasil aplikasi membuka file SWF | 168 |
| Gambar 201. Publish settings | 170 |
| Gambar 202. Hasil proses publishing bertipe exe | 170 |
| Gambar 203. Target AIR for Android | 171 |
| Gambar 204. Tombol setting | 172 |
| Gambar 205. General Setting | 172 |
| Gambar 206. Create certificate | 172 |
| Gambar 207. Panel Create Certificate | 173 |
| Gambar 208. Deployment setting | 173 |
| Gambar 209. Icon aplikasi | 174 |
| Gambar 210. File APK hasil proses publishing..... | 174 |
| Gambar 211. Ujicoba media dengan perangkat berbasis Android | 174 |

Bab I

Pendahuluan

Perkembangan teknologi informasi mendorong perubahan yang besar di berbagai aspek kehidupan. Dalam ranah pembelajaran di sekolah, terjadi perubahan dan pergeseran paradigma pendidikan dimana perkembangan pesat dibidang teknologi informasi tersebut, mempercepat aliran ilmu pengetahuan yang menembus batas-batas dimensi ruang, birokrasi, kemapanan, dan waktu. Teknologi informasi dapat menampilkan data dan mentransfer informasi dengan kecepatan tinggi, menyebabkan ilmu pengetahuan dapat diakses secara cepat oleh penggunanya. Tentu saja kondisi ini berpengaruh pada kebiasaan dan budaya pendidikan yang dikelola selama ini.

Perkembangan ilmu pengetahuan dan teknologi tersebut semakin mendorong upaya-upaya pembaharuan dalam pemanfaatan hasil-hasil teknologi dalam proses belajar mengajar. Para guru dituntut agar mampu menggunakan alat-alat yang dapat disediakan oleh sekolah, dan tidak tertutup kemungkinan bahwa alat-alat tersebut sesuai dengan perkembangan dan tuntutan zaman. Guru sekurang-kurangnya dapat menggunakan alat yang murah dan bersahaja tetapi merupakan keharusan dalam upaya mencapai tujuan pengajaran yang diharapkan. Disamping mampu menggunakan alat-alat yang tersedia, guru juga dituntut untuk dapat mengembangkan keterampilan membuat media pengajaran yang dapat digunakan untuk mengoptimalkan pencapaian tujuan pembelajaran.

Pembelajaran adalah perpaduan dari dua aktivitas, yaitu aktivitas mengajar dan aktivitas belajar. Aktivitas mengajar menyangkut peranan seorang guru dalam konteks mengupayakan terciptanya jalinan komunikasi harmonis antara pengajar itu sendiri dengan si belajar. Menurut Sugandi (2000) ciri-ciri dari pembelajaran antara lain:

1. Pembelajaran dilakukan secara sadar dan direncanakan secara sistematis;
2. Pembelajaran dapat menumbuhkan perhatian dan motivasi siswa dalam belajar;
3. Pembelajaran dapat menyediakan bahan belajar yang menarik dan menantang bagi siswa;
4. Pembelajaran dapat menggunakan alat bantu belajar yang tepat dan menarik;

5. Pembelajaran dapat menciptakan suasana belajar yang aman dan menyenangkan bagi siswa;
6. Pembelajaran dapat membuat siswa siap menerima pelajaran baik secara fisik maupun psikologis.

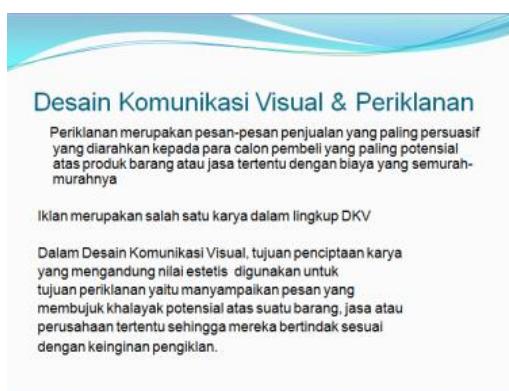
Dalam pembelajaran komponen terpenting adalah pendidik dan peserta didik yang selalu berinteraksi dalam proses belajar mengajar. Dari komponen pelajar meliputi tingkat perkembangan, tingkat kesiapannya, minatnya, aspirasi dan sebagainya. Dari komponen pengajar, meliputi tingkat kemampuan, minat, waktu, wibawa, status, dan sebagainya. Dari komponen interaksi meliputi isi interaksi, apa yang dilakukan pelajar, alat-alat yang dipakai, metode yang dipergunakan dalam mengajar, sikap belajar yang tumbuh pada pelajar sebagai hasil interaksi belajar mengajar.

Dalam belajar mengajar hal yang harus diutamakan adalah proses, karena proses inilah yang menentukan tujuan belajar akan tercapai atau tidak tercapai. Ketercapaian dalam proses belajar mengajar ditandai dengan adanya perubahan tingkah laku. Perubahan tingkah laku tersebut baik yang menyangkut perubahan bersifat pengetahuan (kognitif), keterampilan (psikomotor) maupun yang menyangkut nilai dan sikap (afektif). Seringkali terjadi dalam proses belajar mengajar yang hanya berorientasi pada hasil dengan mengabaikan proses, hal ini tentunya akan mengakibatkan kualitas pendidikan yang sesungguhnya terabaikan. Dalam proses belajar mengajar ada banyak faktor yang mempengaruhi tercapainya tujuan pembelajaran diantaranya pendidik, peserta didik, lingkungan, metode/teknik serta media/alat pembelajaran.

Dalam kaitannya dengan usaha untuk mencapai tujuan pembelajaran, media pembelajaran mempunyai peran yang sangat penting. Media pembelajaran merupakan sarana yang dapat membantu proses pembelajaran karena berkaitan dengan indera pendengaran dan penglihatan. Adanya media pembelajaran bahkan dapat mempercepat proses belajar mengajar menjadi efektif dan efisien dalam suasana yang kondusif, sehingga dapat membuat pemahaman peserta didik lebih cepat. Dengan adanya media pembelajaran maka tradisi lisan dan tulisan dalam proses pembelajaran dapat diperkaya dengan berbagai media pengajaran. Selanjutnya, guru pendidik dapat menciptakan berbagai situasi kelas, menentukan metode pengajaran yang akan dipakai dalam situasi yang berlainan dan menciptakan iklim yang emosional yang sehat diantara peserta didik.

Lebih lanjut, media pembelajaran dapat membantu guru membawa dunia luar ke dalam kelas. Dengan demikian ide yang abstrak dan asing (remote) sifatnya menjadi konkret dan mudah dimengerti oleh peserta didik. Bila media pembelajaran ini dapat di fungsikan secara tepat dan proforsional, maka proses pembelajaran akan dapat berjalan efektif.

Permasalahan yang terjadi di lapangan adalah banyak pendidik yang belum mampu mengoptimalkan teknologi yang ada untuk mewujudkan sebuah pembelajaran yang melibatkan media pembelajaran berbasis teknologi seperti komputer. Sebagian besar pendidik terkendala permasalahan teknis tentang prosedur pembuatan media pembelajaran elektronik baik dari ranah penguasaan teknik pemrograman, maupun tampilan visual atau desain. Pada akhirnya media pembelajaran berbasis komputer hanya sebatas pemindahan media kertas ke media digital dengan tampilan yang sama, dan interaktifitas yang relatif minim. Hal ini yang menyebabkan media pembelajaran menjadi kurang optimal



Gambar 1. contoh media pembelajaran dengan aplikasi Power Point



Gambar 2. media pembelajaran berbasi komputer
(Patriani, 2012)

Sebagai salah satu solusi yang ditawarkan atas permasalahan tersebut, buku ini akan memfokuskan pembahasan terkait proses pembuatan media pembelajaran interaktif dengan menggunakan aplikasi Adobe Flash. Pembahasan dalam buku ini meliputi proses membangun interaktifitas, penggunaan animasi, audio dan video, serta pemanfaatan kode actionscript dalam aplikasi permainan edukatif.

Bab II

Media Pembelajaran

Kata media berasal dari bahasa latin medius yang secara harfiah berarti “tengah”, “perantara”, atau “pengantar”. Dalam bahasa Arab, media adalah perantara atau pengantar pesan dari pengirim kepada penerima pesan (Arsyad, 2011). Pada pengertian lain, media adalah segala sesuatu yang dapat digunakan untuk menyampaikan pesan dari pengirim ke penerima pesan (Prawiradilaga, 2008). Menurut Gagne (1992), media adalah berbagai jenis komponen dalam lingkungan siswa yang dapat merangsangnya untuk belajar. Senada dengan pendapat Gagne, Briggs mendefinisikan media pembelajaran sebagai bentuk fisik yang dapat menyajikan pesan yang dapat merangsang siswa untuk belajar. Dari dua definisi ini tampak pengertian media mengacu pada penggunaan alat yang berupa benda untuk membantu proses penyampaian pesan. Sementara itu Asosiasi Pendidikan Nasional (Nation Education Association/ NEA) memberikan batasan tentang media yaitu bentuk-bentuk komunikasi baik tercetak maupun audi visual serta berbagai peralatannya. Media pendidikan adalah sumber belajar dan dapat juga diartikan dengan manusia dan benda atau peristiwa yang membuat kondisi siswa mungkin memperoleh pengetahuan, keterampilan atau sikap. Selain alat yang berupa benda, yang digunakan untuk menyalurkan pesan dalam proses pendidikan, pendidikan sebagai figur sentral atau model dalam proses interaksi edukatif merupakan alat pendidikan yang juga harus di perhitungkan.

Menurut Oemar Hamalik (1986), media pembelajaran adalah hubungan komunikasi interaksi akan berjalan lancar dan tercapainya hasil yang maksimal, apabila menggunakan alat bantu yang disebut media komunikasi. Dalam pengertian komunikasi, media adalah alat yang memindahkan informasi (pesan) dari sumber kepada penerima. Menurut Edgar Dale (1946), media pembelajaran merupakan klasifikasi pengalaman menurut tingkat dari yang paling konkret ke yang paling abstrak, dimana partisipasi, observasi, dan pengalaman langsung memberikan pengaruh yang sangat besar terhadap pengalaman belajar yang diterima siswa. Penyampaian suatu konsep pada siswa akan tersampaikan dengan baik jika konsep tersebut mengharuskan siswa terlibat langsung didalamnya bila dibandingkan dengan konsep yang hanya melibatkan siswa untuk mengamati saja. Sedangkan menurut Newby (2000), media pembelajaran adalah media

yang dapat menyampaikan pesan pembelajaran atau mengandung muatan untuk membelajarkan seseorang. Dari beberapa pendapat tersebut dapat disimpulkan bahwa media pembelajaran adalah media kreatif yang digunakan dalam memberikan materi pelajaran kepada anak didik sehingga proses belajar mengajar lebih efektif, efisien dan menyenangkan.

Manfaat Media Pembelajaran

Media pembelajaran memiliki peranan yang besar dan berpengaruh terhadap pencapaian tujuan pendidikan yang diinginkan. Kegunaan Media/ alat pembelajaran dalam proses belajar mengajar diantaranya:

1. Memperjelas penyajian pesan supaya tidak terlalu verbalitas (dalam bentuk kata-kata tertulis atau hanya kata lisan)
2. Mengatasi keterbatasan ruang, waktu dan daya indera, misalnya:
 - Objek yang terlalu besar – bisa digantikan dengan realita, gambar,film bingkai, film, atau model.
 - Objek yang kecil – dibantu dengan proyektor mikro, film bingkai, film, atau gambar.
 - Gerak yang terlalu lambat atau terlalu cepat, dapat dibantu dengan *timelapse* atau *high-speed photography*.
 - Kejadian atau peristiwa yang terjadi dimasa lalu bisa ditampilkan lagi lewat rekaman film, video, film bingkai, atau foto objek yang terlalu kompleks, dapat disajikan dengan model, diagram atau melalui program komputer animasi.
 - Konsep yang terlalu luas (gempa bumi, gunung berapi, iklim, planet dan lain-lain) dapat divisualisasikan dalam bentuk film, gambar dan lain-lain.
3. Dengan menggunakan media pendidikan secara tepat dan bervariasi dapat diatasi sikap pasif anak didik. Dalam hal ini media pembelajaran berguna untuk menimbulkan motivasi belajar, memungkinkan interaksi langsung antara anak didik dengan lingkungan secara seperti senyatanya, memungkinkan peserta didik belajar mandiri sesuai dengan kemampuan dan minatnya.
4. Dengan latar belakang dan pengalaman yang berbeda diantara peserta didik, sementara kurikulum dan materi pelajaran di tentukan sama untuk semua peserta

didik dapat diatasi dengan media pendidikan yaitu : memberikan perangsang yang sama, mempersamakan pengalaman, menimbulkan persepsi yang sama.

Selain itu beberapa pakar berpendapat bahwa kegunaan media pembelajaran itu antara lain adalah :

1. mampu mengatasi kesulitan-kesulitan dan memperjelas materi pelajaran yang sulit,
2. mampu mempermudah pemahaman dan menjadikan pelajaran lebih hidup dan menarik,
3. merangsang anak untuk bekerja dan menggerakkan naluri kecintaan menelaah (belajar) dan menimbulkan
4. kemauan keras untuk mempelajari sesuatu,
5. membantu pembentukan kebiasaan, melahirkan pendapat, memperhatikan dan memikirkan suatu pelajaran serta,
6. menimbulkan kekuatan perhatian (ingatan) mempertajam indera, melatihnya, memperluas perasaan dan kecepatan dalam belajar.

Jenis Media Pembelajaran

Rudi Bretz mengidentifikasi ciri utama dari media menjadi tiga unsur pokok yaitu suara, visual dan gerak. Visual sendiri di bedakan menjadi tiga yaitu gambar, garis (*line graphic*) dan simbol yang merupakan suatu *kontinum* dari bentuk yang dapat di tangkap dengan indera penglihatan. Disamping itu Bretz juga membedakan antara media siar (*telecommunication*) dan media rekam (*recording*) sehingga ada 8 klasifikasi media, yaitu:

1. media audio visual gerak
2. media audio visual diam
3. media audio semi gerak
4. media visual gerak
5. media visual diam
6. media semi gerak
7. media audio
8. media cetak.

Sementara menurut Gagne (1992) mengelompokan media dalam 7 macam yaitu;

1. benda untuk didemonstrasikan
2. komunikasi lisan
3. media cetak
4. gambar diam
5. gambar gerak
6. film bersuara dan
7. mesin belajar.

Para ahli telah mengklasifikasikan media pendidikan kepada dua bagian yaitu alat pendidikan yang bersifat benda (materil) dan alat pendidikan yang bukan benda (non materil). Oemar Hamalik menyebutkan, secara umum alat pendidikan materil terdiri dari: pertama, bahan-bahan cetakan atau bacaan, dimana bahan-bahan ini lebih mengutamakan kegiatan membaca atau penggunaan simbol-simbol kata dan visual. Kedua, alat-alat audio visual yakni alat-alat yang dapat digolongkan pada:

1. alat tanpa proyeksi seperti papan tulis dan diagram
2. media pendidikan tiga dimensi, seperti benda asli, peta
3. alat pendidikan yang menggunakan teknik seperti radio, tape recorder, transparansi.

Ketiga, sumber-sumber masyarakat, seperti peninggalan sejarah. Keempat,kumpulan benda-benda (*material collection*), seperti dedaunan, benih, batu, dan sebagainya.

Menurut Ronald H. Anderson yang termasuk media dalam bentuk materil adalah media auditif, dimana pesan-pesan pengajaran dituangkan dalam lambang-lambang auditif, yang termasuk media auditif adalah, tape recorder dan radio. Disamping media visual dan media auditif, media audio visual merupakan media yang berhubungan dengan indera pendengaran dan indera penglihatan sekaligus.

Perkembangan media pembelajaran seiring dengan perkembangan teknologi. Seels dan Richey (dalam Azhar Arsyad, 2006) membagi media pembelajaran dalam empat kelompok berdasarkan perkembangan teknologi, yaitu:

a. Media hasil teknologi cetak

Media hasil teknologi cetak adalah cara untuk menghasilkan atau menyampaikan materi melalui proses pencetakan mekanis atau fotografis. Kelompok media hasil teknologi cetak meliputi teks, grafik, foto, dan representasi fotografik. Materi cetak dan visual merupakan pengembangan dan penggunaan kebanyakan materi pengajaran lainnya. Teknologi ini menghasilkan materi dalam bentuk salinan tercetak, contohnya buku teks, modul, majalah, hand-out, dan lainlain.

b. Media hasil teknologi audio-visual

Media hasil teknologi audio-visual menghasilkan atau menyampaikan materi dengan menggunakan mesin-mesin mekanis dan elektronik untuk menyajikan pesan-pesan audio dan visual. Contohnya proyektor film, televisi, video, dan sebagainya.

c. Media hasil teknologi berbasis komputer

Media hasil teknologi berbasis komputer merupakan cara menghasilkan atau menyampaikan materi dengan menggunakan sumber-sumber yang berbasis mikro-prosesor. Berbagai jenis aplikasi teknologi berbasis komputer dalam pengajaran umumnya dikenal sebagai computer-assisted instruction (pengajaran dengan bantuan komputer).

d. Media hasil teknologi gabungan

Media hasil teknologi gabungan adalah cara untuk menghasilkan atau menyampaikan materi yang menggabungkan beberapa bentuk media yang dikendalikan oleh komputer. Perpaduan beberapa teknologi ini dianggap teknik yang paling canggih. Contohnya: teleconference, realitas maya (*virtual reality*)

Pada tahun 2010 perkembangan teknologi digital menuju ke arah *mobile* (gadget), sebagian peserta didik memiliki gadget berupa *smartphone* yang merupakan media efektif dalam menyampaikan suatu informasi. Dari data yang dikumpulkan oleh *WaiWai Marketing* pada tahun 2014, pengguna mobile phone berbasis *android* di Indonesia adalah yang tertinggi di Asia Tenggara yaitu 44 juta pengguna. Dari jumlah tersebut, 11% adalah pengguna handphone berusia antara 7 - 17 tahun, yang merupakan usia aktif

peserta didik. Berkembangnya media mobile phone telah dimanfaatkan oleh beberapa pihak sebagai salah satu media pembelajaran, sehingga menjadikan klasifikasi atau jenis media pembelajaran bertambah. Meskipun beberapa pakar mengklasifikasikan media pembelajaran berbasis *mobilephone* masuk ke dalam kategori pembelajaran berbasis komputer (CAI), namun demikian dalam buku ini pembelajaran berbasis mobile phone diklasifikasikan secara tersendiri karena memiliki perbedaan yang signifikan dibandingkan dengan pembelajaran berbasis komputer.



(Pakaian dan rumah adat nusantara, Murtiningsih : 2013)



(Yummyland – mengenalkan sayur pada balita, Laras Safitri: 2012)

Gambar 3. contoh media pembelajaran berbasis komputer dan berbasis mobilephone

Bab III

Desain Media Pembelajaran

Pada bab I, sempat disinggung bahwa kelemahan beberapa media pembelajaran berbasis komputer yang sudah ada adalah sifat dari media tersebut yang hanya memindah materi di dalam buku ke layar komputer. Layar pada akhirnya berisi penuh dengan materi yang harus dibaca oleh peserta didik, yang mana menurut Edgar Dale (1946) efektifitas membaca sebuah materi hanya berada dikisaran 10%. Berbeda halnya jika sebuah media pembelajaran dibuat dengan desain yang baik, dengan melibatkan grafis, audio, video dan interaktifitas akan menaikkan efektifitas penyerapan materi hingga 80 – 90%.

Pentingnya sebuah desain yang baik dalam sebuah media pembelajaran terkadang terbentur dengan pengetahuan dasar desain tenaga pengajar yang membuat media tersebut. Pada umumnya tenaga pengajar menguasai materi yang diajarkannya, akan tetapi pengetahuan dasar tentang desain yang baik, komposisi, tata letak, warna dan elemen kesenirupaan jarang dikuasai. Hal inilah yang menyebabkan media pembelajaran pada akhirnya menjadi tidak optimal dalam menyampaikan materinya.

Ketika kita menggunakan istilah “desain”, maka dalam teori desain kita harus melakukan beberapa langkah yaitu: design treatment, visual development phase, programming phase, testing phase.

Design Treatment

Design treatment merupakan sebuah istilah desain yang mengandung pengertian analisis tentang produk apa yang akan dibuat dan apa yang dibutuhkan untuk membuatnya. Pada tahapan ini sebuah media pembelajaran harus melalui sebuah proses analisis secara mendalam tentang apa saja yang terkait dengan media tersebut diantaranya adalah :

- Materi apa sajakah yang akan disampaikan dalam media tersebut. Pembatasan materi mutlak diperlukan untuk menjaga media agar fokus mencapai tujuan yang diharapkan.
- Siapakah target audience media yang akan dibuat, apakah balita, anak-anak, remaja atau orang dewasa. Penentuan target audience digunakan untuk menetapkan pengayaan desain, kekompleksan navigasi, dan interaktifitas.

- Fitur-fitur apa saja yang akan ditampilkan. Apakah media nantinya menampilkan sebatas materi, atau terdapat simulasi, evaluasi dalam bentuk kuis, penyimpanan data evaluasi dan sebagainya.
- Bagaimana dengan media sejenis yang sudah ada sebelumnya. Jika ada, apa yang membedakan media yang akan dibuat dengan media yang sudah ada. Selain itu juga perlu dipikirkan fitur apa yang akan menjadi nilai tambah dari media tersebut, sehingga penggunanya mendapatkan manfaat khusus ketika menggunakan media tersebut.
- Berapakah lamanya waktu untuk mewujudkan media tersebut dan aplikasi yang akan digunakan untuk mewujudkannya.

Semua hal yang terkait dengan ide yang akan kita wujudkan perlu didokumentasikan dalam bentuk catatan atau sebuah sket gambar untuk mempermudah gambaran anda atau tim anda dalam mewujudkannya.

Sebagai contoh design treatment, Laras Safitri (2012) mendesain sebuah media pembelajaran “Pengenalan Sayur dan Buah” pada anak usia dini. Ketika berbicara target audiense dari media yang akan dibuat yaitu anak usia dini, maka harus didefinisikan terlebih dahulu sifat-sifat dari target tersebut yaitu suka sesuatu yang berwarna cerah, sebagian besar belum bisa membaca namun aktif mendengar (dalam hal ini audio akan berperan penting), menyukai benda-benda yang bergerak dan dapat mengoperasikan media dengan cara yang paling sederhana yaitu dengan menekan layar (touch).

Maka desain yang ditampilkan adalah sebuah layar dengan gambar buah dan sayur yang dipersonalifikasi, dalam artian buah dan sayur seolah-olah hidup dan bergerak bebas di alam. Ketika anak menyentuh salah satu objek, maka akan muncul penjelasan secara audio pada objek tersebut.



Gambar 4. Halaman depan Yummy Land “Dunia Buah dan Sayur”



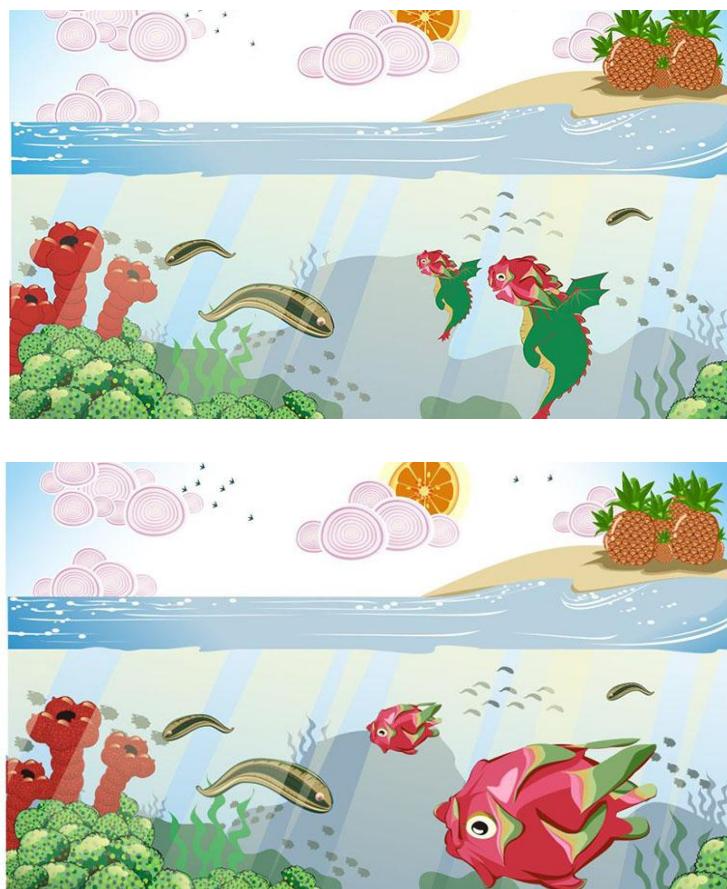
Gambar 5. Pengenalan Buah dalam Taman Bermain

Pada gambar 5, beberapa buah ditampilkan dalam sebuah taman bermain yang bergerak (beranimasi), pepaya ditampilkan seolah-olah kapal kora-kora, buah jeruk seolah-olah adalah bianglala, buah semangka seolah-olah pulau fantasi yang melayang di angkasa dan dengan irungan audio yang sesuai akan menjadikan target audiense menjadi tertarik untuk melihat sampai menyentuh objek. Ketika objek disentuh, maka akan muncul penjelasan secara audio dilengkapi dengan beberapa kalimat pendek bagi audiense yang sudah bisa membaca.



Gambar 6. Penjelasan audio dan video tentang buah jeruk

Dalam gambar di atas ditampilkan sebuah screenshot ketika objek jeruk ditekan. Tampilan audio, grafis dari buah jeruk dan video tentang buah jeruk ditampilkan. Audiense dapat menekan tombol kembali, atau aplikasi akan kembali ke tampilan sebelumnya jika penjelasan sudah selesai.



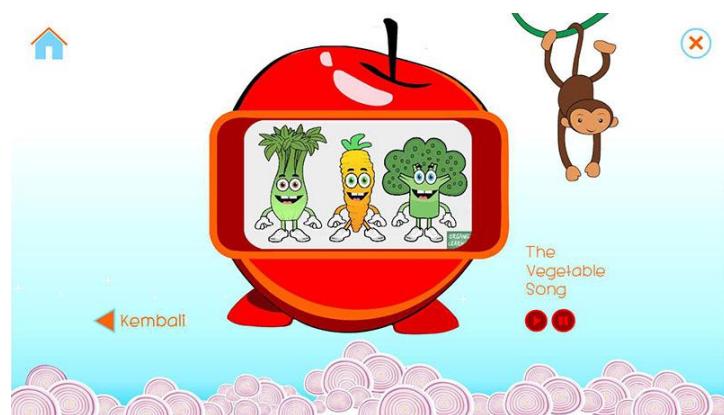
Gambar 7. Animasi kehidupan “buah-buahan” di dalam air

Pada tampilan berikutnya pengenalan buah dan sayur dibuat seolah-olah buah-buahan hidup di dalam ekosistem air. Buah naga ditampilkan seolah-olah naga air yang sedang berenang. Buah nanas sebagai pepohonan di sebuah pulau. Sayur kacang-kacangan berenang bebas. Beberapa objek ditampilkan dalam format animasi berulang, hal ini akan menimbulkan atensi pada audiense sehingga keinginan untuk mempelajari lebih lanjut semakin besar.



Gambar 8. Tampilan sayur dan buah dalam ekosistem danau

Sebuah media pembelajaran interaktif memiliki kemampuan untuk diulang-ulang, namun demikian beberapa media sering kali dimainkan satu kali saja oleh audiense karena merasa sudah cukup. Agar media memiliki nilai pengulangan yang tinggi (dalam artian audiense tidak cepat bosan), maka perlu ditambahkan kreatifitas dalam menyampaikan materi dan visualisasi yang beragam. Dalam Yummy Land tersebut beberapa animasi buah-buahan dan sayuran ditampilkan dalam beberapa ekosistem, sehingga audiense didorong untuk melakukan eksplorasi secara menyeluruh.





Gambar 9. Halaman permainan Interaktif

Untuk lebih menarik audiense yang dalam hal ini adalah anak-anak, ditampilkan pula permainan interaktif sederhana seperti puzzle buah dan sayur serta permainan menebak gambar.

Contoh ke 2 design treatment adalah media interaktif pembelajaran “Mengenal Pakaian dan Rumah Adat Nusantara” karya Murtiningsih (2012). Media ini ditargetkan untuk anak usia sekolah dasar dan menengah pertama, sehingga dari sisi grafis dan interaktifitas jauh lebih kompleks jika dibandingkan dengan contoh pertama.



Gambar 10. Media pembelajaran
“Mengenal pakaian dan rumah adat nusantara”

Dalam media tersebut, pada awalnya audiense akan dihadapkan dengan sebuah tampilan depan dengan 2 menu yaitu “belajar” dan “bermain”, menu belajar akan membawa audiense ke materi interaktif pengenalan pakaian dan rumah adat nusantara, sedang menu bermain akan membawa ke permainan interaktif untuk menguji kemampuan audiense.



Gambar 11. Pulau interaktif

Berbeda dengan interaktifitas yang sederhana pada media untuk anak usia dini, audiense yang lebih besar mampu mengoperasikan media yang lebih kompleks. Dalam hal ini Nusantara ditampilkan dalam masing-masing pulau, dan audiense dapat memilih lokasi berdasarkan propinsi yang ingin diketahui pakaian atau rumah adatnya. Dalam tampilan ini secara tidak langsung audiense juga belajar tentang geografis Nusantara.



Gambar 12. Penjelasan baju dan rumah adat

Selanjutnya audiense akan diberikan halaman penjelasan tentang pakaian dan rumah adat provinsi yang dipilih. Gambar yang ditampilkan merupakan gambar terolah yang sudah digayakan sesuai dengan audiense, namun demikian gambar asli juga diperlukan untuk

memberikan penjelasan lebih lanjut, sehingga diberikan sebuah tombol navigasi yang menunjukkan foto asli dari pakaian dan rumah adatnya.



Gambar 13. Foto pakaian dan rumah adat

Pada menu bermain ditampilkan permainan “*Drag and drop matching*”, yaitu mencocokan pakaian adat. Karena audiense dianggap sudah mampu mengoperasikan mouse atau touch dalam mode drag, maka permainan dapat menggunakan opsi tersebut. Berbeda dengan permainan anak usia dini yang cukup hanya dengan menyentuh (touch).

Permainan dalam media pembelajaran selain dapat menambah daya tarik bagi audiense, juga dapat digunakan sebagai bahan evaluasi untuk menilai pencapaian audiense dalam memahami materi yang ada di dalam media. Dengan menambahkan sistem scoring (penilaian) kita dapat mencatat dengan baik pencapaian masing-masing target audiense.



Gambar 14. Permainan interaktif mencocokan pakaian adat

Visual Development Phase

Pada tahapan ini aset-aset visual dibuat. Aset visual meliputi grafik yang akan dipakai sebagai objek dalam media, efek visual, typografi (penggunaan huruf) dan grafik untuk keperluan interface (antar muka). Pada umumnya tahapan ini lebih mudah dilakukan oleh desainer grafis, karena membutuhkan skill grafis dan kepekaan estetik. Sebuah media pembelajaran interaktif merupakan produk dari multidisiplin ilmu, sehingga apabila pembuat media pembelajaran hanya menguasai pembuatan dan pengolahan materi, maka dibutuhkan tenaga grafis untuk mengerjakan tahapan ini.

Beberapa aplikasi yang dapat dimanfaatkan untuk membangun grafis dari sebuah multimedia pembelajaran diantaranya adalah Photoshop, Corel Draw, Adobe Ilustrator, Adobe Flash dan sejenisnya. Penguasaan teknis terhadap aplikasi tersebut diperlukan agar tampilan multimedia pembelajaran yang dibuat menjadi lebih atraktif dan efisien dari sisi ukuran file.

Pada saat ini terdapat beberapa situs yang menyediakan asset grafis secara gratis maupun berbayar. Penggunaan asset yang sudah tersedia secara online dapat menghemat waktu dan biaya, namun demikian perlu diperhatikan beberapa aturan agar media yang dihasilkan nantinya menjadi lebih optimal, yaitu :

1. Gambar yang tersedia di internet memiliki lisensi yang bermacam-macam, mulai dari 100% gratis, lisensi nama pembuat dan lisensi berbayar. Sebelum memanfaatkan gambar tersebut kita harus memastikan dulu tentang lisensinya.
2. Gambar yang memiliki “watermark” menandakan bahwa gambar tersebut berlisensi dan tidak boleh dipakai tanpa ijin.
3. Gambar yang tersedia secara online memiliki variasi tipe grafis. Pada dasarnya tipe grafis ada 2 jenis yaitu vector dan bitmap. Vector merupakan gambar yang terbentuk atas beberapa kurva, sedangkan bitmap terbentuk atas titik-titik yang berwarna. Karena terdapat perbedaan yang signifikan antara gambar bertipe vektor dan bitmap, maka diperlukan kemampuan dalam mengolahnya.
4. Setiap gambar yang tersedia secara online memiliki ukuran tertentu. Untuk gambar bertipe bitmap, harus dilakukan penyesuaian ukuran sebelum gambar dipakai dalam media. Gambar yang terlalu kecil akan terdistorsi jika diperbesar,

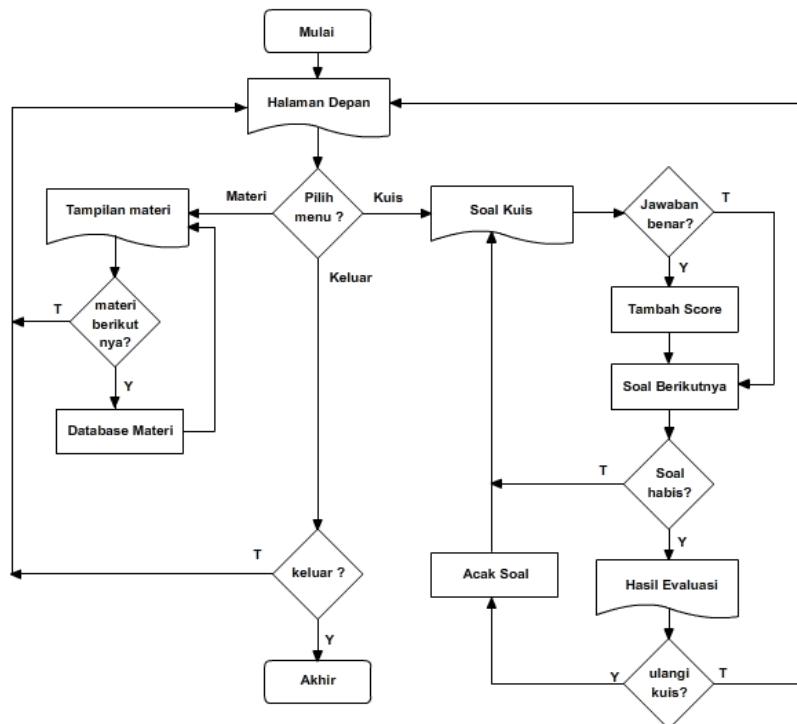
dan sebaliknya gambar yang terlalu besar akan memakan banyak memori dan ukuran file sehingga tidak efektif.

5. Penggunaan gambar yang bersumber secara online, memungkinkan sebuah objek ditampilkan secara berulang-ulang di media lain. Sehingga kesan eksklusif sebuah media menjadi berkurang.

Programming Phase

Pemrograman diperlukan dalam sebuah aplikasi media pembelajaran interaktif. Pekerjaan pemrograman membutuhkan ketelitian dan pemahaman atas logika. Pada umumnya kegiatan pemrograman dapat dilakukan dengan mudah oleh programmer, namun khusus untuk pemrograman multimedia pembelajaran dapat dilakukan oleh non programer karena tidak membutuhkan kode yang terlalu panjang.

Untuk mempermudah proses pemrograman dapat digunakan sistem flowchart. Flowchart adalah suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses (instruksi) dengan proses lainnya dalam suatu program. Perhatikan contoh flowchart berikut :

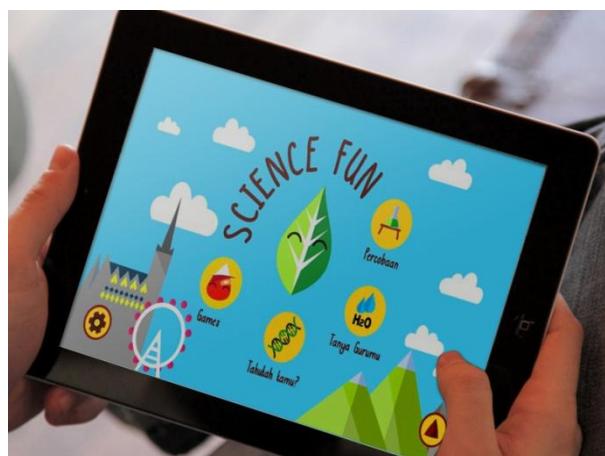


Gambar 15. Contoh flowchart multimedia pembelajaran interaktif

Testing Phase

Tahapan testing adalah sebuah proses pengumpulan informasi untuk memperbaiki segala kesalahan desain dan kesalahan programming. Tahapan ini diperlukan sebagai bentuk quality control dari media pembelajaran yang dibuat. Dari tahapan ini akan diketahui berbagai kesalahan, kesalahan penulisan materi, kesalahan logika, kesalahan ukuran interface dan sebagainya.

Tahapan testing lebih baik dilakukan oleh beberapa orang dengan tingkat keahlian yang berbeda, sehingga akan didapatkan masukan dari berbagai sudut pandang, apakah media pembelajaran sudah sesuai atau belum. Dari masukan-masukan tersebut dilakukan perbaikan ulang, untuk memperoleh hasil akhir yang benar-benar siap untuk dipublikasikan.



Gambar 16. Tahapan testing aplikasi Science Fun (Argani, 2016)
dalam mode mobile/tablet

Prinsip Desain Media Pembelajaran

Di dalam unsur-unsur senirupa dikenal adanya garis, raut, warna, ruang, tekstur dan gelap terang. Unsur tersebut juga muncul dalam sebuah desain media pembelajaran interaktif, sehingga kita dapat mengaitkan prinsip desain/seni rupa pada desain media pembelajaran. Beberapa prinsip yang dapat diaplikasikan dalam desain media pembelajaran antara lain :

1. Prinsip Kesebandingan (Proporsi)

Kesebandingan (proporsi) merupakan hubungan perbandingan antara bagian dengan bagian lain atau bagian dengan elemen keseluruhan. Kesebandingan dapat dijangkau dengan menunjukkan hubungan antara:

- Suatu elemen dengan elemen yang lain,
- Elemen bidang/ ruang dengan dimensi bidang atau ruangnya.
- Dimensi bidang/ruang itu sendiri.

Dalam sebuah media pembelajaran interaktif terdapat beberapa elemen. Secara umum elemen yang terdapat pada pembelajaran interaktif antara lain :

- Antarmuka (*User Interface*) meliputi : tombol navigasi, teks, dan elemen grafis di luar konten.
- Konten, merupakan materi utama dalam media pembelajaran
- Audio
- Video/animasi

Setiap elemen tersebut ditampilkan dalam proporsi yang berbeda. Untuk memenuhi prinsip kesebandingan, harus diperhatikan ukuran output yang ingin dihasilkan. Untuk output aplikasi desktop (komputer) ukuran elemen navigasi dapat dibuat antara 10-20% dari ukuran layar, dan materi yang ditampilkan berukuran 60 – 70%. Berbeda dengan output untuk smartphone (handphone), dimana ukuran yang lebih kecil menuntut proporsi navigasi yang lebih besar yaitu 15 – 25 % dari ukuran layar.



Gambar 17. Penggunaan proporsi tombol dalam pembelajaran rambu lalulintas untuk handphone (Trifandianto, 2016)

Prinsip utama dalam proporsi adalah setiap elemen dapat difungsikan dengan baik. Misal, elemen teks diatur sedemikian rupa dari sisi ukuran dan jenis huruf agar keterbacaannya tinggi.



Gambar 18. Keterbacaan yang kurang baik pada media pembelajaran berbasis mobile/handphone
(Trifandianto, 2016)

2. Prinsip Penekanan (Emphasis)

Di dalam sebuah karya desain/seni terdapat beberapa bahan atau gagasan yang lebih perlu ditampilkan dari pada yang lain. Dalam penciptaan desain/seni tidak seharusnya elemen yang ada menonjol semuanya, dalam artian sama kuatnya, sehingga terlihat ramai dan informasi atau apa yang akan disampaikan akan menjadi tidak jelas.

Tampilnya emphasis merupakan strategi komunikasi yang bertujuan untuk mengarahkan pandangan pembaca pada suatu yang ditonjolkan. Emphasis dalam media pembelajaran dapat dicapai misalnya mengganti ukuran teks, menampilkan ilustrasi dengan proporsi yang lebih besar atau menggunakan warna yang berbeda untuk penekanan.



Gambar 19. Penekanan teks MENGAJI pada halaman judul media pembelajaran
(Rizkiana, 2016)



Gambar 20. Penekanan materi dengan cara membingkai materi dan meletakkannya di pusat layar
(Argaini, 2016)

3. Prinsip Keseimbangan (Balance)

Keseimbangan dipengaruhi berbagai faktor, antara lain faktor tempat posisi suatu elemen, perpaduan antar elemen, besar kecilnya elemen, dan kehadiran elemen

pada luasnya bidang. Keseimbangan akan terjadi bila elemen-elemen ditempatkan dan disusun dengan rasa serasi atau sepadan. Dengan kata lain bila bobot elemen-elemen itu setelah disusun memberi kesan mantap dan tepat pada tempatnya.

Bentuk keseimbangan yang sederhana adalah keseimbangan simetris yang terkesan resmi atau formal, sedangkan keseimbangan asimetris terkesan informal dan lebih dinamis.



Gambar 21. Keseimbangan simetris pada desain media pembelajaran
(Rizkiana, 2016)



Gambar 22. Keseimbangan asimetris pada desain media pembelajaran
(Fajrina, 2016)

4. Prinsip Irama (Ritme)

Ritme terjadi karena adanya pengulangan pada bidang/ruang yang menyebabkan kita dapat merasakan adanya perakan, getaran, atau perpindahan dari unsur satu ke unsur lain. Gerak dan pengulangan tersebut mengajak mata mengikuti arah gerakan yang terjadi pada sebuah karya.

Dari ritme dapat ditentukan eyeflow atau arah baca sebuah desain. Secara umum arah baca suatu desain adalah dari kiri ke kanan dan dari atas ke bawah.



Gambar 23. Unsur pengulangan membentuk ritme dan eyeflow pada media pembelajaran (Argaini, 2016)

5. Prinsip Keselarasan (Harmony)

Kondisi selaras atau harmoni terbentuk karena tidak adanya pertentangan antara satu elemen dengan elemen lainnya. Dalam desain untuk membentuk harmoni dengan dilakukan dengan mengaplikasikan bentuk dan warna yang sama pada elemen sejenis dan menampilkan elemen dengan penggayaan yang sama.



Gambar 24. Elemen yang selaras dari segi warna, gaya desain dan proporsi (Pintaka, 2016)

6. Prinsip Kesatuan (Unity)

Pada akhirnya prinsip kesatuan adalah sebuah kohesi, konsistensi, ketunggalan atau keutuhan, yang merupakan isi pokok dari komposisi. Prinsip ini akan tercapai jika prinsip-prinsip sebelumnya telah diterapkan. Untuk memenuhi prinsip kesatuan, hal yang dapat dilakukan antara lain:

- Menggunakan hanya dua atau tiga type styles (jenis huruf) dengan ukuran yang memiliki keterbacaan yang baik dan relatif sama di seluruh halaman media pembelajaran interaktif.
- Menggunakan palet warna yang identik di seluruh halaman.
- Mengulang warna, bentuk, atau tekstur untuk membentuk irama dan eyeflow
- Menggunakan proporsi ukuran objek grafis yang baik dan peletakan yang teratur
- Menyediakan ruang kosong untuk menghadirkan harmoni



Gambar 25. Kesatuan di tiap-tiap halaman media pembelajaran interaktif
(Pintaka, 2016)

Bab IV

Adobe Flash

Setelah memahami proses perancangan suatu media pembelajaran interaktif, maka tahapan berikutnya adalah proses pengembangan untuk menghasilkan produk yang dimaksud. Pada saat ini cukup banyak tool atau aplikasi yang dapat digunakan untuk mengembangkan sebuah media pembelajaran interaktif. Diantara aplikasi tersebut antara lain :

- Microsoft Power Point
- Prezi
- Android Studio
- Construct
- Unity
- Adobe Flash

Masing-masing aplikasi memiliki keunggulan dan kelemahan. Namun demikian dalam memilih aplikasi kita dapat mengukur dari parameter harga aplikasi, kemudahan untuk didapatkan, kemudahan operasional, fitur yang ditawarkan dan sumber belajar. Dari beberapa parameter tersebut Adobe Flash dipilih oleh penulis sebagai aplikasi untuk mengembangkan media pembelajaran interaktif, karena menawarkan berbagai fitur yang menunjang proses pengembangan media.

Adobe Flash - sebelumnya bernama Macromedia Flash - adalah salah satu perangkat lunak komputer yang didesain khusus oleh Adobe dan merupakan program aplikasi standar authoring tool professional yang digunakan untuk membuat animasi, web maupun aplikasi yang interaktif dan dinamis. Flash didesain dengan kemampuan untuk membuat animasi 2 dimensi yang handal dan ringan sehingga flash banyak digunakan untuk membangun dan memberikan efek animasi pada website, multimedia interaktif, film animasi, game dan yang lainnya. Pada tahun 2012 fitur Flash bertambah sehingga dapat mengelola grafik 3 dimensi melalui bahasa pemrograman dasar bernama AGAL. Flash memiliki fitur untuk membuat animasi berbasis motion, fasilitas action script 3 (bahasa pemrograman), mengelola video lengkap dengan fasilitas playback FLV, mengelola audio dan menghasilkan output dalam berbagai format. Keunggulan lain yang

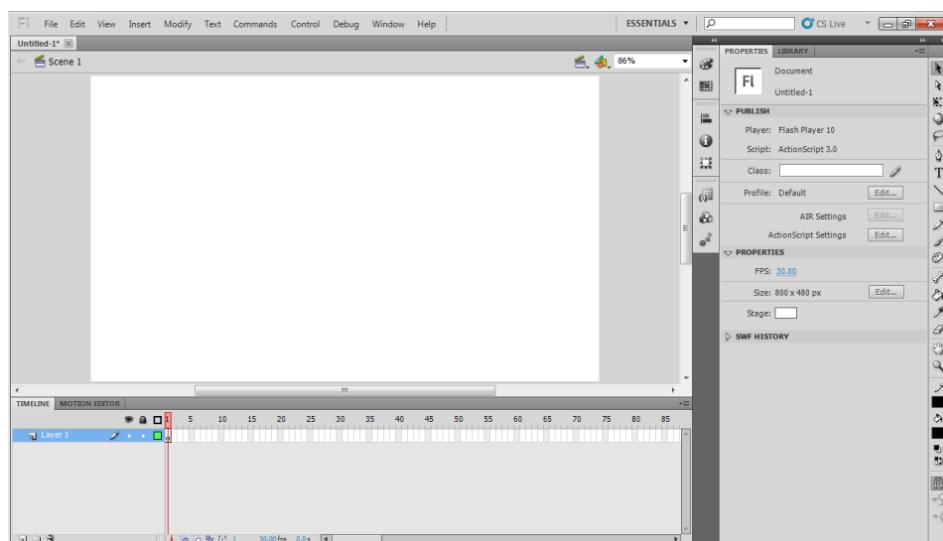
dimiliki Flash adalah output dengan ukuran file yang kecil dan dapat ditampilkan dengan ukuran layar yang dapat disesuaikan dengan keinginan.



Gambar 26. Pemanfaatan Flash dalam aplikasi Sekaran Ville yang memiliki fitur animasi, video dan interaktifitas.

Ruang kerja Adobe Flash

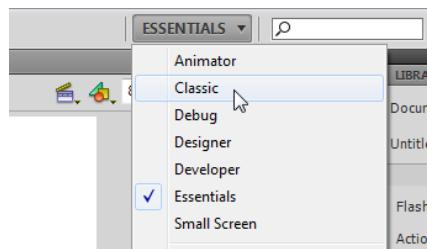
Adobe Flash dapat diinstall dengan menggunakan aplikasi installer yang dengan mudah didapatkan. Pada saat buku ini ditulis Adobe Flash telah memasuki versi CC 2016 (Creative Cloud 2016). Namun demikian buku ini masih sesuai untuk diterapkan dengan menggunakan aplikasi Adobe Flash Profesional CS3 ke atas. Setelah proses installasi selesai dan aplikasi dijalankan maka akan didapatkan tampilan sebagai berikut :



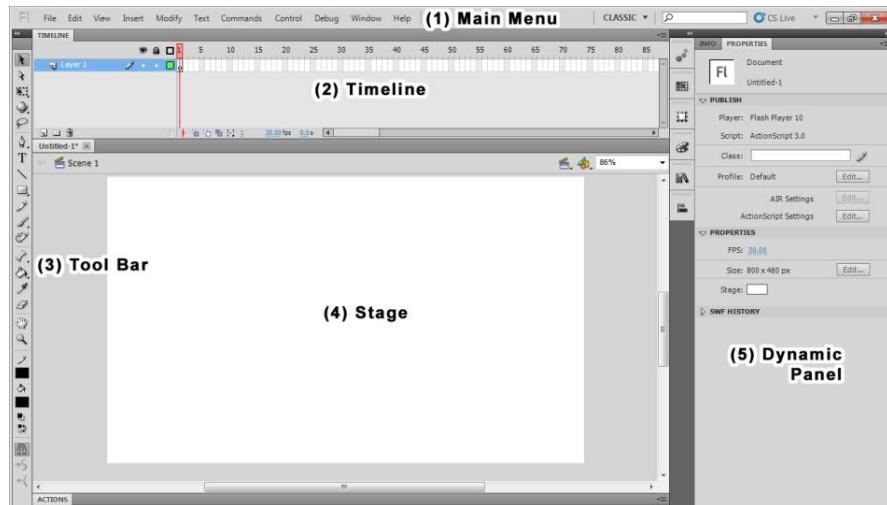
Gambar 27. Tampilan interface Adobe Flash CS5

Dalam buku ini sengaja digunakan Adobe Flash CS5 karena memiliki tampilan yang relatif sama dengan semua versi Adobe Flash terbaru, namun dengan warna yang lebih terang sehingga lebih mudah untuk dijelaskan dengan gambar / screenshot.

Selanjutnya agar lebih familiar dengan versi Flash lama, kita dapat merubah tampilan Adobe Flash menjadi dengan memilih mode Classic. Pilihan mode terdapat di bagian atas tengah, dan mode classic dipilih karena lebih memudahkan kita dalam membangun media pembelajaran interaktif.



Gambar 28. Mengubah ke mode Classic



Gambar 29. Interface Adobe Flash dalam mode Classic

Pada dasarnya interface dari aplikasi Adobe Flash terdiri dari 5 elemen utama, yaitu :

1. Main Menu

Seperti pada aplikasi umumnya, pada bagian atas terdapat menu utama yang di dalamnya terdapat beberapa submenu.

2. Timeline

Timeline merupakan rangkaian waktu yang tersusun atas beberapa frame.

3. Toolbar

Toolbar berisi beberapa perangkat yang dapat digunakan untuk menambah dan mengatur ulang objek yang berada di stage.

4. Stage

Stage merupakan area kerja yang digunakan dalam sebuah proyek.

5. Dynamic Panel

Merupakan panel yang selalu berubah secara dinamis mengikuti tool atau objek aktif, dan berisi properti dari objek atau tool yang aktif tersebut.

Masing-masing tombol, menu dan panel akan dijelaskan fungsinya pada tutorial yang akan di bahas pada bab berikutnya.

Bab V

Membuat animasi sederhana

Fungsi dasar Flash pada saat pertama kali diciptakan adalah untuk membuat animasi sederhana sebagai elemen dalam website. Flash memfasilitasi pembuatan animasi dengan metode yang paling sederhana yaitu frame by frame animation, animasi berbasis motion sampai animasi berbasis kode. Pada bab berikut akan dijelaskan proses membuat animasi sederhana yang nantinya dapat digunakan sebagai salah satu elemen dalam media pembelajaran interaktif.

Dalam Flash kita bisa membedakan jenis animasi berdasarkan tekniknya menjadi 3, yaitu:

1. **Animasi Frame by Frame** yaitu animasi dengan menggunakan beberapa gambar yang berurutan (sequence), sebagai contoh animasi klasik menggunakan teknik ini. Frame by Frame animation digunakan untuk menampilkan animasi yang mengalami perubahan bentuk seperti animasi karakter melakukan gerak tubuh, animasi perubahan bentuk secara dinamis seperti efek gerakan air, api, ledakan dan sebagainya.
2. **Animasi Motion Tween** yaitu animasi dengan menggunakan dua buah keyframe dengan menggerakkan 1 objek dari titik satu ke titik lain, tanpa mengalami perubahan bentuk. Misal kita ingin menganimasikan mobil bergerak dari kiri ke kanan layar.
3. **Animasi Motion Guide** yaitu animasi motion tween yang menggunakan lintasan (guide) sehingga perubahan gerak dapat diatur sedemikian rupa sesuai dengan lintasan yang diinginkan. Misal kita ingin membuat animasi daun yang jatuh dari pohon dan tertutup angin, animasi mobil berjalan disebuah bukit yang tidak rata dan sebagainya

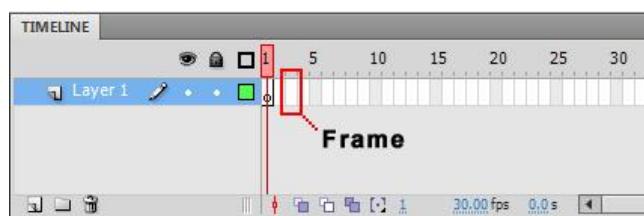
Dengan menggunakan beberapa jenis animasi tersebut, kita dapat memanfaatkan software Flash untuk mengerjakan animasi 2 dimensi untuk film, game, web dan multimedia.

Sebelum mulai membuat animasi sederhana, berikut akan dijelaskan beberapa istilah penting dalam sebuah animasi yang dibuat dengan menggunakan Flash :

1. Frame

Frame dapat kita ibaratkan sebagai lembaran kertas yang di dalamnya terdapat sebuah gambar. Ketika lembaran-lembaran kertas tadi dimainkan satu lembar demi satu lembar, maka gambar yang terdapat dalam frame tersebut seolah-olah mengalami ilusi pergerakan.

Adobe Flash memberikan frame secara digital dan tersusun pada panel timeline. Satu kotak dalam timeline mengindikasikan satu frame.



Gambar 30. Frame

2. KeyFrame

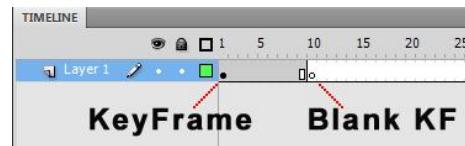
Keyframe adalah sebuah frame yang berisi gambar yang menjadi tolak ukur dari suatu perubahan (gerak, ukuran, warna). Sebagai contoh ketika akan dibuat sebuah animasi tangan melambai, maka yang dimaksud dengan keyframe adalah gambar dengan posisi tangan di awal dan gambar dengan posisi tangan di akhir.



Gambar 31. Keyframe

3. Blank KeyFrame

Dalam Flash Blank Keyframe, adalah Keyframe yang tidak memiliki objek / gambar di dalamnya. Blank keyframe ditandai dengan titik putih pada frame sedang keyframe ditandai dengan titik hitam.

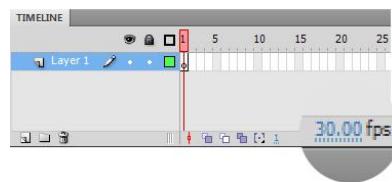


Gambar 32. Keyframe dan Blank Keyframe pada timeline

4. Frame per Second

Frame per Second (FPS) adalah jumlah frame yang digunakan untuk menyusun animasi berdurasi 1 detik. Secara default Flash menggunakan 24 fps, namun demikian standart animasi untuk film adalah 30 fps dan standart untuk game adalah 48 – 60 fps.

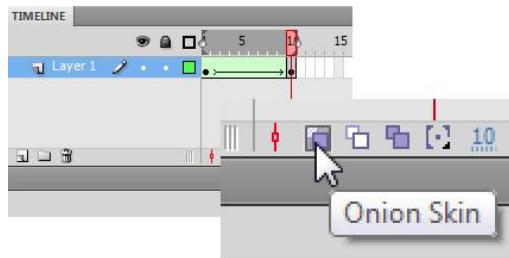
Semakin tinggi nilai fps akan menghasilkan animasi yang terkesan halus. Untuk merubah nilai fps cukup dengan klik ganda angka fps dan ubah nilainya.



Gambar 33. Frame Per Second

5. Onion skin

Sesuai dengan namanya Onion skin akan menampilkan secara transparan frame yang ada sebelum atau sesudah frame aktif. Untuk mempermudah pemahaman terhadap Onion skin, bayangan anda akan membuat sebuah gerakan animasi di beberapa lembar kertas. Setelah kertas pertama selesai digambar, gambar tersebut diletakkan di atas meja kaca dengan sumber cahaya di bawahnya, dan lembaran kertas baru diletakkan di atas gambar tersebut. Hal ini dimaksudkan agar gambar di kertas ke 2 lebih mudah dibuat karena berdasarkan gambar ke 1.



Gambar 34. Onion Skin perubahan bentuk lingkaran ke bentuk tidak beraturan

6. Symbol

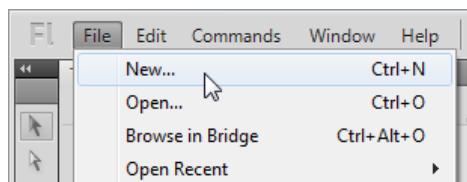
Symbol adalah sebuah objek yang dapat digunakan secara berulang. Symbol terdiri dari tiga jenis yaitu graphic, button (tombol) dan movieclip. Pemahaman tentang symbol akan lebih lengkap seiring dengan tutorial pada bab selanjutnya.

Animasi Frame by Frame

Animasi Frame by Frame adalah sebuah ilusi pergerakan dari sebuah gambar/objek yang diam (still image) yang tersusun oleh gambar demi gambar. Animasi ini sering dipakai dalam animasi klasik. Seorang animator harus membuat gambar di tiap-tiap frame sesuai dengan durasi yang diinginkan.

Untuk membuat animasi frame by frame sederhana, perhatikan langkah berikut :

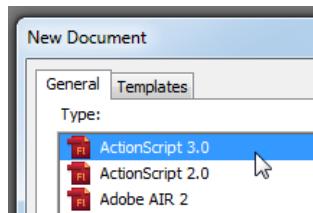
1. Buka aplikasi Adobe Flash, lalu buatlah sebuah file baru dengan menekan tombol **File > New**



Gambar 35. New File

2. Pada panel pilihan New Document, pilih type **Actionscript 3.0**. Actionscript 3.0 (AS3) adalah sebuah bahasa pemrograman yang digunakan oleh Flash. Versi 3 merupakan versi pembaruan dari AS1 dan AS2 dan menjadi standart sejak

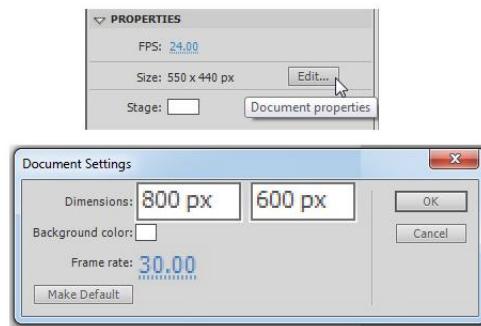
kemunculannya di tahun 2005. Untuk itu setiap project nantinya menggunakan pengaturan AS3 tersebut.



Gambar 36. New Document AS3

3. Klik OK, maka stage akan terbentuk secara default. Pada awal installasi default dari Adobe Flash adalah proyek dengan ukuran 550 x 440 pixel dan 24 fps. Standart tersebut perlu kita ganti sesuai dengan output yang kita inginkan. Sebagai contoh untuk output berupa aplikasi komputer/desktop gunakan ukuran standart minimal 800x600 pixel dan 30 fps, sedang untuk handphone menggunakan ukuran minimal 800x480 pixel dan 30fps.

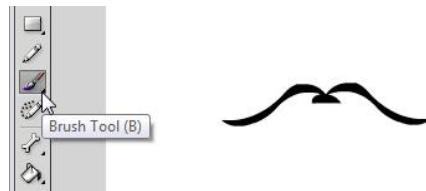
Untuk merubah setting dari projek perhatikan pada bagian **Dynamic panel** di sebelah kanan layar. Pada bagian **properties** terdapat tombol edit untuk mengatur properties projek. Klik tombol **edit** dan ubah parameter Dimensions menjadi width :**800** pixel height:**600** pixel dan fps **30**.



Gambar 37. Document Settings

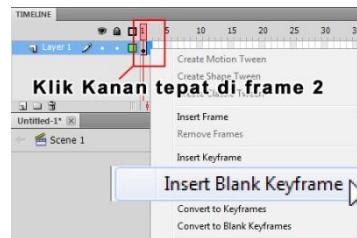
4. Tekan OK, dan ukuran stage akan berubah mengikuti pengaturan setting tersebut.
5. Untuk membuat animasi frame by frame, langkah pertama adalah membuat sebuah gambar sebagai KeyFrame 1. Sebagai contoh kita akan menggambar burung sederhana.

Dengan menggunakan brush tool, gambarlah burung sederhana seperti berikut :



Gambar 38. Brush tool

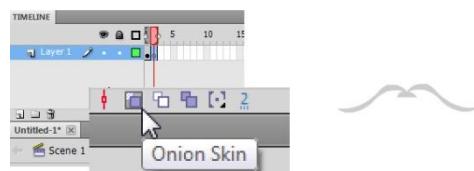
6. Selanjutnya klik frame ke 2, dan tambahkan **Blank Keyframe**. Ketika ada perintah seperti pada kalimat tersebut, maka langkah yang dilakukan adalah klik kanan frame ke 2 lalu memilih opsi Insert Blank Keyframe.



Gambar 39. Insert Blank Keyframe

Blank Keyframe diibaratkan sebagai kertas kosong yang siap untuk ditambahkan gambar animasi berikutnya. Dalam hal ini akan ditambahkan gambar burung dengan posisi sayap yang berbeda

7. Untuk menambahkan gambar burung kita membutuhkan panduan agar gambar di “kertas” (frame) ke dua tidak berbeda ukuran dari gambar di “kertas” pertama. Untuk itu dapat digunakan fitur **Onion Skin**. Tekan tombol Onion Skin pada timeline, maka gambar di frame 1 akan muncul dalam format transparan.



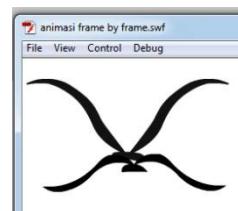
Gambar 40. Onion skin dan hasilnya

8. Kemudian dengan menggunakan brush tool, gambar burung dengan posisi sayap yang berbeda. Lihat gambar :



Gambar 40. Gambar di frame 2

9. Non aktifkan **Onion Skin**, lalu jalankan animasi dengan menekan tombol **Ctrl+Enter**. Maka akan muncul jendela Flash Player yang menampilkan animasi burung tersebut, namun dalam gerakan yang terlalu cepat.



Gambar 41. Hasil animasi frame by frame

Gerakan yang terlalu cepat dikarenakan durasi animasi yang hanya sepersekian detik. Dalam proyek ini FPS di set di angka 30, artinya untuk menghasilkan 1 detik animasi dibutuhkan 30 frame, sedangkan dalam animasi yang telah dibuat hanya berdurasi 2 frame, sehingga dapat diartikan animasi tersebut berdurasi $2/30$ detik atau sekitar 7 milidetik.

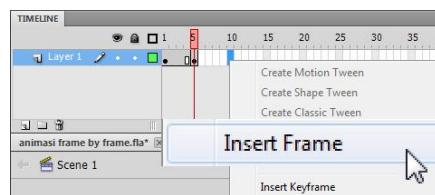
10. Untuk memperlambat animasi tersebut, maka diperlukan durasi frame yang lebih panjang. Untuk melakukannya **drag KeyFrame 2 menuju ke frame 5**.



Gambar 42. Menggeser Keyframe

Dengan cara ini frame 1 akan ditampilkan sebanyak 5/30 detik (5 kali lebih lama dari waktu sebelumnya), sehingga jika animasi dijalankan gambar pertama akan muncul lebih lama dari gambar kedua.

11. Agar gambar ke 2 juga ditampilkan dalam waktu yang sama dengan gambar 1, maka gambar ke 2 membutuhkan durasi. Untuk memberikan durasi yang sama, **klik kanan frame 10 lalu pilih Insert Frame**.



Gambar 43. Insert Frame

12. Jalankan kembali dengan menekan **Ctrl+Enter**, maka animasi akan berjalan lebih lambat dari sebelumnya. Sampai dengan langkah ini animasi frame by frame sudah selesai dibuat, namun jika diinginkan gerakan yang lebih halus, maka dapat dilakukan dengan memperbanyak jumlah keyframe dengan perubahan posisi sayap sedikit demi sedikit.

Animasi Frame by Frame dengan Gambar Siap Pakai

Bagi beberapa orang, menggambar dengan aplikasi lain lebih familiar daripada membuat gambar dengan drawing tool Flash secara langsung. Sebagai contoh dalam proyek selanjutnya akan dibuat sebuah animasi frame by frame burung terbang dengan gambar yang sudah siap pakai. Pada proyek ini digunakan gambar siap pakai yang disimpan dari situs artists-help.tumblr.com, gambar dipisahkan menjadi 9 file dengan penamaan yang berurutan.

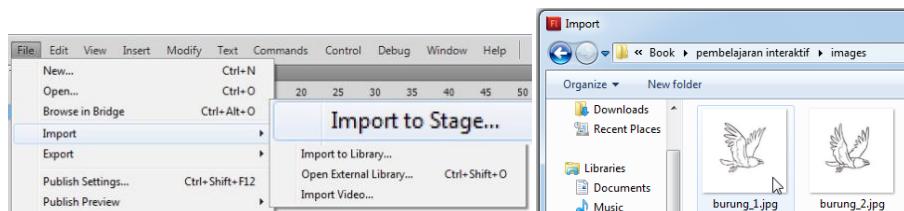


Gambar 44. File gambar dengan penamaan berurutan

Penamaan yang berurutan merupakan standart untuk produksi animasi, karena akan mempermudah penataan gambar. Selain itu masing-masing gambar harus memiliki ukuran yang sama agar proses pengaturan animasi menjadi lebih mudah.

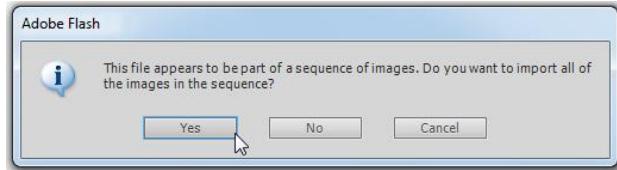
Selanjutnya buatlah sebuah file baru dalam Flash dan ikuti langkah-langkah berikut :

1. Klik menu **File>Import to Stage**, untuk mengimport gambar yang telah disiapkan sebelumnya.
2. Setelah jendela Import muncul pilih file gambar pertama (gambar dengan no urut paling kecil). Dalam contoh ini gambar yang dipilih adalah file **burung_1.jpg**. Kemudian klik **OK**.



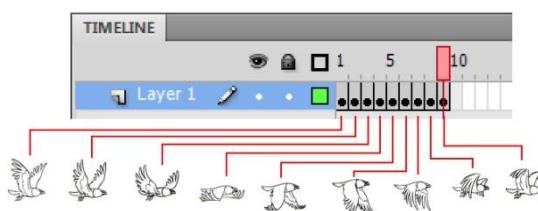
Gambar 45. Import file

3. Selanjutnya akan muncul dialog import sequence. Klik **Yes** untuk mengimport seluruh gambar dan menatanya secara otomatis ke dalam frame sebagai satu sequence.



Gambar 46. Import sequence

4. Apabila penomoran file benar dan tidak terputus, maka gambar secara otomatis akan tertata sebagai animasi frame by frame. Hal ini tentu saja mempermudah seorang animator dalam mengatur sebuah rangkaian animasi yang terdiri dari beberapa gambar.



Gambar 47. Animasi Frame by Frame hasil import

5. Jalankan file dengan menekan **Ctrl + Enter**, maka akan dihasilkan animasi burung terbang yang lebih baik dari tutorial sebelumnya.

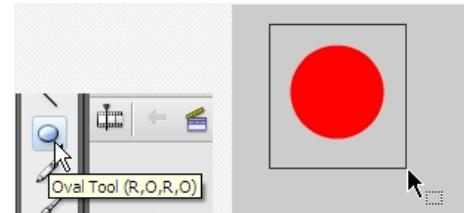
Animasi Motion Tween

Motion Tween adalah animasi pergerakan suatu objek dari keadaan awal sampai keadaan akhir. Motion tween dapat digunakan untuk jenis animasi yang merubah posisi objek, merubah ukuran objek, merubah visibilitas, kecerahan (brightness), dan transparansi warna benda.

Sebagai contoh, pada tutorial berikut akan dibuat sebuah animasi motion tween bola yang bergerak dari kiri ke kanan layar. Perhatikan langkah-langkah berikut :

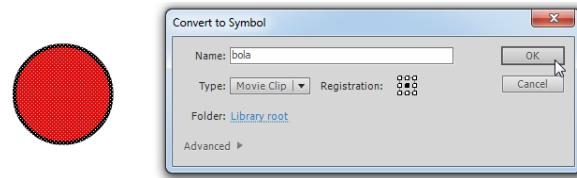
1. Buatlah sebuah file baru
2. Dengan menggunakan **Oval Tool** buatlah sebuah lingkaran, dengan cara *drag and release* pada Stage. Perlu diketahui bahwa Flash adalah program berbasis vector. Setiap objek gambar memiliki 2 penyusun yaitu outline/stroke (garis tepi)

dan fill (isian). Anda dapat merubah warna, ukuran dan posisi objek gambar dengan menggunakan panel properties (dynamic panel).



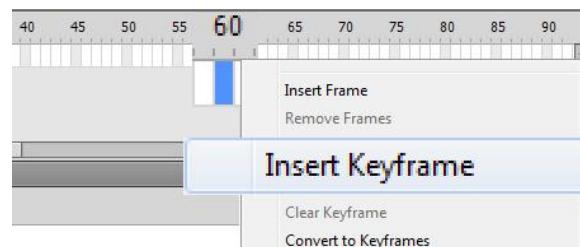
Gambar 48. Oval tool

3. Selanjutnya seleksi lingkaran yang anda buat (stroke dan fill nya), kemudian pilih menu **Modify > Convert to Symbol** atau menekan tombol **F8**. Kemudian anda ketikan “bola” pada kolom *Name*, pilih **Movieclip** pada opsi *Type* dan pilih tengah pada opsi *Registration* Lihat gambar!



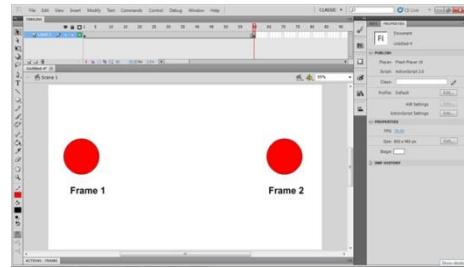
Gambar 49. Convert to Symbol

4. Untuk membuat animasi bola sepanjang 2 detik, klik kanan Frame 60 pada timeline kemudian pilih **Insert Keyframe**.



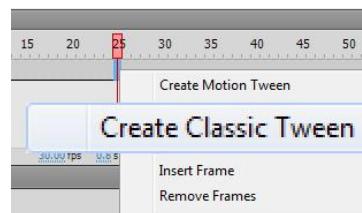
Gambar 50. Insert Keyframe

5. Selanjutnya geser posisi bola yang berada di Frame 60 seperti di gambar. Jadi di frame 1 bola berada di posisi sebelah kiri stage dan di frame 2 berada di kanan.



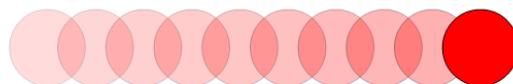
Gambar 51. Peletakan objek di frame 2

6. Klik kanan Frame di antara 2 Key Frame (bebas antara frame 2 sampai frame 59, misal anda klik kanan frame 25) pada timeline kemudian pilih **Create Create Classic Tween**. Maka timeline akan berubah menjadi berwarna biru dan bergaris.



Gambar 52. Create Classic Tween

7. Simpan file, kemudian jalankan animasi dengan menekan tombol **Ctrl+Enter** Maka akan terbentuk sebuah file animasi bertipe SWF dimana bola bergerak dari kiri ke kanan selama 2 detik dan looping secara terus menerus.



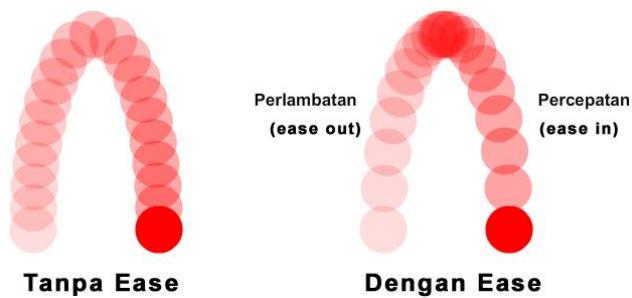
Gambar 53. Hasil animasi motion tween

Prinsip Dasar Animasi Tween Dalam Flash

1. Objek harus di convert menjadi symbol jika tidak flash secara otomatis akan mengkonvert objek dengan nama otomatis, yang akan mempersulit kita ketika sudah memiliki banyak objek untuk dianimasikan.
2. Satu keyframe khusus untuk satu symbol dalam animasi tween, 1 keyframe hanya boleh ditempati 1 buah symbol. Apabila terdapat 2 objek atau lebih maka garis putus-putus pada timeline akan muncul dan animasi tween tidak akan terbentuk sempurna. Gunakan beberapa layer untuk menganimasikan beberapa objek sekaligus
3. Minimal harus memiliki 2 keyframe animasi motion-tween selalu membutuhkan keyframe awal dan keyframe akhir.

Animasi Tween dengan Ease

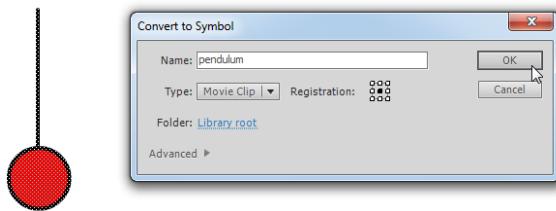
Ease adalah upaya untuk membuat gerakan sebuah animasi menjadi lebih realistik, bisa disebut sebagai perlambatan atau percepatan sebuah animasi. Untuk mendapatkan kesan realis dalam sebuah animasi, kita perlu mengacu pada kondisi riil di dunia nyata. Sebagai contoh ketika kita membuat animasi bola memantul dari atas ke bawah, akan ada efek percepatan dan perlambatan akibat adanya gravitasi. Ketika bola baik ke atas, maka bola akan melambat karena tertarik gravitasi dan ketika bola bergerak ke bawah maka bola akan tertarik gravitasi (mengalami percepatan). Perhatikan gambar berikut



Gambar 54. Animasi dengan ease

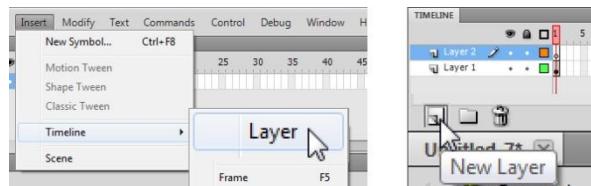
Pada tutorial berikut akan dijelaskan teknik membuat animasi pendulum dengan menggunakan ease untuk menambah kesan realis. Untuk melakukannya perhatikan langkah berikut :

- Buatlah sebuah file baru. Kemudian dengan menggunakan Oval tool dan line tool, buatlah sebuah bola yang tergantung pada seutas tali. Kemudian seleksi semua dan pilih menu **Modify > Convert to Symbol** atau menekan tombol **F8**. Kemudian anda ketikan “pendulum” pada kolom *Name*, pilih **Movieclip** pada opsi *Type* dan pilih atas tengah pada opsi *Registration*. Perhatikan gambar.



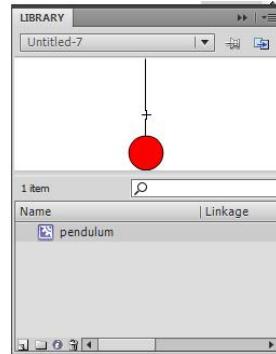
Gambar 55. Symbol “pendulum”

- Untuk membuat animasi 2 buah pendulum yang saling memantul diperlukan 2 gambar pendulum dan 2 buah layer (Karena peraturan bahwa satu keyframe hanya boleh diletakkan 1 buah symbol jika dianimasikan dengan motion tween). Untuk itu buatlah sebuah layer baru di atas layer 1 dengan menekan menu **Insert > Timeline > Layer** atau menekan tombol **New Layer** yang berada di timeline.



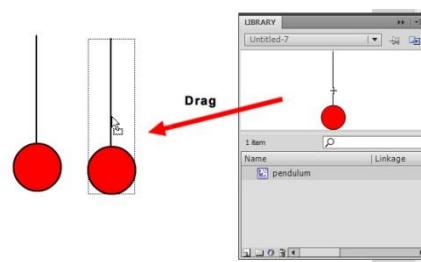
Gambar 56. New Layer

- Karena gambar pendulum yang digunakan sama, maka tidak perlu dibuat ulang. Setiap symbol yang sudah dibuat, secara otomatis akan masuk ke dalam **Library** dan dapat digunakan secara berulang. Buka Library dengan menekan menu **Window > Library** atau dengan menekan tombol **Ctrl+L**. Pada panel Library akan anda dapati symbol pendulum yang telah dibuat sebelumnya.



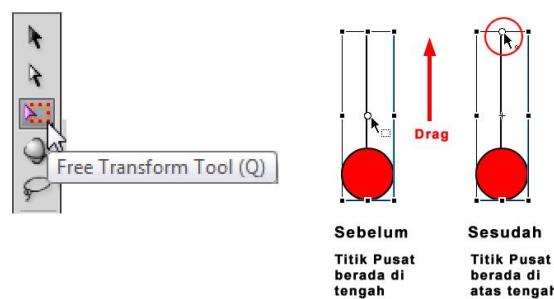
Gambar 57. Library

- Untuk menambahkan pendulum kedua, klik frame 1 layer 2, kemudian **drag symbol pendulum dari library ke stage**. Atur posisinya, sesuaikan dengan pendulum pertama.



Gambar 58. Menambahkan symbol dari library ke stage

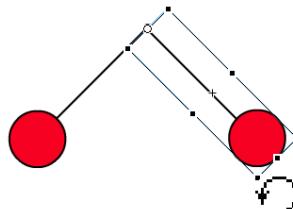
- Karena gerakan pendulum adalah gerakan melengkung / parabola, kita harus memperhatikan titik pusat rotasi. Dengan menggunakan **Free Transform Tool**, geser titik pusat rotasi (pivot point) menuju ke ujung atas benang. Lakukan ada ke dua pendulum.



Gambar 59. Menggeser Pivot Point

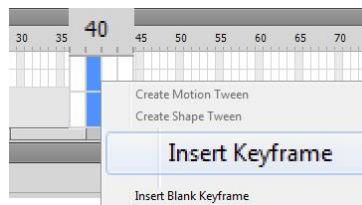
- Setelah pengaturan pivot point kedua pendulum, dengan menggunakan **Free Transform Tool** atur posisi ke dua pendulum tersebut ke posisi awal. Untuk

merotasi objek dekatkan mouse ke ujung objek sehingga muncul ikon rotasi, kemudian drag sesuai dengan kebutuhan. Lihat gambar!



Gambar 60. Posisi awal pendulum

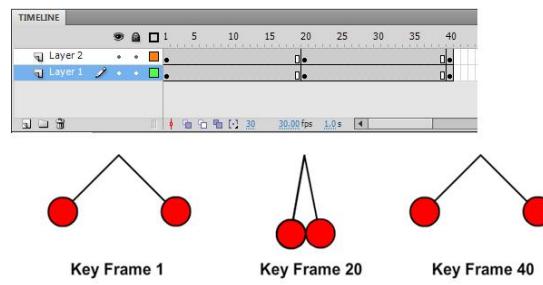
7. Perlu diperhatikan bahwa gerakan pendulum akan berulang, dimana gerakan utama pendulum ada tiga langkah seperti pada gambar 62, sehingga masing-masing layer membutuhkan 3 buah Key Frame.
8. Karena posisi awal dan posisi akhir dari pendulum sama, maka langkah yang paling tepat adalah mengerjakan terlebih dahulu posisi awal dan akhir. Seleksi **frame 40 layer 1 dan layer 2** (dengan menahan tombol shift + klik pada frame 40), kemudian pertahankan posisi mouse, klik kanan dan pilih **Insert Key Frame**.



Gambar 61. Insert Keyframe 40 – tanpa perubahan apapun

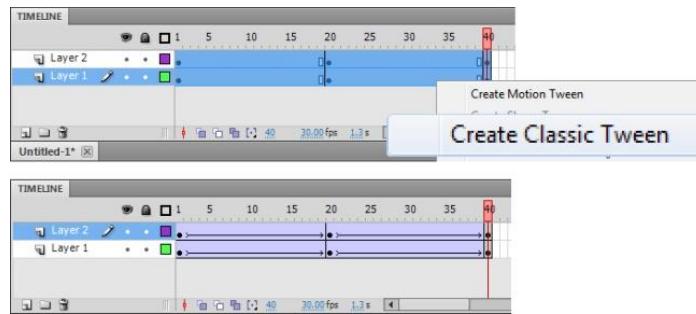
9. Ketika posisi pendulum sedang turun (di tengah / di bawah), maka waktu yang tepat adalah setengah dari waktu total. Karena waktu total adalah 40 frame, maka setengah dari waktu adalah 20 frame. Seleksi **frame 20 layer 1 dan layer 2** (dengan menahan tombol shift + klik pada frame 20), kemudian pertahankan posisi mouse, klik kanan dan pilih **Insert Key Frame**.

Kemudian dengan menggunakan **Free Transform Tool**, atur posisi kedua pendulum sesuai dengan gambar berikut:



Gambar 62. Pengaturan Keyframe pendulum

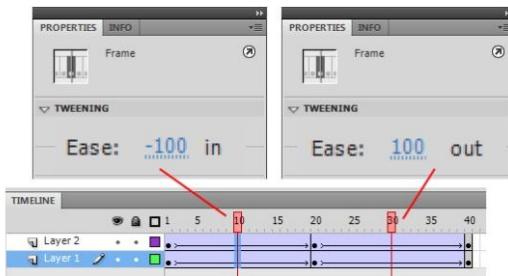
- Untuk menambahkan animasi, **seleksi seluruh frame layer 1 dan layer 2** (klik frame 1 layer 2, tahan tombol shift dan klik frame 2 layer 1). Pertahankan posisi kursor mouse, klik kanan dan pilih **Create Classic Tween**.



Gambar 63. Menambahkan Classic Tween pada seluruh layer

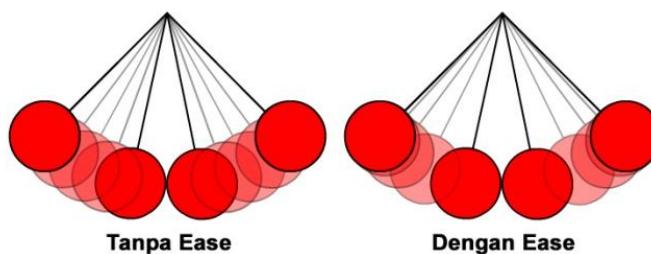
- Jalankan aplikasi dengan menekan tombol **Ctrl+Enter**. Maka akan terbentuk animasi gerakan pendulum. Namun demikian jika anda perhatikan gerakan tersebut masih terkesan kaku (tidak alami), karena efek grafitasi tidak bekerja.
- Untuk membuat animasi tersebut lebih terkesan realis, perlu ditambahkan **ease**. Ketika objek tertarik grafitasi (frame 1 sampai frame 20) maka diperlukan ease bernilai negatif (**ease in**) dan sebaliknya ketika objek memantul ke atas maka pendulum akan semakin melambat, sehingga diperlukan ease bernilai positif (**ease out**).

Untuk mengatur ease, **klik frame 10 layer 1 dan layer 2**. Pada panel properties ubah nilai parameter ease menjadi **-100 (ease in)**. Selanjutnya **klik frame 30 layer 1 dan layer 2** lalu ubah nilai parameter ease menjadi **100 (ease out)**.



Gambar 64. Pengaturan ease

13. Jalankan aplikasi dengan menekan **Ctrl+Enter**. Maka animasi pendulum yang dihasilkan akan lebih realis.



Gambar 65. Perbedaan Animasi pendulum tanpa dan dengan ease

Animasi Tween dengan Rotasi

Dalam animasi, rotasi merupakan salah satu gerak yang sering dipakai. Sebagai contoh : gerakan roda, perubahan sudut gerak dan sebagainya. Flash memiliki opsi rotasi pada animasi motion tween yang bisa mempermudah kita untuk memperoleh animasi yang kita butuhkan.

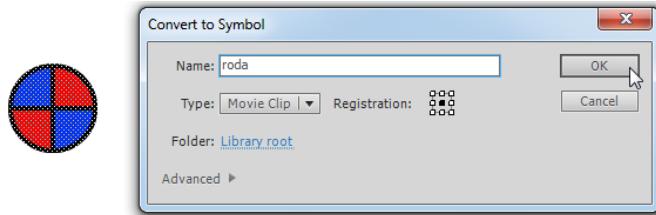
Rotasi dalam Flash terbagi menjadi 3 yaitu :

1. CW (Clockwise) yaitu rotasi searah jarum jam
2. CCW (Counter Clockwise) yaitu rotasi berlawanan jarum jam
3. Auto rotation yaitu rotasi otomatis tergantung perubahan rotasi yang dilakukan dengan transform tool.

Pada tutorial berikut, akan dijelaskan proses pembuatan animasi roda yang berputar dan bergerak dari kiri ke kanan. Perhatikan langkah-langkah berikut :

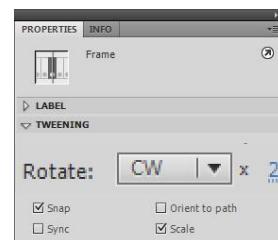
1. Buatlah sebuah file baru. Kemudian dengan menggunakan **Oval tool** buatlah sebuah lingkaran sempurna (dengan menahan tombol **Shift** saat membuat lingkaran). Agar lingkaran terlihat berputar, maka perlu ditambahkan beberapa jari-jari dan warna pembeda. Dalam contoh ini lingkaran dibagi menjadi 4.

2. Seleksi roda tersebut kemudian convert menjadi movieclip dengan menekan menu **Modify > Convert to Symbol** atau menekan tombol **F8**. Ketikan **roda** pada nama, dan pastikan titik registrasi berada di tengah.



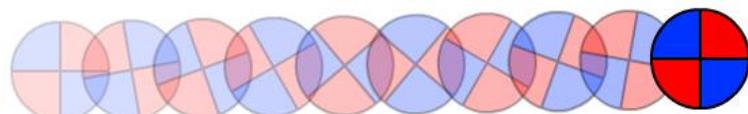
Gambar 66. Mengconvert Movieclip roda

3. Buatlah sebuah animasi roda dari kiri ke kanan, seperti pada tutorial sebelumnya (**Klik kanan frame 60 pilih Insert Keyframe**, geser posisi roda ke kanan, kemudian **klik kanan frame 25** dan pilih **Add Classic Tween**.)
4. Selanjutnya klik **frame 25** (salah satu diantara 2 keyframe) dan perhatikan panel **properties**. Kemudian pada opsi rotate pilih CW dan ketikan 2 pada kolom times untuk mendapatkan efek 2 kali putaran.



Gambar 67. Pengaturan rotasi

5. Simpan file, kemudian jalankan animasi dengan menekan tombol **Ctrl+Enter** Maka akan terbentuk sebuah file animasi bertipe SWF dimana 2 bola bergerak dari kiri ke kanan dengan rotasi 2 putaran searah jarum jam. Perhatikan gambar!



Gambar 68. Hasil animasi rotasi

Animasi Roda Gigi

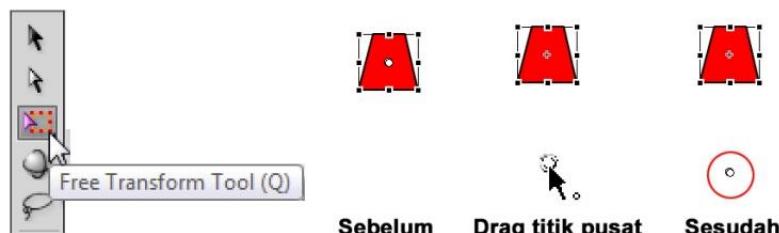
Dalam media pembelajaran yang terkait dengan ilmu teknik khususnya permesinan, sering-kali terdapat materi tentang roda gigi. Membuat animasi roda gigi dengan Flash dapat dilakukan dengan menggunakan metode animasi tween dengan penambahan rotasi. Perhatikan contoh berikut :

1. Buatlah file baru. Dengan menggunakan **Rectangle tool** buatlah sebuah kotak kecil yang akan kita jadikan sebagai gigi-gigi roda. Klik **Selection Tool**, dekatkan kurSOR mouse ke ujung kotak lalu drag sedikit sehingga kotak terdistorsi menjadi trapesium sesuai gambar.



Gambar 69. Membuat gigi

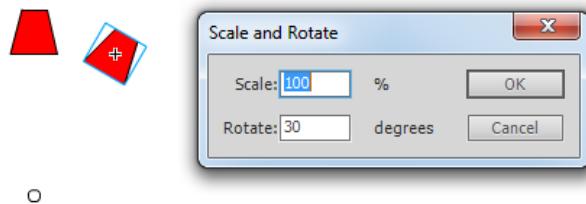
2. Seleksi gambar tersebut, kemudian **Convert to Symbol** menjadi **Movieclip** dengan nama “gigi”. Untuk membuat gigi-gigi yang berulang dalam bentuk lingkaran, maka kita dapat melakukan metode “copy-paste”. Awalnya kita tentukan jari-jari roda dengan menggeser pivot point dari gigi. Dengan menggunakan **Free Transform Tool**, drag pusat movieclip jauh ke bawah sesuai dengan jari-jari yang diinginkan.



Gambar 70. Drag titik pusat

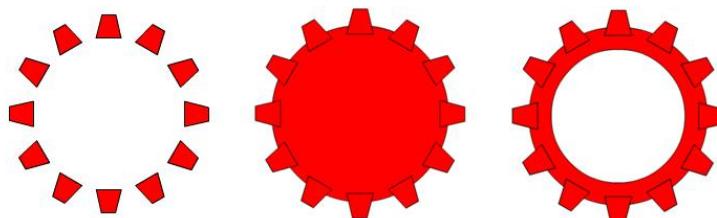
3. Seleksi movieclip “gigi” tersebut, kemudian **Copy** dengan menekan **Ctrl+C**. Selanjutnya **Paste in Place** dengan menekan **Ctrl+Shift+V**. (Dalam Flash **Paste** terdiri dari 2 macam, yaitu **Paste** standar – dimana objek baru akan diletakkan di tengah layar, dan **Paste in Place** – dimana objek baru akan diletakkan tepat di posisi ketika mengcopy).

Selanjutnya dalam kondisi “gigi” hasil paste terseleksi, tekan tombol **Ctrl+Alt+S**, untuk mengeluarkan panel **Scale and Rotate**. Ketikkan angka **30** pada kolom **rotate**, maka gigi akan terotasi 30 derajat searah jarum jam.



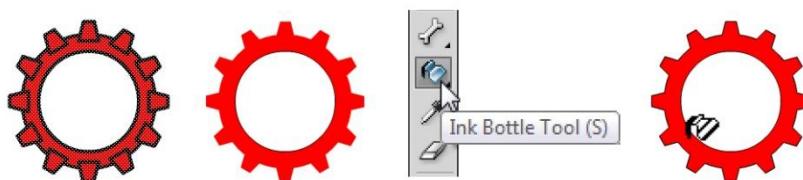
Gambar 71. Scale and Rotate

- Ulangi langkah nomor 3 di atas beberapa kali, sehingga akan terbentuk “gigi” yang melingkar seperti pada gambar. Selanjutnya dengan menggunakan **Oval tool** anda dapat menyempurnakan bentuk gigi-gigi tersebut menjadi sebuah roda gigi.



Gambar 72. Tahapan membuat roda gigi

- Untuk menyempurnakan bentuk roda gigi, seleksi semua objek kemudian tekan **Ctrl+B** (Break Apart). Break Apart adalah mengubah kembali sebuah symbol atau group menjadi grafik dasar yang dapat diedit kembali. Selanjutnya dobel klik outline hitam yang ada dan hapus dengan menekan tombol **Delete**. Selanjutnya klik **Ink Bottle Tool (S)** dan klik objek roda gigi untuk menambahkan **outline**.

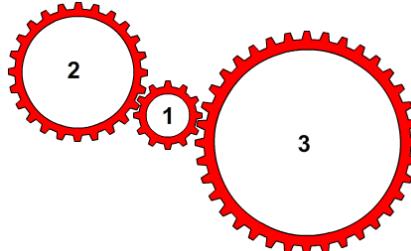


Gambar 73. Menyempurnakan roda gigi

- Seleksi roda gigi tersebut, kemudian **Convert to Symbol** menjadi **Movieclip** dengan nama “roda gigi 1”. Dengan cara yang sama, buatlah 2 buah roda gigi

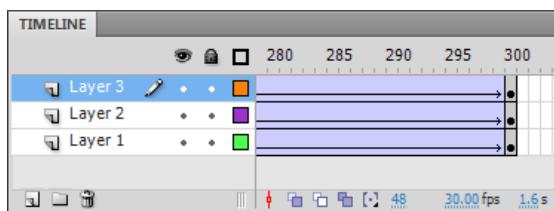
baru dengan ukuran yang berbeda, dan letakkan masing-masing roda gigi di sebuah layer baru.

(Dalam contoh ini ukuran roda gigi dibuat memiliki rasio 1: 2 : 3).



Gambar 74. Pengaturan roda gigi

7. Untuk menambahkan animasi (misalkan untuk putaran roda terbesar berdurasi 10 detik), maka **seleksi frame 300 seluruh layer** (layer 1 sampai layer 3), kemudian tambahkan **Keyframe**. Selanjutnya seleksi seluruh frame di seluruh layer dan **Create Classic Tween**.



Gambar 75. Menambahkan animasi

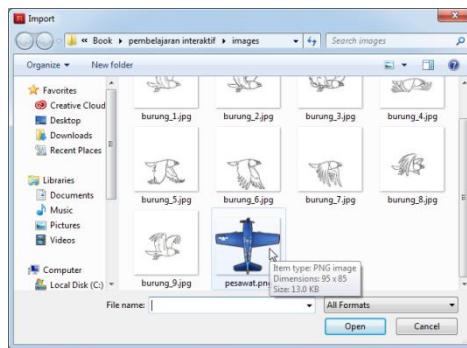
8. Selanjutnya untuk menambahkan rotasi, **klik frame 1 layer 3** pada panel **Properties** atur parameter **rotate** menjadi **CW x1**. **klik frame 1 layer 2** pada panel **Properties** atur parameter **rotate** menjadi **CW x2** **klik frame 1 layer 1** pada panel **Properties** atur parameter **rotate** menjadi **CCW x1**.
9. Simpan file dan jalankan animasi dengan menekan **Ctrl+Enter**, maka akan terbentuk animasi roda gigi.

Animasi Motion Guide

Dalam membuat animasi terkadang kita membutuhkan sebuah gerakan yang tidak beraturan, tidak seperti motion tween yang berbasis pada perubahan tempat/bentuk dalam garis lurus. Sebagai contoh ketika kita akan membuat sebuah animasi kupu-kupu yang sedang terbang, maka secara riil kupu-kupu tersebut akan terbang secara bebas (tidak lurus), sehingga untuk animasi yang seperti itu dibutuhkan sebuah garis bantu atau disebut sebagai **motion guide**.

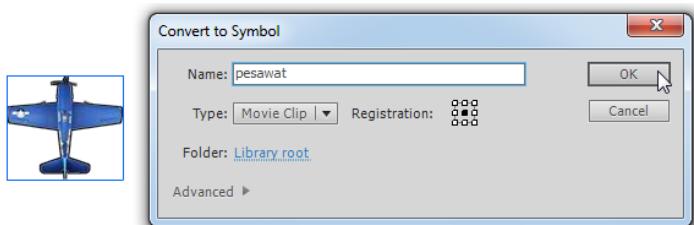
Pada tutorial ini akan dijelaskan cara membuat animasi pesawat yang bergerak mengikuti sebuah pola, sebagai berikut:

1. Buatlah sebuah file baru. Selanjutnya klik menu **File > Import > Import to Stage** kemudian pilihlah sebuah gambar **pesawat.png**. Pesawat tersebut nantinya akan kita gerakkan sesuai dengan pola guide.



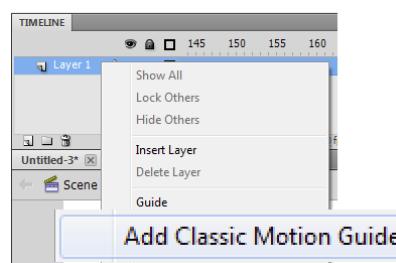
Gambar 76. Mengimport gambar pesawat

2. Setelah gambar pesawat muncul di stage, klik pesawat kemudian pilih menu **Modify > Convert to Symbol**. Ketikan “pesawat” pada kolom nama, dan pastikan titik registrasi-nya ada di tengah.



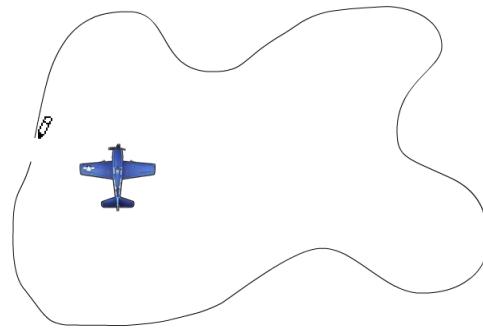
Gambar 77. Mengconvert symbol pesawat

3. Klik **frame 200** kemudian tambahkan **keyframe**.
4. Untuk menambahkan pola / guide yang akan menjadi lintasan dari gerakan pesawat, **klik kanan Layer 1**, kemudian pilih **Add Classic Motion Guide**, maka sebuah layer baru akan muncul di atas layer 1.



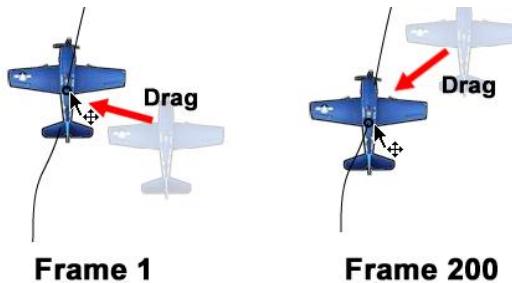
Gambar 78. Add Classic Motion Guide

5. Klik **frame 1 layer guide**, kemudian dengan menggunakan **Pencil Tool** atau **Pen Tool** buatlah sebuah pola lintasan. Untuk membuat pola tersebut terdapat 2 syarat yang harus terpenuhi : - garis yang dibuat tidak boleh putus - garis yang dibuat memiliki 2 buah ujung (awal lintasan dan akhir lintasan).



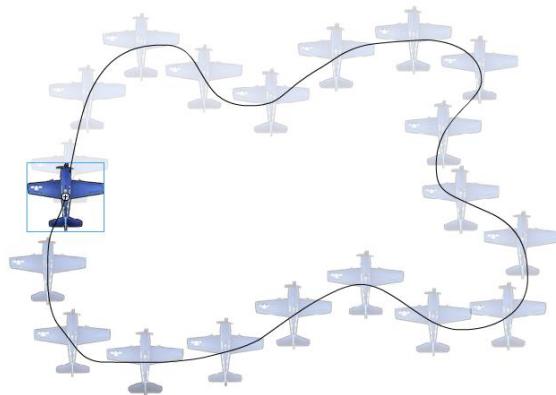
Gambar 79. Membuat lintasan

6. Selanjutnya klik “pesawat” pada frame 1 kemudian **drag ke awal lintasan**, sampai menempel (agar menempel dengan baik, pastikan ikon magnet di drawing tool menyala, dan pastikan lingkaran kecil di tengah movieclip berubah menjadi lingkaran yang lebih besar).



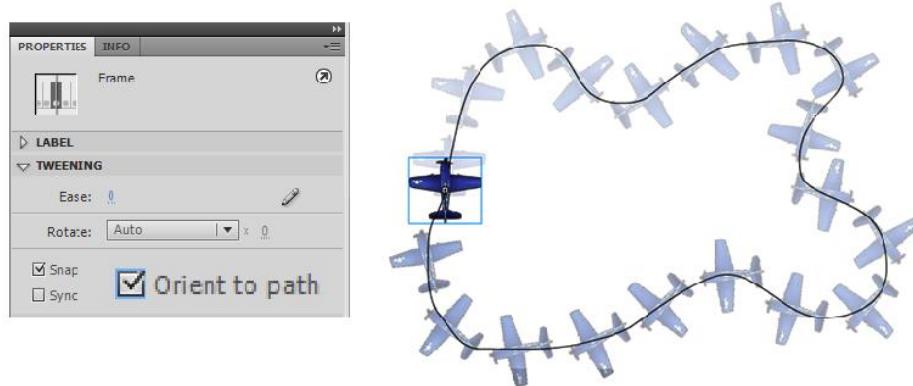
Gambar 80. Drag ke awal dan ke ahir lintasan

7. Dengan cara yang sama klik “pesawat” di **frame 200** lalu **drag menuju akhir lintasan**.
8. Klik kanan **frame 100** pada timeline kemudian pilih **Create Classic Tween**. Maka timeline akan berubah menjadi berwarna biru dan bergaris, dan animasi pesawat bergerak mengikuti pola akan terbentuk.



Gambar 81. Animasi Motion Guide

9. Simpan file, kemudian jalankan animasi dengan menekan tombol **Ctrl+Enter**. Jika anda perhatikan pesawat akan bergerak mengikuti pola, akan tetapi posisi pesawat tetap tegak. Agar pesawat dapat mengikuti pola lintasan, sekaligus menyesuaikan rotasi pesawat terhadap lintasan, maka seleksi frame 100 layer 1 kemudian buka panel **properties** dan mengaktifkan opsi **Orient to path**, maka pesawat akan bergerak dan berotasi sesuai dengan pola.

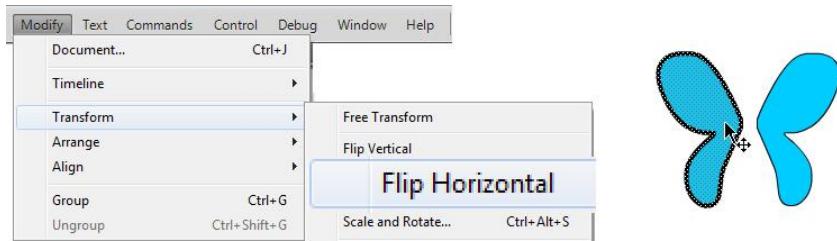


Gambar 82. Penambahan Orient to Path

Memahami Movieclip

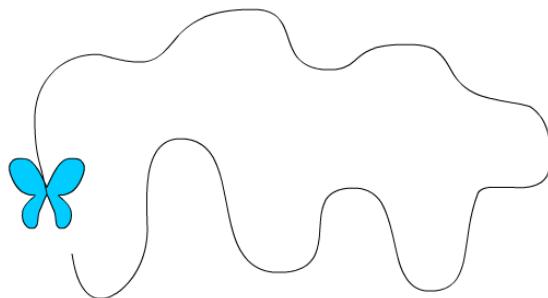
Movieclip merupakan salah satu jenis symbol di Flash yang memiliki timeline tersendiri. Sehingga dapat dikatakan bahwa Movieclip merupakan sebuah animasi di dalam animasi. Sebagai contoh ketika akan membuat animasi kupu-kupu yang sedang terbang, maka kupu-kupu memiliki animasi bergerak bebas atau mengikuti lintasan path, namun demikian ketika kupu-kupu tersebut bergerak melintasi lintasan sayap kupu-kupu tersebut juga beranimasi/bergerak. Untuk lebih jelasnya perhatikan contoh berikut :

- Buatlah file baru. Selanjutnya dengan menggunakan Drawing tools, buatlah sebuah gambar kupu-kupu sederhana. Untuk mempermudah proses, buatlah satu sisi sayap, kemudian duplikasi dan balik posisinya dengan menu **Modify > Transform > Flip Horizontal**.



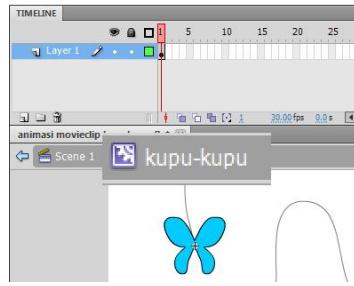
Gambar 83. Flip Horizontal

- Seleksi gambar tersebut dan **Convert to Symbol** menjadi **Movieclip** “kupu-kupu”.
- Klik **frame 200** dan tambahkan **Keyframe**. Selanjutnya klik kanan **Layer 1** dan pilih **Create Classic Motion Guide**. Pada layer guide tersebut buatlah sebuah lintasan dengan menggunakan **Pencil Tool** atau **Pen Tool**.



Gambar 84. Lintasan kupu-kupu

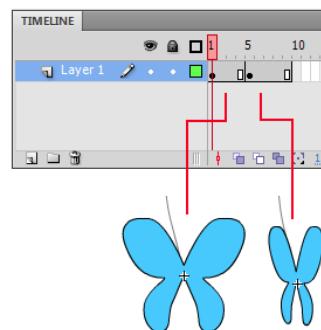
- Separl pada tutorial sebelumnya klik frame 1 layer 1 dan **drag movieclip “kupu-kupu” ke awal lintasan**. Selanjutnya klik frame 200 layer 1 dan **drag movieclip “kupu-kupu” ke akhir lintasan**.
- Untuk menganimasikan kupu-kupu, klik kanan **frame 100 layer 1** dan pilih **Create Classic Tween**.
- Simpan file, dan kemudian jalankan animasi dengan menekan tombol **Ctrl+Enter**. Maka akan didapatkan sebuah animasi kupu-kupu melintasi lintasan seperti halnya pada tutorial gerakan pesawat sebelumnya.
- Untuk menambahkan gerakan sayap kupu-kupu, perlu dilakukan perubahan pada movieclip kupu-kupu. Klik **keyframe 1 layer 1**, kemudian **dobel klik movieclip kupu-kupu** untuk memasuki **mode edit symbol**.



Gambar 85. Mode edit symbol “kupu-kupu”

Pada **mode edit symbol**, timeline yang ada adalah milik movieclip “kupu-kupu”, sehingga tampak hanya 1 keyframe. Apapun yang dilakukan pada mode edit ini hanya berdampak pada movieclip namun tidak merubah animasi yang sudah dibuat sebelumnya di Scene 1.

8. Untuk mendapatkan animasi kepakan sayap kupu-kupu, **klik kanan frame 5**, kemudian tambahkan **Blank Keyframe**. Selanjutnya dengan bantuan **Onion skin**, buatlah gambar sayap kupu-kupu dengan posisi yang berbeda. Klik kanan **frame 9** kemudian **Insert Frame** untuk menambah durasi gambar sayap di Keyframe ke dua.



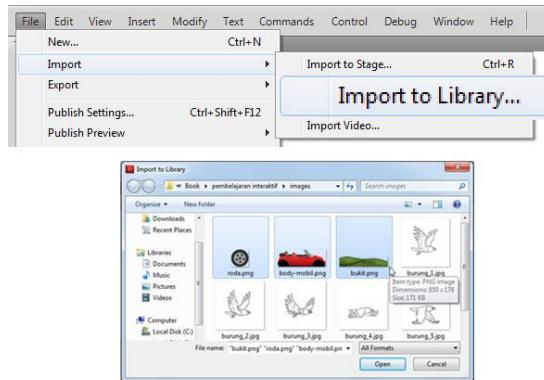
Gambar 86. Animasi frame by frame kepakan kupu-kupu

9. Keluar dari mode edit dengan menekan **Ctrl+E**, kemudian jalankan kembali aplikasi dengan menekan **Ctrl+Enter**. Maka akan didapat animasi gerakan kupu-kupu melintasi lintasan sekaligus mengepakkan sayap.

Animasi Motion Guide dengan Movieclip Kompleks

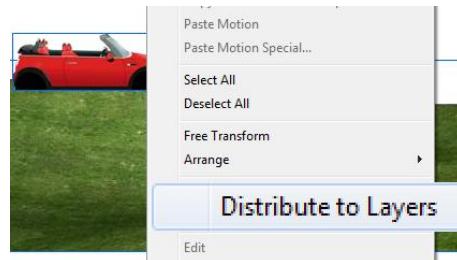
Dalam beberapa kasus diperlukan animasi yang lebih komplek, misalnya membuat animasi mobil yang melintasi sebuah bukit. Bagi animator profesional hal ini terdengar mudah, namun bagi pemula akan sulit sekali memulai pembuatan animasi tersebut. Mobil sendiri memiliki beberapa roda yang ikut bergerak bersama mobil. Tantangan seperti ini yang harus diselesaikan secara bertahap. Perhatikan contoh berikut :

- Buatlah sebuah file baru. Selanjutnya klik menu **File > Import > Import to Library** kemudian pilihlah 3 buah gambar (ada di dalam file sumber) yaitu body_mobil.png, roda.png dan bukit.png. Pemilihan file bertipe PNG dikarenakan file tersebut mendukung mode transparan.



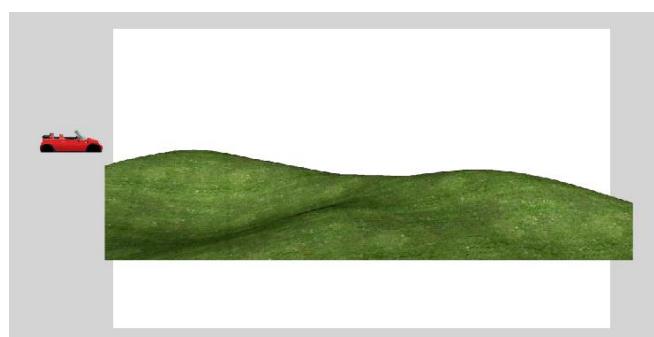
Gambar 87. Import to Library

- Setelah semua objek gambar muncul di **Library**, drag body_mobil.png dan bukit.png ke **Stage**. Selanjutnya seleksi semuanya kemudian klik kanan dan pilih **Distribute to Layers**. Dengan opsi tersebut, masing-masing objek akan dibuatkan layer masing-masing. (perlu diingat bahwa dalam animasi motion, 1 buah objek membutuhkan 1 keyframe khusus, sehingga dalam hal ini 1 objek membutuhkan 1 layer khusus)



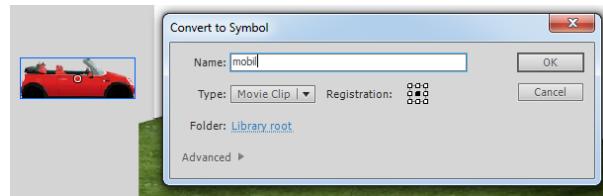
Gambar 88. Distribute to Layers

- Kemudian atur posisi body-mobil dan bukit seperti tampak pada gambar.



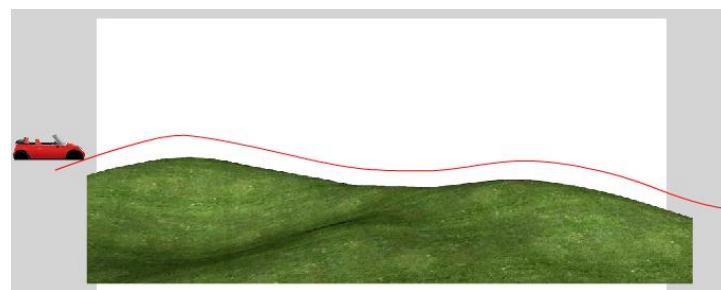
Gambar 89. Pengaturan Objek

- Klik kanan **frame 100 layer bukit** kemudian pilih **Insert frame**. Penambahan frame dimaksudkan untuk memberikan jeda waktu tanpa ada perubahan apapun.
- Klik body-mobil kemudian pilih menu **Modify > Convert to symbol**. Ketikan mobil pada kolom nama, dan pastikan titik registrasi-nya ada di tengah.



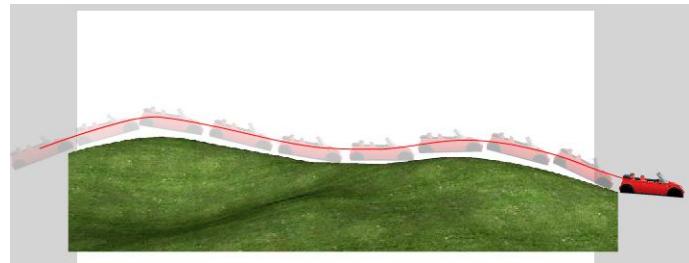
Gambar 90. Mengconvert movieclip mobil.

- Untuk menambahkan pola / guide yang akan menjadi lintasan dari gerakan mobil klik kanan **frame 100 Layer body-mobil** kemudian pilih **Insert Keyframe**, Kemudian klik kanan **Layer body-mobil**, kemudian pilih **Add Classic Motion Guide**, maka sebuah layer baru akan muncul di atas layer body-mobil.
- Klik **frame 1 layer guide**, kemudian dengan menggunakan **Pencil Tool** atau **Pen Tool** buatlah sebuah pola lintasan mengikuti pola bukit.



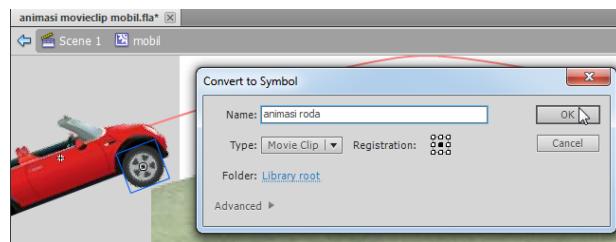
Gambar 91. Lintasan mobil di atas bukit

- Selanjutnya klik movieclip mobil pada **frame 1** kemudian **drag ke awal lintasan**, sampai. Atur rotasinya agar sesuai dengan pola lintasan. Dengan cara yang sama klik mobil di **frame 100** lalu **drag menuju akhir lintasan**, atur juga rotasi lintasannya.
- Klik kanan **frame 50 Layer body-mobil** kemudian pilih **Create Classic Tween** maka animasi mobil bergerak mengikuti pola akan terbentuk. Agar rotasi mobil mengikuti pola lintasan, **klik frame 50 layer body-mobil** kemudian buka panel properties dan aktifkan opsi **Orient to path**.



Gambar 92. Animasi mobil tanpa roda

10. Setelah animasi mobil bergerak di atas bukit terbentuk, tahapan selanjutnya adalah menambahkan **animasi roda**. Hal yang harus dipahami adalah roda menempel ke body-mobil sehingga bergerak bersama-sama. Selain itu terdapat 2 roda yang sama bentuk dan gerakannya. Dalam kasus tersebut kita cukup membuat 1 buah animasi roda dan kemudian menduplikatnya menjadi 2.
11. **Klik frame 1 layer body-mobil**, selanjutnya dobel klik movieclip mobil, maka kita akan memasuki **mode edit movieclip mobil**.
12. Buka panel **Library** kemudian **drag roda ke stage**, atur posisinya (letakkan di bagian depan). Selanjutnya seleksi gambar roda tersebut dan **Convert menjadi movieclip "animasi roda"**.



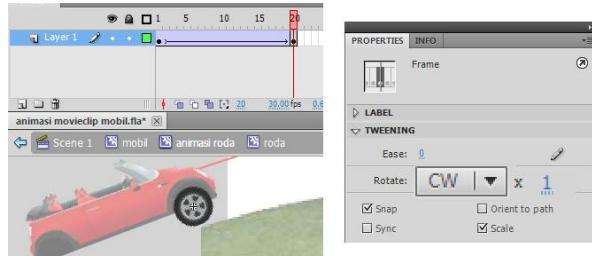
Gambar 93. Movieclip animasi roda

13. Dobel klik movieclip animasi roda sehingga kita berada dalam **mode edit movieclip "animasi roda"** Seleksi gambar roda, kemudian **Convert menjadi movieclip "roda"**



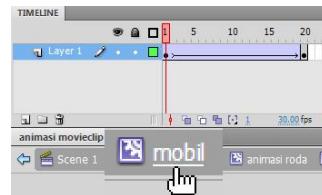
Gambar 94. Membuat animasi roda

14. Untuk membuat animasi roda berputar klik kanan **frame 20 layer 1**, kemudian pilih **Insert Keyframe**. Selanjutnya buka panel properties dan pilih **CW** pada bagian **rotate** untuk menambahkan rotasi searah jarum jam.



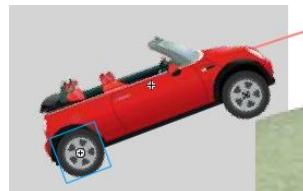
Gambar 95. Membuat rotasi roda

15. Tekan **MovieClip mobil** untuk kembali ke **mode edit movieclip mobil**



Gambar 96. Kembali ke mode edit mobil

16. Selanjutnya **Copy movieclip "animasi roda"** lalu **paste** dan atur posisinya sebagai roda belakang.



Gambar 97. Pengaturan roda belakang

17. Simpan project, tekan **Ctrl+E** untuk keluar dari mode edit symbol, kemudian jalankan animasi dengan menekan tombol **Ctrl+Enter**. Maka animasi mobil bergerak di atas bukit akan terbentuk.



Gambar 98. Hasil animasi mobil

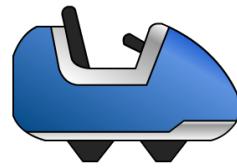
Animasi Menggunakan Metode Graphic

Salah satu symbol dalam Flash adalah **graphic**. Pada dasarnya graphic memiliki konsep yang hampir sama dengan movieclip, yaitu memiliki timeline sendiri. Akan tetapi terdapat perbedaan konsep mendasar antara movieclip dan graphic sebagai berikut :

- Movieclip adalah sebuah **animasi independen**. Animasi ini akan bergerak secara terus menerus apabila diletakkan di dalam sebuah keyframe. Sebuah movieclip berdurasi beberapa frame dapat muncul/dimainkan secara penuh hanya dengan menempati 1 buah keyframe.
- Dalam stage, Movieclip hanya ditampilkan pada posisi frame 1 saja. Animasi Movieclip baru akan terlihat jika animasi/movie sudah dijalankan dengan Ctrl+Enter. Sehingga ketika di eksport menjadi file bertipe AVI/MOV, akan terjadi kesalahan karena **hanya menampilkan frame 1** dari movieclip - bukan animasi secara keseluruhan.
- **Movieclip selalu dimulai dari frame 1** dan tidak dapat diatur frame awal dimulainya animasi. Kecuali diatur menggunakan kode Actionscript.
- Berbeda dengan symbol bertipe **Graphic**. Symbol bertipe Graphic membutuhkan **durasi frame**, untuk menampilkan animasi secara penuh. Animasi **graphic akan tampak di timeline tergantung dari durasi (frame) yang ada** pada timeline.
- Symbol bertipe Graphic **memiliki opsi pengaturan untuk awal animasi**, waktu animasi dan pemutaran ulang animasi. Sehingga dapat digunakan untuk memanipulasi waktu dari animasi yang dibuat.
- Symbol bertipe Graphic **tidak dapat diakses menggunakan kode Actionscript**. Sehingga proses pengaturan animasi harus dilakukan secara manual.

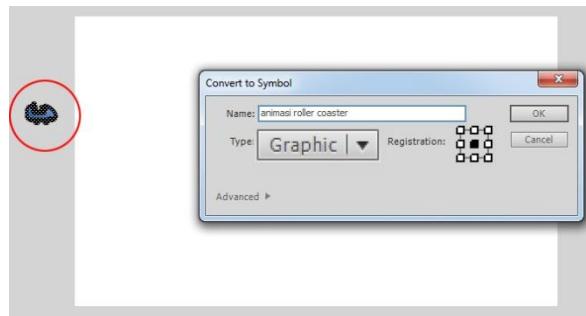
Meskipun terdapat perbedaan, pada dasarnya masing-masing symbol dapat diganti type nya melalui panel **Properties**, sehingga kita dapat menyesuaikan dengan kebutuhan. Untuk lebih memahami konsep animasi dengan metode graphic, perhatikan contoh membuat animasi roller coaster sebagai berikut :

1. Buatlah sebuah file baru, kemudian dengan menggunakan Drawing tools buatlah sebuah gambar gerbang kereta roller coaster seperti pada gambar.



Gambar 99. Gerbong roller coaster

2. Seleksi gambar tersebut, kemudian letakkan di sebelah kiri luar Stage. Selanjutnya **Convert to Symbol** menjadi **graphic** “animasi roller coaster”.



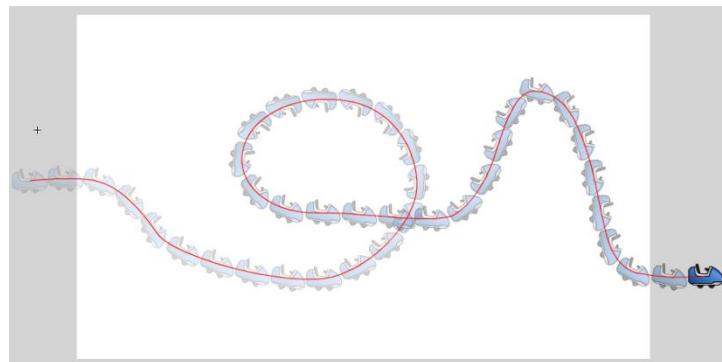
Gambar 100. Posisi dan convert to symbol graphic “animasi roller coaster”

3. Dobel klik **graphic** “animasi roller coaster” untuk memasuki **mode edit symbol**. Selanjutnya buatlah sebuah animasi **motion guide** pada timeline dengan durasi 200 frame. Untuk melakukannya **Convert to Symbol** terlebih dahulu gambar gerbong menjadi **movieclip** “gerbong”. (Perhatikan bahwa meskipun sebelumnya kita sudah mengconvert gerbong menjadi graphic “animasi roller coaster”, namun ketika memasuki mode edit symbol, gambar gerbong tersebut perlu di convert sekali lagi agar dapat dianimasikan)



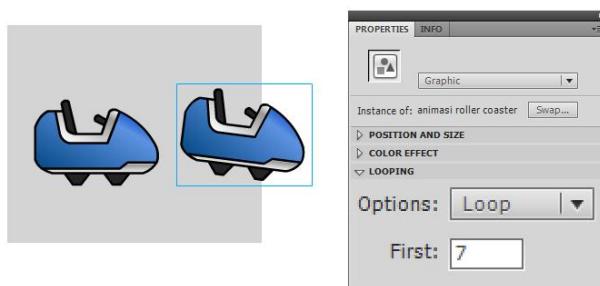
Gambar 101. Mengconvert kembali gerbong menjadi movieclip

4. Untuk membuat animasi **motion guide**, **klik frame 200**, kemudian tambahkan **Keyframe**. Selanjutnya tambahkan **Layer Guide** dan buatlah sebuah lintasan roller coaster dengan Pencil tool. Drag “gerbong” yang berada di frame 1 ke awal lintasan dan “gerbong” yang berada di frame 200 ke akhir lintasan. **Klik kanan frame 100 layer 1** dan pilih **Create Classic Tween**, dan aktifkan opsi **Orient to Path** pada panel **properties** agar gerbong bergerak dan berotasi mengikuti pola lintasan.



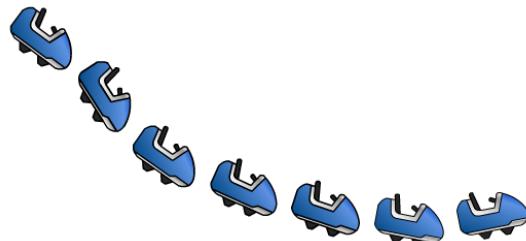
Gambar 102. Animasi motion guide roller coaster

5. Kembali ke **Scene** utama dengan menekan **Ctrl+E**. Kemudian jalankan aplikasi dengan menekan **Ctrl+Enter**. Maka tidak akan muncul animasi yang diharapkan. Hal ini dikarenakan pada **Scene 1 hanya ada 1 frame** saja, sedangkan graphic “animasi roller coaster” memiliki durasi 200 frame. Sesuai dengan konsep sebelumnya bahwa sebuah **graphic membutuhkan durasi frame** untuk menampilkan animasi yang ada di dalamnya.
6. Untuk menampilkan graphic “animasi roller coaster” seutuhnya, **klik kanan frame 200** dan pilih **Insert Frame** (Perhatikan bahwa disini tidak ada perubahan apapun, hanya penambahan durasi, sehingga cukup ditambahkan frame tidak perlu keyframe). Jalankan kembali aplikasi dan akan muncul animasi seutuhnya. Berbeda dengan Movieclip, ketika anda tekan tombol **Enter** (Play), maka animasi akan tampak di **Stage**. Selain itu apabila anda seleksi graphic “animasi roller coaster” dan anda perhatikan panel **properties**, maka akan terdapat opsi khusus yaitu opsi **Looping**.
7. Untuk menambahkan gerbong ke dua, **Copy** graphic “animasi roller coaster”, kemudian **Paste in Place (Ctrl+Shift+V)**. Klik gerbong ke 2 tersebut, kemudian pada panel **Properties** pada bagian **Looping** ubah parameter **First** menjadi **7**. Maksud dari pengaturan ini adalah untuk mengatur waktu dijalankannya animasi, yaitu dimulai dari frame ke 7.



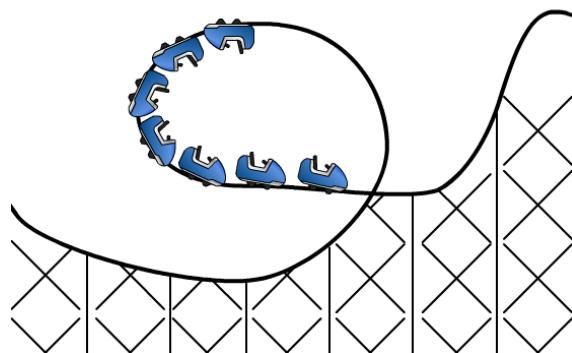
Gambar 103. Pengaturan gerbong ke 2

- Untuk gerbong ke tiga, lakukan **Copy Paste In Place** lagi, dan ubah First menjadi **13**. Lakukan beberapa kali dengan nilai First bekelipatan yang sama sehingga didapatkan beberapa gerbong roller coaster yang saling bersambungan. (Perlu diperhatikan bahwa dalam 1 keyframe ini dapat terdapat beberapa symbol sekaligus, hal ini dikarenakan tidak adanya penambahan animasi tween, karena animasi tween sudah berada di dalam simbol graphic).



Gambar 104. Hasil duplikasi graphic “animasi roller coaster”

- Sebagai sentuhan akhir, **buatlah sebuah layer baru** dan **buatlah gambar lintasan (rel)** seperti pada gambar.

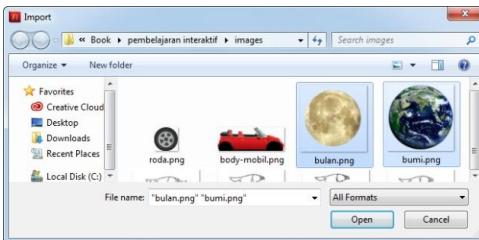


Gambar 105. Penambahan rel (lintasan)

Memanfaatkan Layer dalam Animasi

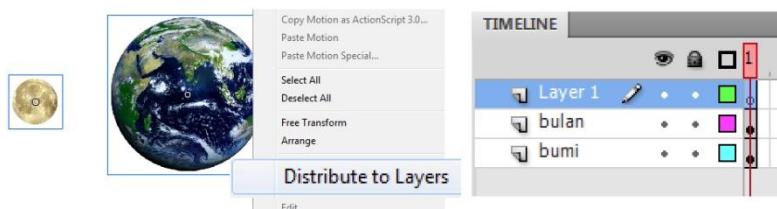
Layer pada dasarnya adalah lapisan yang dapat kita susun berdasarkan urutan tertentu. Semakin tinggi posisi suatu layer dalam Flash, berarti objek yang berada di layer tersebut akan ditampilkan paling depan. Dengan menggunakan prinsip tersebut, kita dapat memanfaatkannya untuk beberapa keperluan. Dalam contoh selanjutnya, akan dibuat sebuah animasi Bulan mengitari Bumi. Hal yang menarik dalam animasi tersebut adalah posisi Bulan dimana pada suatu saat berada di depan Bumi, dan pada suatu saat dibelakannya. Untuk lebih jelasnya perhatikan contoh berikut :

- Buatlah sebuah file baru, kemudian import 2 buah gambar yaitu gambar Bumi dan Bulan ke stage dengan memilih menu **File > Import > Import to Stage**



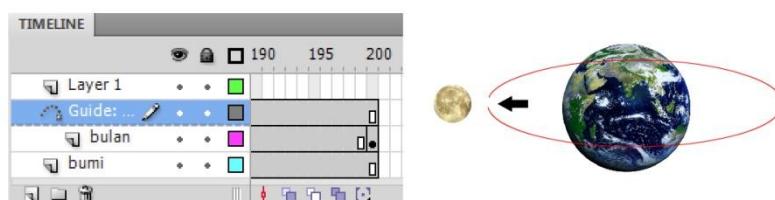
Gambar 106. Import to Stage

- Selanjutnya **Convert to Symbol** gambar tersebut menjadi movieclip “Bumi” dan movieclip “Bulan”. Untuk mendistribusikan masing-masing symbol ke layer baru, seleksi kedua movieclip, klik kanan dan pilih opsi **Distribute to Layers**. Pastikan layer bumi pada posisi paling bawah.



Gambar 107. Distribute to Layers

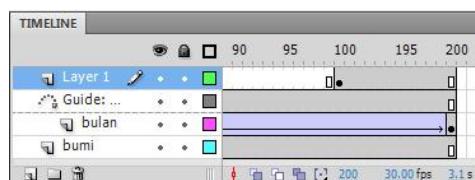
- Untuk membuat animasi bulan mengitari bumi dapat digunakan teknik **motion guide**. Misal durasi dari animasi adalah 200 frame, maka **klik kanan frame 200 layer bumi** dan pilih **Insert Frame** (hanya penambahan durasi tanpa perubahan apapun, sehingga menggunakan Insert Frame). Selanjutnya **klik kanan frame 200 layer bulan** dan pilih **Insert KeyFrame** (karena bulan mengalami perubahan/animasi, maka digunakan keyframe).
- Untuk menambahkan pola lintasan bulan **klik kanan layer bulan** dan pilih **Add Classic Guide**. Kemudian **klik frame 1 layer Guide**, dan buatlah lintasan bulan dengan menggunakan **Oval Tool** (hanya outlinenya saja, dan hapus fill nya). Dengan menggunakan **Erase Tool**, hapus sedikit dibagian kiri oval agar lintasan memiliki awal dan akhir. Lihat gambar!



Gambar 108. Struktur layer

5. Klik **frame 1 layer bulan**, kemudian **drag movieclip bulan ke awal lintasan**. Selanjutnya **klik frame 200 layer bulan** dan **drag movieclip bulan ke akhir lintasan**. Untuk menganimasikan movieclip bulan, **klik kanan frame 100 layer bulan** dan pilih **Add Classic Tween**.
6. Jalankan animasi dengan menekan tombol **Ctrl+Enter**. Bulan akan bergerak pada lintasan oval, akan tetapi posisi bulan selalu berada di atas bumi. Hal ini dikarenakan layer bulan berada di layer bumi. Hal tersebut dapat kita manipulasi dengan sedikit trik. Jika kita perhatikan pada saat bulan berada di frame 1 sampai frame 100 posisi bulan sudah benar yaitu di depan bumi, akan tetapi pada frame 101 sampai 200 posisi bulan seharusnya berada di belakang bumi. Berarti pada saat bulan melewati frame 100 layer bumi harus digeser ke atas.

Untuk melakukan hal tersebut, kita dapat memanipulasi layer. Klik **kanan frame 100 layer 1** (layer paling atas), kemudian pilih **Insert Blank Keyframe**. Selanjutnya **klik kanan movieclip “bumi”** dan pilih **Copy**, lalu **klik Blank Keyframe 100 layer 1**. **Klik kanan Stage** dan pilih **Paste in Place**. Sekarang movieclip Bumi ada di layer paling atas (layer 1). **Klik kanan frame 200 layer 1**, dan pilih **Insert Frame** agar bumi di layer 1 terlihat sampai frame 200.



Gambar 109. Struktur layer

7. Jalankan animasi dengan menekan tombol **Ctrl+Enter**. Maka bulan akan mengitari bumi dengan benar.



Gambar 110. Posisi bulan saat di depan dan di belakang bumi

Memahami Masking

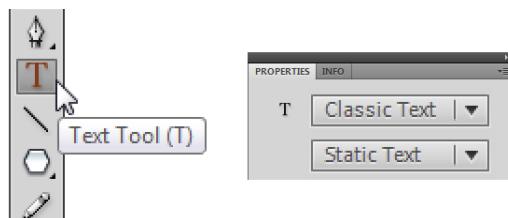
Masking adalah menampilkan suatu area dan menyembunyikan area lain dengan metode menutupi dengan sebuah objek di atasnya. Dalam Flash, sebuah objek dalam sebuah layer yang di tutupi oleh layer mask akan tampak, sebaliknya yang tidak tertutupi akan tidak tampak. Untuk jelasnya perhatikan gambar berikut :



Gambar 111. Masking

Pada contoh berikut teknik masking digunakan untuk menambahkan efek kilau cahaya pada sebuah teks :

1. Buatlah sebuah file baru, dengan menggunakan **Text Tool** mode **Classic Text - Static Text** buatlah sebuah tulisan “MULTIMEDIA”.



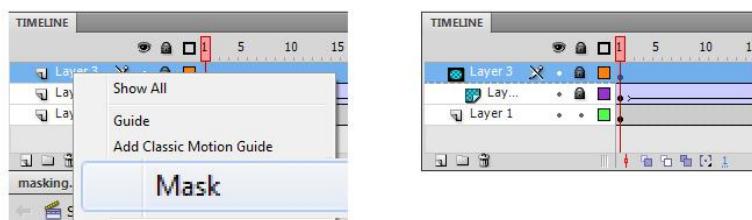
Gambar 112. Text Tool

2. **Klik kanan frame 50 layer 1**, kemudian berikan durasi dengan memilih **Insert Frame**.
3. Buatlah sebuah layer baru (layer 2) di atas layer 1. Pada layer 2 tersebut, buatlah sebuah animasi kotak putih yang bergerak dari kiri teks ke kanan teks dengan langkah sebagai berikut : Dengan menggunakan Rectangle Tool buatlah sebuah kotak berwarna putih (terang) di sebelah kiri teks lalu convert menjadi movieclip “kotak”. Klik kanan frame 50 layer 2 lalu tambahkan Keyframe. Geser kotak tersebut ke kanan teks. Klik kanan frame 20 dan pilih Create Classic Tween, maka kotak akan bergerak dari kiri ke kanan.



Gambar 113. Posisi movieclip “kotak” terhadap teks

4. Klik kanan teks “MULTIMEDIA” lalu pilih **Copy**. Selanjutnya buatlah sebuah **layer baru** (layer 3) dan letakkan di posisi paling atas. **Klik frame 1 layer 3**, kemudian klik kanan **Stage** dan pilih **Paste in Place** untuk meletakkan teks tepat di atas teks yang berada di layer 1.
5. Selanjutnya klik kanan **layer 3**, dan pilih opsi **Mask**. Maka tampilan layer pada timeline akan berubah, dimana layer 2 termasking oleh layer 3.



Gambar 114. Masking

6. Jalankan animasi dengan menekan tombol **Ctrl+Enter**. Maka gerakan movieclip kotak akan terbingkai oleh huruf, sehingga seolah-olah menghasilkan sebuah efek *highlight / cahaya*.



Gambar 115. Hasil efek kilau dengan masking

Membuat Animasi Bumi Berotasi

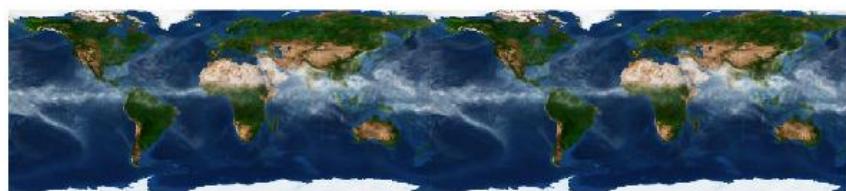
Salah satu manfaat penggunaan masking adalah untuk membuat sebuah animasi yang melibatkan perulangan konten, sebagai contoh adalah animasi bumi berotasi. Flash pada tahun 2012 memiliki fitur tambahan yaitu pengolahan grafis berbasis *hardware* sehingga memungkinkan pengolahan grafis 3 dimensional, dalam artian kita dapat membuat animasi bumi berputar dalam format 3 dimensi. Namun demikian, proses animasi bumi berotasi disini dapat kita buat dengan menggunakan teknik masking dan bersifat 2 dimensional. Untuk lebih jelasnya perhatikan langkah-langkah berikut:

1. Siapkan sebuah gambar peta dunia yang dapat dengan mudah kita dapatkan via internet, sebagai berikut :



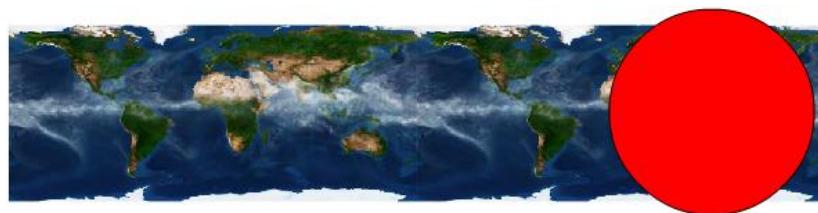
Gambar 116. Peta dunia

2. Selanjutnya buat file baru, dan **Import to Stage** gambar tersebut.
3. Untuk membuat gerakan bumi yang berputar secara penuh, maka diperlukan 2 buah gambar. **Copy** gambar peta dunia tersebut, dan **Paste**. Lalu atur posisinya saling berurutan seperti pada gambar berikut :



Gambar 117. Pengaturan peta

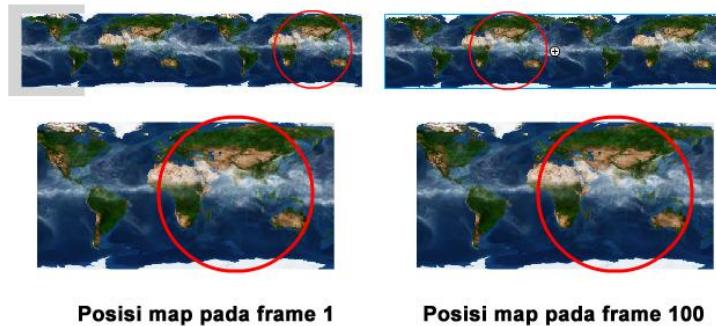
4. Seleksi kedua gambar peta tersebut, kemudian **Convert to Symbol** menjadi movieclip “peta”.
5. Buatlah sebuah layer baru (layer 2). Pada **frame 1 layer 2**, buatlah sebuah lingkaran di tengah stage sebagai bentuk dasar bumi dengan menggunakan **Oval Tool**. **Klik kanan frame 100 layer 2** dan pilih **Insert Frame**, untuk menambahkan durasi.



Gambar 118. Bentuk dasar bumi

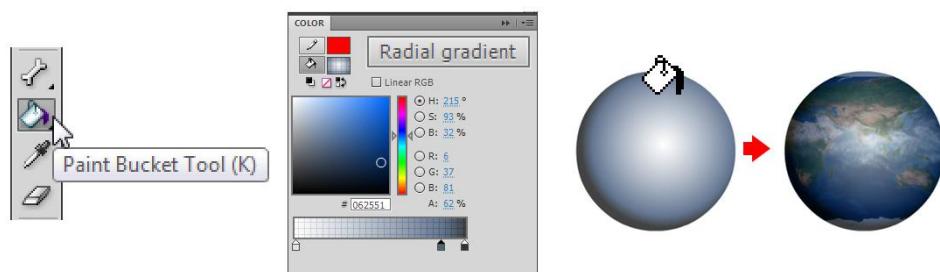
6. Klik **frame 1 layer 1**, kemudian geser posisi movieclip “bumi” ke sebelah kiri dari lingkaran. Selanjutnya **klik kanan frame 100 layer 1** dan pilih **Insert Keyframe**, lalu geser movieclip bumi ke sebelah kiri. Perhatikan prinsip berikut :

- Bumi berputar dari barat ke timur, sehingga peta digerakkan dari kiri ke kanan.
- Posisi peta pada frame 1 (awal) dan frame 100 (akhir) harus sama, agar rotasi tidak putus.



Gambar 119. Posisi map terhadap lingkaran

7. **Klik kanan frame 50 layer 1**, kemudian pilih **Create Classic Tween**. Maka peta akan bergerak. Selanjutnya tambahkan Masking dengan cara **klik kanan layer 2** dan pilih opsi **Mask**. Jalankan animasi dengan menekan tombol **Ctrl+Enter**, maka akan di dapatkan animasi bumi berputar.
8. Untuk mempercantik tampilan, **Copy** lingkaran pada layer 2, selanjutnya buat layer baru (layer 3) di atas layer 2 dan **Paste in Place**. Selanjutnya ubah warna lingkaran tersebut menjadi **Radial Gradient** dengan pengaturan warna transparan, sehingga bumi seolah-olah memiliki kesan 3 dimensi.

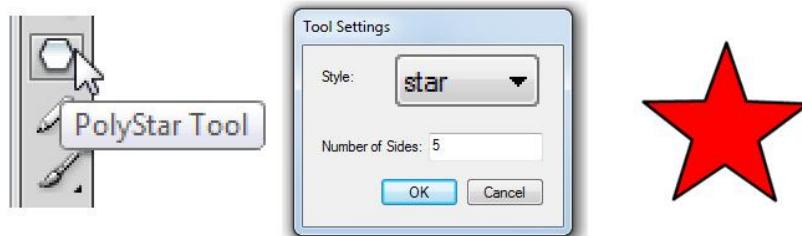


Gambar 120. Pengaturan warna gradasi

Animasi Shape Tween

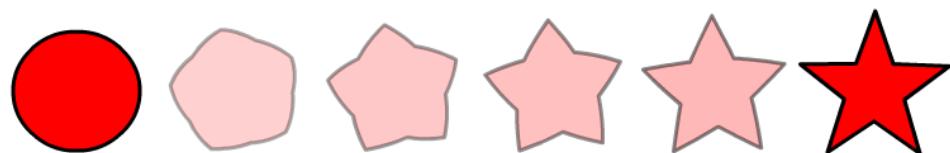
Animasi Shape Tween adalah animasi berdasarkan perubahan bentuk objek. Sebagai contoh jika ingin membuat animasi transformasi bentuk lingkaran menjadi bentuk bintang, maka dapat digunakan teknik shape tween. Perhatikan contoh berikut :

1. Buatlah sebuah file baru, kemudian buatlah sebuah lingkaran dengan menggunakan **Oval Tool**.
2. **Klik kanan frame 25**, kemudian pilih **Insert Blank Keyframe**. Lalu dengan menggunakan **PolyStar Tool**, atur option **Tool Settings** di panel **properties** dan buatlah sebuah gambar bintang.



Gambar 121. PolyStar Tool

3. **Klik kanan frame 10**, kemudian pilih **Create Shape Tween**. Jalankan animasi dengan menekan tombol **Ctrl+Enter**, maka akan didapati sebuah transformasi lingkaran menjadi bintang.



Gambar 122. Hasil animasi shape tween.

Bab VI

Dasar Pemrograman Action Script 3

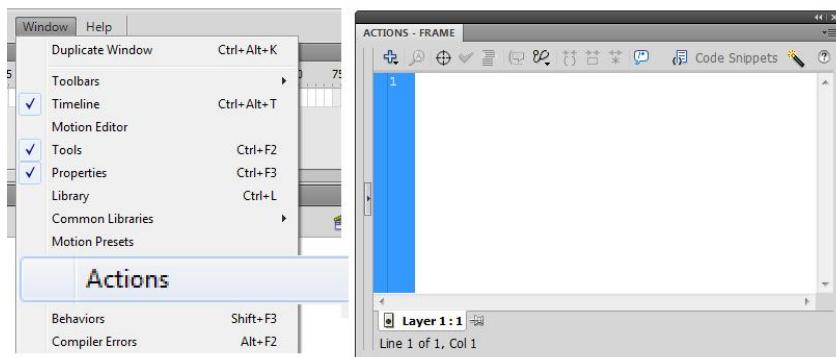
Untuk membentuk sebuah interaktifitas dalam sebuah multimedia pembelajaran interaktif diperlukan tahapan pemrograman. Pemrograman adalah suatu kegiatan menuliskan kumpulan urutan perintah ke komputer untuk mengerjakan sesuatu, dimana instruksi tersebut menggunakan bahasa yang dimengerti oleh komputer atau dikenal dengan bahasa pemrograman. Adobe Flash memiliki bahasa pemrograman yang disebut sebagai Action Script, dan sejak tahun 2007 standart industri yang digunakan adalah Actionscript 3.0.

Untuk memahami dengan baik konsep dasar pemrograman Actionscript 3, maka diperlukan pemahaman secara bertahap. Di lapangan sering ditemui seorang pemula yang sudah memiliki konsep terlalu kompleks sehingga dalam proses membangun aplikasi akan menemui banyak kendala. Seyogyanya mempelajari sebuah kode pemrograman harus dimulai dari teknik yang paling dasar terlebih dahulu. Seiring waktu belajar, maka kemampuan seseorang akan meningkat dan mampu mengerjakan projek yang lebih kompleks.

Mencoba Kode Actionscript

Dalam Flash kode Actionscript dituliskan di action panel. Pada saat aplikasi dijalankan, maka compiler akan mengecek kode baris demi baris mulai dari yang teratas sampai kode baris terakhir. Dari proses pengecekan tersebut, apabila ditemukan kesalahan maka proses akan berhenti dan kesalahan akan ditampilkan. Hal ini tentu saja mempermudah seorang pengembang aplikasi untuk menyempurnakan kode. Sebagai contoh perhatikan aplikasi sederhana sebagai berikut :

1. Buatlah sebuah file baru. Kemudian klik frame 1 layer 1 dan buka action panel dengan menekan menu **Window > Action (F9)**.
2. Dalam Flash Actionscript 3 hanya dapat diletakkan di sebuah frame atau disebuah file terpisah (dengan metode Class), berbeda dengan versi sebelumnya yang dapat dituliskan di dalam movieclip dan tombol. Oleh karena itu perlu selalu diingat bahwa sebelum menuliskan kode, terlebih dahulu harus menyeleksi Keyframe tempat kode tersebut akan dijalankan.

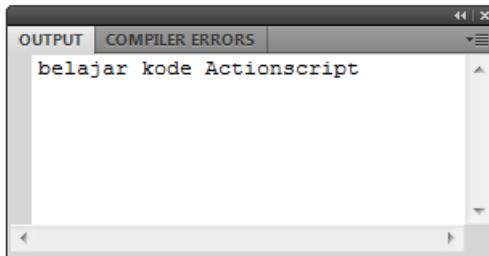


Gambar123. Panel Action

- Klik panel action, dan ketikkan kode berikut pada editor kode.

```
trace ("belajar kode Actionscript");
```

- Jalankan aplikasi dengan menekan **Ctrl+Enter**. Maka akan muncul panel **Output** dengan tulisan “belajar kode Actionscript”. Hal ini dikarenakan kode trace berfungsi untuk menampilkan parameter yang ada di dalam tanda (); yang dalam hal ini parameter tersebut berisi sebuah tulisan.

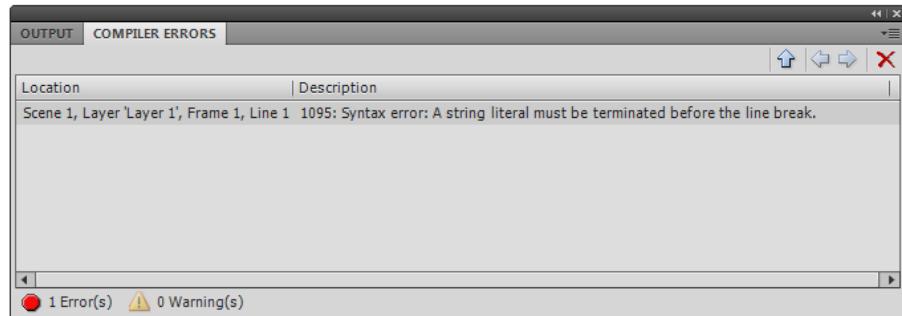


Gambar 124. Hasil pada output panel.

- Selanjutnya ubah sedikit kode tersebut dengan menghapus salah satu tanda petik, misalkan menjadi

```
trace ("belajar kode Actionscript);
```

Selanjutnya jalankan kembali aplikasi dengan menekan **Ctrl+Enter**. Maka akan muncul panel **Compiler Errors** dengan menampilkan pesan khusus bahwa terjadi kesalahan pada kode yaitu kesalahan sintaks (Syntax Error) : A String literal must be terminated before the line break atau dapat diartikan sederhana bahwa sebuah variabel berupa String harus diakhiri dengan tanda “.



Gambar 125. Tampilan Compiler Errors.

Dari percobaan di atas, diketahui bahwa penulisan sebuah kode sangatlah spesifik dan detail. Sebuah kesalahan kecil saja bisa berakibat berhentinya aplikasi atau gagalnya proses compiling kode. Selain itu juga terdapat istilah-istilah seperti **Syntax**, **String** dan sebagainya yang bagi seseorang yang baru belajar pemrograman akan cukup membingungkan. Untuk itu pada sub bab berikut akan dijelaskan secara mendetail tentang fundamental dalam mempelajari Actionscript 3.

Syntax

Dalam pemrograman komputer sintaks berarti seperangkat aturan yang harus dipenuhi dalam penulisan kode. Sintaks meliputi simbol, kata, dan fungsi yang dapat digunakan serta bagaimana menggunakannya dalam sebuah struktur kode. Kesalahan umum yang sering dilakukan oleh programer adalah kesalahan sintaks (Syntax error), dimana kesalahan sintaks akan menyebabkan kode tidak bisa dijalankan oleh aplikasi. Berikut adalah aturan dalam penulisan Actionscript 3 (AS3).

1. Case Sensitive

Penulisan kode AS3 menggunakan format sensitif case, sehingga setiap kode harus ditulis tepat sebagaimana mestinya. Satu perbedaan karakter akan menghasilkan perbedaan, sebagai contoh 2 baris kode berikut berbeda.

```
1. var bilanganGenap:int;  
2. var BilanganGenap:int;
```

2. Semicolons

Semicolons (;) atau titik koma dalam AS3 digunakan sebagai akhir dari sebuah baris kode. Meskipun dituliskan dalam satu baris, jika ditambahkan tanda titik koma dapat diartikan sebagai baris perintah baru

```
1. kode baris 1;  
2. kode baris 2; kode baris 3;
```

3. Parentheses

Parentheses atau tanda kurung () digunakan untuk mengubah urutan dalam sebuah operasi/fungsi. Operasi yang diberikan tanda kurung akan dijalankan terlebih dahulu oleh program. Perhatikan contoh berikut :

```
1. trace(2 + 3 * 4); // akan menghasilkan nilai 14  
2. trace((2 + 3) * 4); // akan menghasilkan nilai 20
```

Parentheses juga dapat digunakan untuk memberikan masukan pada sebuah fungsi. Perhatikan contoh berikut :

```
1. trace("Belajar AS3"); // menghasilkan kata Belajar AS3  
2. trace(kuadrat(4)); // menghasilkan nilai 16
```

4. Code blocks

Dalam pemrograman AS3, satu baris atau lebih yang terletak di dalam kurung kurawa { } disebut sebagai satu blok kode. Satu blok kode merupakan sebuah paket yang biasanya digunakan dalam sebuah fungsi, class, loop dan sebagainya. Perhatikan contoh berikut :

```
1. function kuadrat(num:Number):Number{  
2.     return num*num;  
3. }  
4.  
5. for(var i:uint=1; i<10; i++){  
6.     var hasil_kuadrat:Number = kuadrat(i);  
7.     trace("hasil kuadrat "+i+" = "+hasil_kuadrat);  
8. }
```

5. Whitespace

Whitespace merupakan istilah untuk jarak penulisan, tab, enter dan spasi yang ditujukan untuk mempermudah penulisan kode. Perhatikan kode di atas, pada baris 2 ditulis menjorok masuk, pada baris 4 sengaja dikosongkan agar terdapat jarak antara fungsi kuadrat dengan operasi loop di baris 5 – hal tersebut diistilahkan sebagai whitespace (ruang kosong). Perhatikan kode di bawah berikut ini, yang bermakna sama namun ditulis dengan cara yang berbeda.

```
1. for(var i:uint=0; i<10; i++) {
```

```
2.     trace(i);  
3. }  
4.  
5. for(var i:uint=0; i<10; i++) {trace(i);}
```

6. Comments

Comments atau komentar merupakan catatan yang bisa ditambahkan pada kode untuk mempermudah penjelasan kode. Ketika menulis kode dalam jumlah banyak (misal 2000 baris kode) dibutuhkan penjelasan-penjelasan pada baris kode untuk mempermudah proses pengecekan. Program secara otomatis akan mengabaikan baris komentar, sehingga kita tidak perlu khawatir dalam menggunakan komentar. Perhatikan contoh berikut :

```
1. trace("comment"); // komentar dalam 1 baris  
2.  
3. /* gunakan tanda ini jika ingin  
4. komentar lebih dari satu baris */
```

Kegunaan lain dari komentar yang umum digunakan oleh programer adalah untuk menonaktifkan sementara kode yang tidak dibutuhkan. Pada saat proses pengecekan kode yang panjang (debugging), sering kali beberapa baris kode dinonaktifkan sementara untuk menguji kesalahan/kebenarannya.

7. Literals

Literals adalah nilai yang kita tuliskan ke kode dalam bentuk angka, string atau array.

Misal 1, 832009, "kata-kata", [4, 5, 6] dan sebagainya.

8. Keywords and reserved words

Merupakan kata kunci dan kata-kata khusus yang telah dipakai oleh kode, sehingga tidak bisa digunakan sebagai identifier. AS3 memiliki beberapa keyword dan kata khusus yang pada panel actionscript akan diberi warna berbeda dan akan diberikan peringatan apabila kita memakainya. Terdapat 3 jenis keyword yaitu :

a. Lexical Keywords

Merupakan kata-kata yang sudah digunakan dalam rangkaian kode.

Misalnya as if return break super case

b. Syntactic Keywords

Merupakan kata-kata yang dapat digunakan sebagai identifier, namun memiliki fungsi tertentu jika digunakan dalam konteks yang benar. Misalnya

each include override get dynamic static

c. Future Reserved Word

Merupakan kata-kata yang disimpan untuk fitur actionscript versi selanjutnya. Misalnya abstract export throws float to byte

Variable

Sistem kerja komputer adalah dengan menyimpan data dan mengolahnya. Dalam mengelola data komputer akan meyimpan sebuah nilai ke dalam memori, dimana seorang programer dapat mengatur nilai tersebut dan memberikannya nama. Nama dan nilai yang akan disimpan ke dalam memory itulah yang disebut sebagai variabel. Sebagai contoh, ketika kita menulisakan kode var waktu:Number = 60; ini berarti komputer akan menyimpan sebuah data bernama waktu bertipe Number (angka) dan bernilai 60. Variabel ini akan terus disimpan oleh memori komputer sepanjang aplikasi berjalan.

Dalam menuliskan nama dan dilai variabel terdapat aturan yang harus dipenuhi, yaitu :

1. Tidak boleh menggunakan keywords atau reserved words (lihat point 8 subbab sebelumnya)
2. Harus dimulai dengan huruf, underscore (_) atau tanda dollar (\$)
3. Masing-masing variabel harus unik (berbeda antara satu dengan yang lain)
4. Sebaiknya menggunakan kata-kata yang mudah dipahami, misal
`var bilanganGenap : number = 2;`
5. Lebih baik menggunakan sistem *camelCase* atau membedakan kata dengan huruf kapital pada huruf pertama setiap kata.
6. Underscore (_) hanya boleh untuk variabel yang bersifat *private*.
7. Tanda dollar (\$) hanya boleh untuk variabel yang bersifat *static*.

Access control attributes

Dalam pemrograman berbasis objek (OOP), masing-masing variabel, fungsi, metode dan class dapat diakses secara khusus dengan menggunakan Access control attributes. Terdapat 4 jenis Access control attributes yaitu `Private`, `Public`, `Protected` dan `Internal`. Dalam buku ini tidak digunakan metode tersebut, sehingga tidak dibahas secara mendalam.

Mendeklarasikan Variabel

Dalam mendeklarasikan variabel, AS3 menggunakan kata kunci `var`, diikuti dengan nama variabel, jenis data dan nilai variabel. Perhatikan contoh berikut :

```
var nama_siswa: String = "Azka";
```

`var` merupakan deklarasi variabel sehingga komputer mengalokasikan memorinya, `nama_siswa` merupakan nama variabel yang akan disimpan ke dalam memori. `String` merupakan jenis data dan “`Azka`” adalah nilai dari variabel tersebut.

Constants

Berbeda dengan variabel yang bersifat dinamis, konstanta bersifat statis sepanjang aplikasi berjalan. Nilai konstanta akan selalu tetap dan tidak dapat diubah kembali oleh kode atau fungsi setelah pendeklarasiannya. Konstanta dalam AS3 dideklarasikan dengan keyword `const`. Contoh :

```
const gravitasi:Number = 0.98; // mengeset nilai grafitasi
```

Data Types

Data types atau jenis data adalah beberapa klasifikasi data yang digunakan dalam pemrograman AS3. Secara mendasar jenis data di AS3 ada 7 yaitu `String`, `Number`, `int`, `uint`, `Boolean`, `Null`, dan `void`.

1. String

`String` adalah data yang bernilai textual dan penulisannya dalam AS3 menggunakan tanda petik, misalnya

```
var nama:String = "Izzan";
```

2. Number

Number adalah data yang bernilai angka tanpa batasan. Misal :

```
var nilaiUjian:Number = 97;
```

3. Int

Int adalah kependekan dari integer yaitu data yang bernilai angka dari -2,147,483,648 sampai dengan 2,147,483,647 (32 bit data). Penggunaan variable bertipe int lebih efisien dari segi kecepatan pengolahan data dibanding dengan variabel bernilai Number.

4. UInt

Berbeda dengan int yang bernilai negatif sampai positif, uint hanya bernilai positif yaitu mulai dari angka 0 sampai dengan 4,294,967,295.

5. Boolean

Boolean merupakan nilai benar atau salah. Dalam AS 3 bernilai `true` atau `false`.

6. Null

Null merupakan nilai kosong atau tidak bernilai, namun bukan tidak terdefinisikan (*undefined*).

7. Void

Void adalah sesuatu yang tidak menghasilkan nilai, pada umumnya digunakan untuk fungsi yang tidak mengembalikan/menghasilkan nilai spesifik.

Pada pengembangan berikutnya akan ditemukan tipe data yang lebih kompleks seperti `Array`, `Object`, `Date` dan sebagainya.

Operators

Operators merupakan simbol yang digunakan untuk operasi matematika seperti penjumlahan (+), pengurangan (-), perkalian (*), pembagian (/), sama dengan (=) dan sebagainya. Perhatikan contoh berikut :

```
1. trace (4*5+8); //menghasilkan 28
```

Function

Function atau fungsi merupakan satu atau beberapa baris kode dalam satu blok, yang ditujukan untuk menyederhanakan suatu operasi dan dapat digunakan beberapa kali. Sebagai contoh, ketika membuat aplikasi yang melibatkan perhitungan jarak antara objek dan mouse, maka fungsi menghitung jarak tersebut ditulis dalam sebuah blok kode yang berisi beberapa baris. Untuk menghitung jarak tentunya ada beberapa variabel yang diperlukan yaitu titik kordinat awal dan titik kordinat akhir, selanjutnya dilakukan perhitungan menggunakan rumus pitagoras dan terakhir hasil perhitungan ditampilkan oleh fungsi.

Mendefinisikan sebuah fungsi di dalam AS3 memiliki beberapa elemen sebagai berikut :

1. Deklarasi keyword `function`
2. Pemberian nama fungsi
3. Parameter (variabel) awal sebagai masukan
4. Hasil dari fungsi (`return`)
5. Blok kode fungsi (function body)

Untuk mempermudah pemahaman terhadap fungsi perhatikan beberapa contoh berikut :

```
1. function haloDunia():void{
2.     trace("Hello world");
3. }
4.
5. haloDunia(); // Hello world
```

Pada fungsi `haloDunia` di atas, berisi satu baris kode yaitu kode `trace`. Kode trace digunakan untuk menampilkan hasil di output panel. Sehingga jika kode tersebut dijalankan, pada output panel akan muncul teks “Hello world”.

```
1. function kuadrat(angka:int):int{
2.     return(angka*angka);
3. }
4.
5. trace(kuadrat(5)); //fungsi kuadrat 5 akan menghasilkan 25
6. trace(kuadrat(6)); //fungsi kuadrat 6 akan menghasilkan 36
```

Pada fungsi `kuadrat` tersebut, terdapat parameter `angka` yang selanjutnya diproses oleh kode blok dimana hasil dari fungsi adalah `angka*angka`. Berbeda dengan fungsi

sebelumnya yang tidak menghasilkan apa-apa (`void`), fungsi kuadrat menghasilkan sebuah nilai yang bertipe int (`:int`). Sebuah fungsi baru akan dijalankan jika fungsi tersebut dipanggil/diaktifkan. Sebagai contoh di atas fungsi kuadrat dipanggil 2 kali pada baris ke 5 dan baris ke 6.

Loops

Loops atau operasi berulang adalah satu blok kode yang dijalankan secara terus menerus sampai kondisi yang ditentukan terpenuhi. Misalnya menghitung angka mulai dari angka 1 sampai 10, menampilkan data siswa satu kelas, mensortir nilai ujian beberapa peserta sekaligus dan sebagainya. Secara umum loops dalam AS3 terbagi menjadi 3 yaitu `for`, `while` dan `do..while`. Perhatikan struktur operasi berulang `for` berikut ini :

```
1. for(var i:int=0; i<10; i++) {  
2.     trace(i);  
3. }
```

`for` merupakan keyword yang digunakan untuk operasi berulang

`var i:int = 0;` disebut sebagai *inisiasi* atau kondisi awal

`i<10;` disebut sebagai *test* atau kondisi dimana operasi berulang akan terhenti bila sudah tercapai

`i++;` disebut sebagai *update* atau kondisi yang terjadi pada variabel penguji loop.

`{ ... }` disebut sebagai *loop body* atau blok kode yang dijalankan sepanjang loop

Apabila kode tersebut dijalankan akan menghasilkan angka 0 sampai 9 di layar. Output 0 karena merupakan kondisi awal saat inisiasi, selanjutnya bertambah 1 demi 1 karena ada update `i++`, yang sama artinya dengan `i = i + 1;` dan diakhiri pada `i = 9` karena tepat pada kondisi `i < 10;`

Sejenis dengan `for` loop, operasi `while` loops hampir sama dengan operasi `for` hanya saja tidak memiliki perintah untuk update, atau dengan kata lain blok kode akan dijalankan secara terus menerus sampai kondisi terpenuhi. Untuk lebih jelasnya perhatikan contoh berikut :

```
1. var angka:int = 0;  
2. while(angka < 10) {  
3.     angka ++;
```

```
4.     trace(angka);  
5. }
```

Apabila kode tersebut dijalankan akan menghasilkan angka 1 sampai 10 di layar. Output 1 karena kondisi awal saat inisiasi yaitu 0 bertambah 1 akibat kode angka++, selanjutnya bertambah 1 demi 1 karena adanya kode baris ke-3, dan diakhiri pada angka = 9 namun karena adanya kode angka++; maka nilai akhir dari angka adalah 10.

Sedangkan `do..while loops` adalah format lain dari operasi `while loops` namun test kondisi dilakukan di akhir. Perhatikan contoh berikut :

```
1. var angka:int = 0;  
2. do{  
3.     angka++;  
4.     trace(angka);  
5. }while(angka < 10);
```

Apabila kode tersebut dijalankan akan menghasilkan angka 1 sampai 10 di layar, dengan penjelasan sama dengan operasi `while loops`.

Break dan Continue

Break dan continue digunakan untuk menginterupsi sebuah operasi berulang (loops). Break digunakan untuk menghentikan operasi berulang ketika suatu kondisi dalam blok terpenuhi. Perhatikan contoh berikut :

```
1. for(var i:int=1; i<23; i+=4) {  
2.     if(i==13) {  
3.         break;  
4.     }  
5.     trace(i);  
6. }  
7. //output 1 5 9
```

Pada contoh di atas, operasi for seharusnya akan menaikkan nilai variabel i sebanyak 4 poin setiap langkah, sehingga menghasilkan angka 1, 5, 9, 13, 17, dan 20. Namun akibat kondisi (`i==13`) terpenuhi maka kode break dijalankan dan operasi berulang dihentikan sehingga hanya dihasilkan angka 1, 5 dan 9.

```
1. for(var i:int=1; i<23; i+=4) {  
2.     if(i==13) {  
3.         continue;  
4.     }
```

```
5.     trace(i);
6. }
7. //output 1 5 9 17 21
```

Berbeda dengan kode `break` yang menghentikan secara langsung. Kode `continue` akan melewati satu langkah apabila kondisi terpenuhi, maka pada kode di atas ketika (`i == 13`) terpenuhi perulangan dilewati satu langkah, sehingga angka 13 tidak muncul.

Conditional if

Kondisi `if` merupakan logika dasar sebagian besar bahasa pemrograman. Dengan kode `if` sebuah kondisi dicek kebenarannya dan blok kode akan dieksekusi berdasarkan kondisi tersebut. Perhatikan contoh berikut :

```
1. function bilanganGenap(angka:int):void{
2.     if (angka % 2 == 0) {
3.         trace(angka+" adalah bilangan genap");
4.     }
5. }
6.
7. bilanganGenap(6);
```

Pada baris 2 terdapat sebuah kondisi dimana angka mengalami operasi modulus (%) dengan angka 2. Apabila angka dibagi dengan bilangan 2 menyisakan bilangan 0 (disimbolkan dengan operator ==) maka angka tersebut dipastikan sebagai bilangan genap.

```
1.     function bilanganGenap(angka:int):void{
2.         if (angka % 2 == 0) {
3.             trace(angka+" adalah bilangan genap");
4.         }else{
5.             trace(angka+" adalah bilangan ganjil");
6.         }
7.     }
8.
9.     bilanganGenap(8); // 6 adalah bilangan genap
10.    bilanganGenap(3); // 6 adalah bilangan ganjil
```

Pada contoh di atas digunakan operasi kondisi `if .. else`. Pada kondisi ini, apabila kondisi bernilai salah, kode pada blok `else` akan dijalankan. Kondisi `if .. else` juga dapat dilakukan beberapa kali dengan kondisi yang berbeda-beda. Perhatikan contoh berikut:

```

1. function cekNilai(nilai:int):void{
2.     if (nilai < 6) {
3.         trace("Maaf anda tidak lulus");
4.     }else if (nilai < 8) {
5.         trace("Lumayan!, anda mendapat nilai C");
6.     }else if (nilai < 10) {
7.         trace("Hebat!, anda mendapat nilai B");
8.     }else{
9.         trace("Luar biasa!, nilai sempurna");
10.    }
11. }
12.
13. cekNilai(7); // Lumayan!, anda mendapat nilai C

```

Pada kondisi `if .. else if` di atas, pengecekan logika dimulai dari kondisi teratas dan apabila terpenuhi maka akan dilanjutkan ke logika berikutnya. Sehingga ketika kode `cekNilai(8);` dijalankan, maka kondisi `(nilai < 6)` terpenuhi, kondisi `(nilai < 8)`, sedangkan kondisi `(nilai < 10)` tidak terpenuhi. Untuk itu ditampilkan hasil kondisi terakhir yang terpenuhi yaitu `(nilai < 8)`.

Conditional switch

Kondisi switch digunakan ketika terdapat beberapa kondisi yang berbeda-beda. Perhatikan contoh kode berikut :

```

1. function cekNilai(nilai:String):void{
2.     switch(nilai){
3.         case "A":
4.             trace("Sempurna - anda mendapat nilai A");
5.             break;
6.         case "B":
7.             trace("Bagus - anda mendapat nilai B");
8.             break;
9.         case "C":
10.            trace("Lumayan - anda mendapat nilai C");
11.            break;
12.         case "D":
13.             trace("Lebih giat lagi - anda mendapat nilai
D");
14.             break;
15.         case "E":
16.             trace("Maaf anda tidak lulus");
17.             break;
18.         default:
19.             trace("Masukan data salah");
20.             break;
21.     }
22. }
23.

```

```
24. cekNilai("C");
```

Pada kode tersebut beberapa kondisi telah ditentukan, dan ketika kondisi yang dimaksud terpenuhi di salah satu kode `case`, maka kode akan dijalankan sampai dengan eksekusi kode `break`. Apabila tidak ada kondisi yang terpenuhi, maka blok kode `default` yang akan dijalankan.

Array

Array adalah sebuah struktur data yang bisa digunakan untuk menyimpan beberapa variabel dengan nama yang sama. Sebagai contoh data nama siswa kelas 1, data soal kuis dan sebagainya. Perhatikan contoh berikut :

```
1. var namaHari:Array = ["Senin", "Selasa", "Rabu", "Kamis",
   "Jumat", "Sabtu", "Minggu"];
2.
3. trace(namaHari[0]); //Senin
4. trace(namaHari[3]); //Kamis
```

Pada kode di atas dideklarasikan variabel `namaHari` dengan tipe `Array`. Variabel tersebut memiliki beberapa nilai yaitu “Senin” sampai “Minggu”. Untuk menggunakan data yang berada di dalam `Array`, dapat digunakan nama variable diikuti tanda [no data]. Contoh variabel `namaHari[0]` bernilai “Senin”, karena berada di urutan pertama (dimulai dari angka 0), dan seterusnya.

Mengendalikan Timeline dengan Actionscript

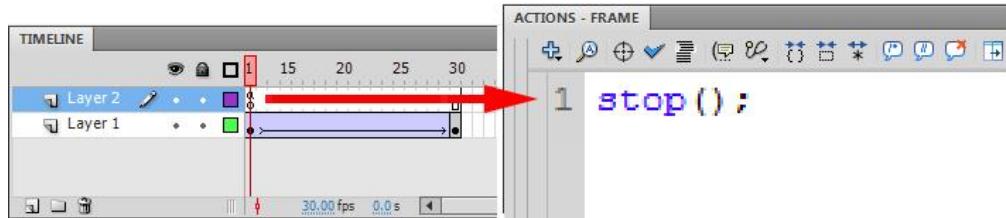
Salah satu kegunaan actionscript adalah untuk mengatur timeline. Dalam beberapa kasus kita memerlukan fitur untuk mengendalikan animasi yang kita buat, sebagai contoh dalam sebuah media pembelajaran terdapat animasi yang pada awalnya merupakan gambar diam, namun setelah tombol ditekan gambar tersebut bergerak – hal ini membutuhkan pengaturan yang melibatkan kode.

stop();

Kode `stop()` berfungsi untuk menghentikan timeline pada frame dimana lokasi kode tersebut berada. Untuk lebih jelasnya perhatikan contoh berikut :

- Buatlah sebuah file baru. Kemudian dengan teknik **Motion Tween**, buatlah sebuah animasi bola yang bergerak dari kiri ke kanan dengan durasi 30 frame.
- Buatlah sebuah layer baru (layer 2), selanjutnya klik **frame 1 layer 2**. Buka panel **Action (F9)**, kemudian ketikan kode

```
stop();
```



Gambar 126. Peletakan kode stop();

- Jalankan aplikasi dengan menekan **Ctrl+Enter**. Maka aplikasi akan memunculkan bola tersebut namun bola tidak bergerak. Hal tersebut dikarenakan kode `stop();` berjalan sehingga timeline berhenti.
- Klik frame 1 layer 2**, kemudian **drag Keyframe** tersebut menuju ke **frame 15**. Dengan kata lain kode `stop();` saat ini berada di frame 15. Jalankan kembali aplikasi dengan menekan **Ctrl+Enter**. Maka bola akan bergerak dan berhenti tepat ketika berada di posisi frame 15.

gotoAndStop();

Kode `gotoAndStop(frame);` berfungsi untuk melompat dan berhenti pada frame yang diset dalam parameter `frame`. Untuk lebih jelasnya perhatikan contoh berikut :

- Buatlah sebuah file baru. Kemudian dengan teknik **Motion Tween**, buatlah sebuah animasi bola yang bergerak dari kiri ke kanan dengan durasi 30 frame.
- Buatlah sebuah layer baru (layer 2), selanjutnya klik **frame 1 layer 2**. Buka panel **Action (F9)**, kemudian ketikan kode

```
gotoAndStop(15);
```
- Jalankan aplikasi dengan menekan **Ctrl+Enter**. Maka aplikasi akan memunculkan bola pada posisi frame 15 dan bola tidak bergerak.

gotoAndPlay();

Berkebalikan dengan kode `gotoAndStop(frame);` kode `gotoAndPlay(frame);` berfungsi untuk melompat dan memainkan timeline mulai dari frame yang diset dalam

parameter frame. Untuk lebih jelasnya, pada tutorial di atas edit kode `gotoAndStop(15);` menjadi

```
gotoAndPlay(15);
```

Jalankan aplikasi dengan menekan **Ctrl+Enter**. Maka aplikasi akan memunculkan bola pada posisi frame 15 dan bola bergerak sampai frame 30, lalu kembali lagi ke frame 1. Selanjutnya, karena di frame 1 terdapat kode tersebut, maka timeline akan melompat langsung ke frame 15, sehingga animasi frame 1 sampai frame 14 terlewati.

Bab VII

Interaktifitas

Dalam perkembangan teknologi terdapat istilah media dan new media. New media mengacu pada materi yang tersedia secara online, diakses melalui perangkat digital dan pada umumnya memberikan pengalaman interaktif kepada penggunanya. Kebanyakan teknologi yang digambarkan sebagai "media baru" bersifat digital, sering memiliki karakteristik yang dimanipulasi, melalui jaringan, padat, kompresibel, dan interaktif. Sehingga menurut definisi tersebut, multimedia pembelajaran interaktif tergolong pada kategori new media.

Lister (2003) menjelaskan bahwa sebuah media baru harus memiliki konsep dasar yang menjadi karakteristik dari media baru, yaitu :

1. Digitality

Berarti seluruh data terpisah dari bentuk fisiknya (contoh : teks tidak lagi harus berbentuk buku), data dapat disimpan dalam ruang yang kecil, data dapat diakses dengan kecepatan tinggi, dan data dapat dimanipulasi dengan mudah.

2. Interaktivitas

Berarti bahwa pengguna dapat melakukan intervensi manipulasi dan intervensi terhadap media yang diakses. Interaktivitas merupakan salah satu kelebihan dari media baru karena memungkinkan : (1) pengguna menjadi memiliki kekuatan untuk terlibat dengan teks media, (2) pengguna mempunyai hubungan yang lebih independen dengan sumber-sumber, (3) penggunaan media yang bersifat personal, (4) pengguna memiliki lebih banyak pilihan.

3. Hypertext

Hypertext adalah istilah yang menggambarkan teks yang mampu menghubungkan dengan teks lain yang berada di atas atau di luar teks tersebut. Hypertext memungkinkan pengguna dapat membaca teks tanpa harus berurutan dan memulai teks dari mana saja yang ia inginkan.

4. Dispersal

Berarti media baru sangat spesifik dari segi produksi dan konsumsi. Beragamnya pesan dan sumber yang dimiliki oleh media baru menyebabkan audiens menjadi

lebih selektif, semakin tersegmentasi, dan adanya hubungan personal antara pengguna dan media.

5. Virtuality

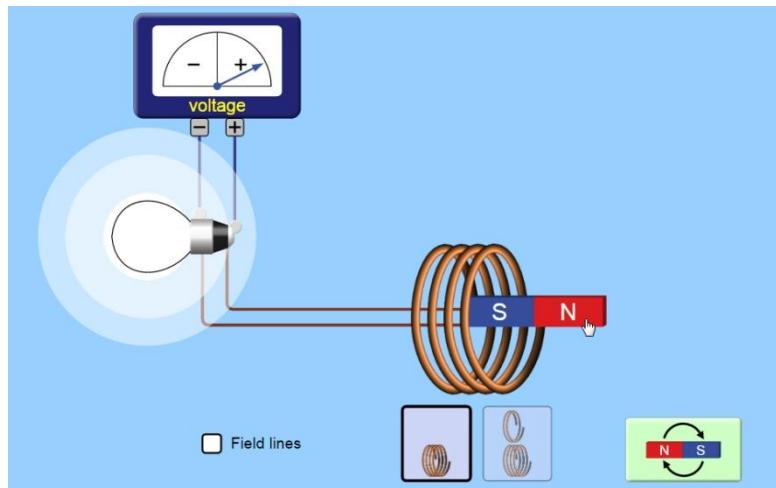
Berarti media baru dapat memberikan pengalaman kepada pengguna dalam sebuah lingkungan yang dikonstruksi media di mana pengguna dapat melakukan proses interaksi.

Dari beberapa karakteristik new media, karakter yang paling menonjol dalam multimedia pembelajaran interaktif adalah interaktivitas. William (dalam Jancowski & Hanssen, 1996) mendefinisikan interaktivitas sebagai derajat di mana partisipan dalam proses komunikasi memiliki kontrol, dan dapat bertukar peran dalam mutual discourse. Dengan menggunakan konsep mutual discourse, pertukaran, kontrol dan partisipan tersebut dapat dibedakan tiga level interaktivitas, yaitu percakapan tatap muka dengan derajat interaktivitas tertinggi; derajat yang berikutnya yaitu interaktivitas yang dimungkinkan antara orang dengan medium, atau orang dengan sistem di mana isi dapat dimanipulasi; ketiga adalah interaktivitas yang diperoleh dalam sistem informasi yang tak memungkinkan adanya intervensi dari pengguna untuk merubah konten.

Interaktifitas dalam multimedia interaktif adalah keleluasaan pengguna (operator/user) dalam mengontrol media dan kemampuan media dalam merespon input yang diberikan oleh pengguna. Interaktivitas dalam multimedia interaktif terbagi menjadi 2 yaitu interaktivitas mental dan interaktivitas fisik. Interaktivitas mental adalah interaktivitas di mana pengguna mencoba memahami materi dengan cara menangkap informasi-informasi yang ditampilkan, mengolah dan menyimpannya dalam otak. Sedangkan interaktivitas secara fisik dalam multimedia interaktif adalah keterlibatan kegiatan fisik dari pengguna untuk memberikan interaksi kepada media. Interaktivitas fisik bervariasi dari yang paling sederhana hingga yang kompleks. Interaktivitas sederhana misalnya menekan keyboard atau melakukan klik sebuah tombol dengan mouse atau menyentuh layar untuk berpindah halaman atau memasukkan jawaban dari suatu latihan yang diberikan oleh aplikasi. Interaktivitas yang kompleks misalnya aktivitas di dalam suatu simulasi sederhana di mana pengguna bisa mengubah-ubah suatu variabel tertentu atau di dalam simulasi kompleks atau interaksi di mana pengguna menggerakkan objek virtual.



Gambar 127. Interaktivitas sederhana



Gambar 128. Interaktivitas kompleks

Keunggulan multimedia di dalam interaktivitas adalah media ini secara inheren mampu memaksa pengguna untuk berinteraksi dengan materi baik secara fisik dan mental.

Sebagai contoh adalah simulasi kumparan Faraday di atas. Berbeda halnya jika materi yang sama disajikan dengan buku atau video, pada simulasi tersebut pengguna dapat menggerakkan magnet (interaktivitas fisik), sehingga mengetahui posisi dan gerakan terbaik dalam kumparan tembaga untuk menghasilkan daya listrik (interaktivitas mental).

Actionscript yang dimiliki oleh Flash mendukung seorang programer untuk memanipulasi objek baik dari segi posisi, ukuran, gerakan sampai dengan input. Hal tersebut dapat dimanfaatkan untuk membangun interaktivitas dalam multimedia interaktif, baik itu untuk membentuk interaktifitas yang sederhana sampai interaktifitas kompleks.

Movieclip properties

Untuk memahami proses pengembangan interaktifitas dalam media yang akan dibuat, perlu dipahami terlebih dahulu dasar dalam memanipulasi objek. Movieclip sebagai salah satu bentuk objek akan sering diaplikasikan dalam multimedia interaktif, sehingga seorang programer harus mampu memanipulasi Movieclip.

Dalam Flash setiap objek termasuk movieclip memiliki parameter-parameter yang disebut sebagai properties. Sebagai contoh movieclip memiliki posisi dalam bentuk kordinat x dan y (z jika memasuki mode 3D), warna, timeline, skala, rotasi dan sebagainya. Untuk memahami properties dari movieclip perhatikan contoh berikut :

1. Buatlah sebuah file baru. Selanjutnya dengan buatlah sebuah gambar burung sederhana dengan **Brush tool**.



Gambar 129. Brush tool

2. Seleksi gambar tersebut dan **Convert to Symbol** menjadi movieclip “animasi burung”.
3. Dobel klik movieclip “animasi burung” tersebut, kemudian pada mode edit movieclip buatlah **animasi frame by frame** (seperti pada contoh bab sebelumnya).



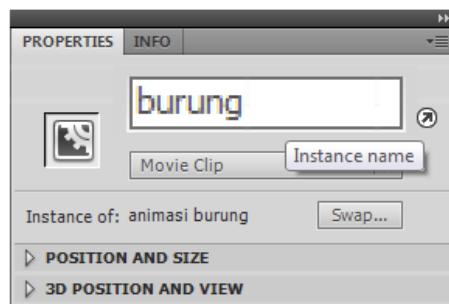
Gambar 130. Movieclip “animasi burung”

4. Selanjutnya keluar dari mode edit dengan menekan **Ctrl+E**.

Instance Name

Movieclip “animasi burung” tersebut telah berada di Stage, namun tidak akan terdeteksi oleh kode karena belum memiliki nama instansi yang spesifik (nama symbol tidak dapat di deteksi oleh kode secara langsung), sehingga perlu ditambahkan **Instance Name**.

5. Klik movieclip “animasi burung” tersebut kemudian buka panel properties dan ketikan “burung” pada kolom instance name.



Gambar 131. Instance name “burung”

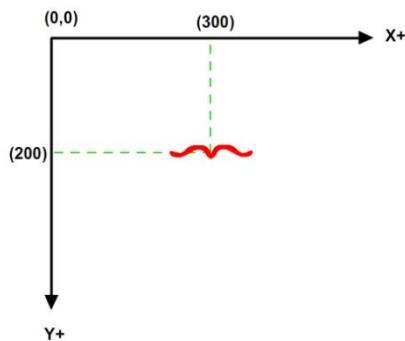
Dengan menambahkan instance name, movieclip “animasi burung” dapat diakses dengan kode. Namun demikian akses kode harus diinisiasi dengan nama instance yaitu “burung”. Untuk lebih jelasnya lanjutkan projek tersebut, sebagai berikut.

6. Selanjutnya **buatlah layer baru** (layer 2). **Klik frame 1 layer 2**, kemudian buka panel **Action** dan ketikan kode berikut

```
1. burung.x = 300;  
2. burung.y = 200;  
3. burung.stop();
```

7. Jalankan aplikasi dengan menekan **Ctrl+Enter**, maka akan didapati burung berpindah posisi ke kordinat x = 300 dan y = 200; dan burung dalam kondisi diam karena adanya kode `stop()`;

Sistem kordinat yang digunakan dalam Flash hampir sama dengan sistem kordinat kartesian, hanya saja untuk sumbu Y semakin ke bawah semakin bernilai positif.



Gambar 132. kordinat kartesian

8. Untuk menguji beberapa properties dari movieclip “animasi burung”, **klik** kembali **frame 1 layer 2** dan tambahkan kode menjadi :

```

1. burung.x = 300;
2. burung.y = 200;
3. burung.stop();
4. burung.rotation = 45;
5. burung.alpha = 0.4;
6. burung.scaleX = 2;
7. burung.scaleY = 2;

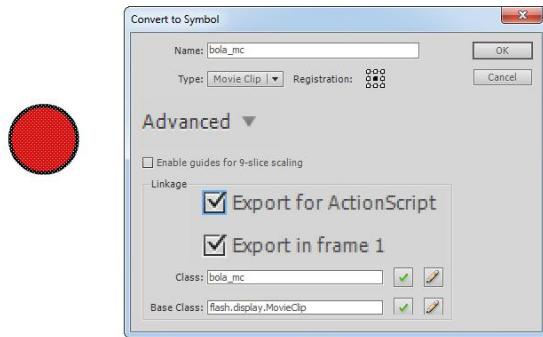
```

Selain properties di atas, masih ada beberapa properties dari movieclip yang bisa dimanfaatkan. Dalam tutorial-tutorial selanjutnya akan dibahas secara mendetail tentang mengatur properties dari movieclip.

Menambah objek dengan AddChild

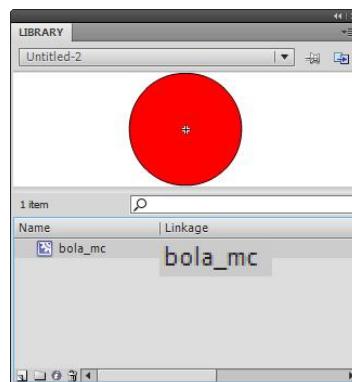
Salah satu metode yang paling sering digunakan oleh programer dalam membuat aplikasi adalah menambahkan objek ke layar dengan menggunakan kode. Berbeda dengan projek sebelumnya dimana movieclip “animasi burung” sudah berada di stage, dan harus diberikan instance name, pada projek berikutnya simbol movieclip ditambahkan ke stage dengan menggunakan kode. Untuk lebih jelasnya perhatikan contoh berikut :

1. Buatlah sebuah file baru, kemudian buatlah sebuah lingkaran dengan menggunakan **Oval Tools**.
2. Seleksi lingkaran tersebut kemudian **Convert to Symbol** menjadi movieclip “bola_mc”. Namun jangan di klik OK terlebih dahulu. Buka panel **advanced**, kemudian seleksi (centang) opsi **Export for Actionscript**. Pada kolom **Class** ketikan “bola_mc” (secara default akan memiliki nama yang sama dengan nama symbol). Lalu klik **OK**.



Gambar 133. Export for Actionscript

3. **Hapus movieclip tersebut** dari layar dengan cara menyeleksi movieclip dan menekan tombol **Delete**. Alasan dari menghapus movieclip dari layar (Stage) adalah agar layar bersih (tidak ada objek apapun), sehingga penambahan objek dengan kode tidak terganggu oleh objek lain.
4. Buka panel **library** dengan menekan **Ctrl+L** atau memilih menu **Window > Library**. Pada panel tersebut terdapat movieclip “bola_mc”, dan jika dilihat lebih detail maka akan didapati **Linkage : export : bola_mc**. Nama yang tertera di kolom linkage inilah yang dapat diakses dengan menggunakan kode (hampir sama dengan instance name).



Gambar 134. Linkage “bola_mc” pada library

5. **Klik frame 1 layer 1**, kemudian buka panel **Action** (F9). Kemudian ketikan kode berikut :

```
1. var bola : bola_mc = new bola_mc();
2. addChild(bola);
```

6. Jalankan aplikasi dengan menekan **Ctrl+Enter**. Maka akan didapati bola tersebut muncul tepat di pojok kiri atas layar. Hal ini dikarenakan secara default penambahan objek dengan menggunakan kode akan meletakkan objek pada posisi 0,0.

7. Edit kembali kode pada frame 1 layer 2 menjadi sebagai berikut :

```
1. var bola : bola_mc = new bola_mc();  
2. addChild(bola);  
3. bola.x = 200;  
4. bola.y = 100;
```

Jalankan aplikasi dengan menekan **Ctrl+Enter**. Maka bola akan bergeser ke kordinat 200, 100.

Penjelasan kode :

`var bola: bola_mc` merupakan deklarasi variabel bola. Variabel bola tersebut bertipe **object** dimana object yang digunakan adalah **bola_mc** (nama **linkage** dari movieclip bola_mc).

`= new bola_mc();` merupakan deklarasi untuk membuat objek baru yang mengacu pada movieclip dengan linkage bola_mc().

`addChild(bola);` setelah variabel bola didefinisikan sebagai objek baru, maka objek tersebut siap ditambahkan ke layar (Stage) dengan perintah `addChild`;

`bola.x = 200;` Setelah objek bola ditambahkan ke layar, objek tersebut dapat diatur propertiesnya sebagaimana mengatur properties movieclip.

EventListener

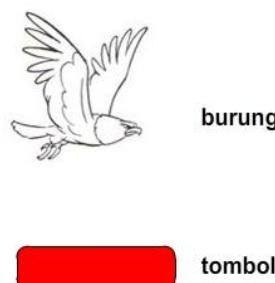
EventListener adalah sebuah urutan kejadian (event) yang dapat diperintahkan kepada sebuah objek. Sebagai contoh ketika sebuah tombol ditekan, maka akan menjalankan animasi sebuah movieclip. Berarti pada tombol tersebut perlu ditambahkan sebuah EventListener yang berkaitan dengan proses klik mouse yaitu MouseEvent.

Pada dasarnya pada Actionscript 3 terdapat beberapa event, namun untuk membentuk interaktivitas bagi seorang programer pemula cukup memahami beberapa event saja yaitu MouseEvent, KeyboardEvent dan EnterFrame Event.

MouseEvent

MouseEvent digunakan untuk mendeteksi adanya aktifitas mouse seperti klik kiri, klik kanan, mouse roll over, mouse roll out, dobel klik dan scroll mouse. Untuk mempermudah pemahaman tentang MouseEvent perhatikan contoh berikut :

1. Buatlah sebuah file baru. Selanjutnya **buatlah sebuah movieclip “animasi burung”**, dimana di dalamnya terdapat animasi burung mengepakan sayap secara **frame by frame**.
2. Klik movieclip “animasi burung” tersebut kemudian buka panel **properties** dan ketikkan “**burung**” pada kolom **instance name**.
3. Buatlah sebuah kotak dengan menggunakan **Rectangle tool**, kemudian **Convert to Symbol** menjadi **tombol** (button), dengan nama “tombol”.
4. Klik button “tombol” kemudian buka panel **properties** dan ketikkan “**tombol**” pada kolom **instance name**.



Gambar 135. Pengaturan instance name.

5. **Buatlah sebuah layer baru** (layer 2). **Klik frame 1 layer 2** buka panel **Action** (F9) dan ketikan kode berikut :

```
1. burung.stop();  
2. function jalankanAnimasi(e:MouseEvent):void{  
3.     burung.play();  
4. }  
5. tombol.addEventListener(MouseEvent.CLICK,  
    jalankanAnimasi);
```

6. Jalankan aplikasi dengan menekan **Ctrl+Enter**. Klik tombol tersebut dan animasi burung akan berjalan.

Penjelasan program :

`burung.stop();` merupakan perintah untuk menghentikan animasi movieclip dengan instance name “burung”.

`function jalankanAnimasi() { . . . }` merupakan deklarasi sebuah fungsi bernama jalankanAnimasi.

`e:MouseEvent` berarti fungsi tersebut akan dikaitkan dengan event yang berhubungan dengan aktivitas mouse.

`burung.play();` yang berada dalam blok fungsi jalankanAnimasi berarti animasi burung akan dijalankan apabila fungsi dipanggil.

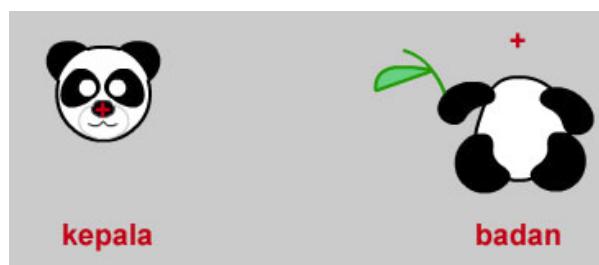
`tombol.addEventListener(MouseEvent.CLICK, jalankanAnimasi);` berarti tombol diberikan sebuah perintah, dimana apabila tombol diklik dengan menggunakan mouse, maka fungsi jalankanAnimasi akan dipanggil.

Aplikasi yang memiliki listener dengan event `MouseEvent.CLICK` ketika dipublish untuk aplikasi mobile, sama fungsinya dengan efek sentuh (touch).

Drag and Drop

Dalam beberapa kasus, interaktifitas dapat dilakukan dengan cara **drag and drop**. Sebagai contoh dalam simulasi hukum Faraday (gambar 13..), objek magnet dapat didrag oleh pengguna dan digerakkan ke area kumparan. Dalam beberapa aplikasi juga dapat kita temukan fitur drag and drop. Untuk membuat fitur tersebut, perhatikan contoh berikut :

1. Buatlah sebuah file baru. Selanjutnya buatlah 2 buah gambar yaitu kepala panda dan badan panda yang terpisah, kemudian **Convert to Symbol** masing-masing gambar menjadi movieclip kepala dan movieclip badan. Perhatikan letak titik pusat (titik registrasi) dari movieclip tersebut (+).



Gambar 136. Movieclip kepala dan badan panda.

2. Klik movieclip kepala, kemudian buka panel properties dan ketikan “kepala” pada instance name. Berikan instance name “badan” untuk movieclip badan
3. Buatlah sebuah layer baru (layer 2). Klik frame 1 layer 2 buka panel Action (F9) dan ketikan kode berikut :

```

1. kepala.addEventListener(MouseEvent.MOUSE_DOWN,
   penceMouse);
2. kepala.addEventListener(MouseEvent.MOUSE_UP,
   lepasMouse);
3.
4. function penceMouse(e:MouseEvent):void{
5.   kepala.startDrag();
6. }
7. function lepasMouse(e:MouseEvent):void{
8.   kepala.stopDrag();
9. }

```

10. Jalankan aplikasi dengan menekan **Ctrl+Enter**. Klik dan drag kepala panda tersebut untuk menguji coba kode.

Drag dan Snap

Pada contoh di atas untuk mendeteksi apakah movieclip kepala sudah berada tepat diposisi movieclip badan digunakan metode jarak, yaitu menggunakan teori pitagoras sederhana dengan menghitung akar (`Math.sqrt`) kuadrat dari selisih kordinat 2 objek. Sehingga ketika movieclip kepala berjarak (ber radius) kurang dari 50 pixel dan tombol mouse dilepas, maka kordinat movieclip kepala dijadikan sama dengan kordinat movieclip badan sehingga akan muncul efek **Snap**.

Sebenarnya kode AS3 memiliki sistem deteksi antara 2 objek yang di drag mouse, yaitu **dropTarget**. Kodenya kurang lebih sebagai berikut :

```

1. function lepasMouse(e:MouseEvent):void{
2.   kepala.stopDrag();
3.   //snap ke badan
4.   if (kepala.dropTarget != null){
5.     kepala.x = badan.x;
6.     kepala.y = badan.y;
7.   }
8. }

```

Akan tetapi kode tersebut kurang optimal, karena deteksi posisi masih terlalu luas, sehingga ketika kepala baru menyentuh titik terluar badan sudah dianggap sebagai **dropTarget**, sehingga lebih baik menggunakan metode pengukuran jarak agar akurasi lebih tepat. Untuk lebih jelasnya, perhatikan langkah berikut :

1. Lanjutkan file pada tutorial sebelumnya (drag movieclip).
2. **Klik frame 1 layer 2** buka panel **Action (F9)** dan ubah kodennya menjadi sebagai berikut :

```
1. var awal_x:Number = kepala.x;
2. var awal_y:Number = kepala.y;
3.
4. kepala.addEventListener(MouseEvent.MOUSE_DOWN,
    pincetMouse);
5. kepala.addEventListener(MouseEvent.MOUSE_UP,
    lepasMouse);
6.
7. function pincetMouse(e:MouseEvent):void{
8.     kepala.startDrag();
9. }
10. function lepasMouse(e:MouseEvent):void{
11.     kepala.stopDrag();
12.     //snap ke badan
13.     var jarak = Math.sqrt((kepala.x -
    badan.x)*(kepala.x - badan.x)+(kepala.y -
    badan.y)*(kepala.y - badan.y));
14.     if (jarak < 50){
15.         kepala.x = badan.x;
16.         kepala.y = badan.y;
17.     }else{
18.         kepala.x = awal_x;
19.         kepala.y = awal_y;
20.     }
21. }
```

3. Jalankan aplikasi dengan menekan **Ctrl+Enter**. Klik dan drag kepala panda tersebut mendekati posisi yang tepat (sekitar leher panda) untuk menguji coba kode.

Penjelasan kode :

```
var awal_x:Number = kepala.x; Pada awalnya kordinat x dan y dari movieclip
kepala disimpan di dalam variabel awal_x dan awal_y. Hal ini dimaksudkan agar apabila
kepala tidak didrag pada posisi yang tepat, maka akan kembali ke posisi awal
```

```
kepala.addEventListener(MouseEvent.MOUSE_DOWN, pincetMouse);
```

Selanjutnya movieclip kepala diberikan listener ketika mouse ditekan. Ketika event terjadi (yaitu ketika Mouse ditekan), maka fungsi pincetMouse akan dijalankan.

```
kepala.startDrag(); merupakan kode untuk memulai mode drag sebuah objek.
```

```
kepala.stopDrag(); merupakan kode untuk mengakhiri mode drag sebuah objek.
```

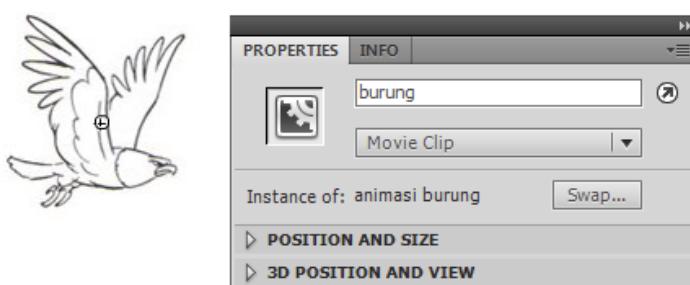
```
var jarak = Math.sqrt(...); merupakan perhitungan jarak menggunakan metode pitagoras yaitu akar dari selisih kuadrat jarak antara movieclip kepala dengan movieclip badan.
```

```
if (jarak < 50){...}else {...} merupakan kondisi untuk melakukan proses snap. Apabila jarak kurang dari 50 pixel maka posisi akan di snap, sedangkan jika lebih akan dikembalikan ke posisi awal_x dan awal_y.
```

EnterFrame Event

Enterframe Event merupakan kejadian yang dilakukan secara berulang-ulang sejak inisialisasi listener sampai dengan dihentikannya listener atau aplikasi ditutup. Sebagai contoh event ini digunakan untuk mendapatkan gerakan objek secara terus menerus. Perhatikan contoh berikut :

1. Buatlah sebuah file baru. Selanjutnya **buatlah sebuah movieclip “animasi burung”**, dimana di dalamnya terdapat animasi burung mengepakkan sayap secara **frame by frame**.
2. Klik movieclip “animasi burung” tersebut kemudian buka panel **properties** dan ketikkan **“burung”** pada kolom **instance name**.



Gambar 137. Movieclip dengan instance name “burung”

3. Buatlah sebuah layer baru (layer 2). Klik frame 1 layer 2 buka panel Action (F9) dan ketikan kode berikut :

```
1. function terbang(e:Event):void{
2.     var ob:Object = e.currentTarget;
3.     ob.x += 5;
4. }
5.
6. burung.addEventListener(Event.ENTER_FRAME, terbang);
```

4. Jalankan aplikasi dengan menekan **Ctrl+Enter**. Maka burung tersebut akan bergerak ke kanan secara terus menerus sampai keluar dari layar.
5. Selanjutnya kita dapat menambahkan sedikit logika agar burung tersebut dapat bergerak secara berulang ke kanan dan ke kiri. Ubah kode pada frame 1 layer 2 menjadi sebagai berikut :

```
1. var kecepatan:int = 5;
2. function terbang(e:Event):void{
3.     var ob:Object = e.currentTarget;
4.     ob.x += kecepatan;
5.     if (ob.x > 600 || ob.x < 100) {
6.         kecepatan *= -1;
7.         ob.scaleX = kecepatan/Math.abs(kecepatan);
8.     }
9. }
10.
11. burung.addEventListener(Event.ENTER_FRAME, terbang);
```

6. Jalankan kembali aplikasi dengan menekan **Ctrl+Enter**, dan burung akan terbang secara berulang ke kanan dan ke kiri.

Penjelasan program :

`var kecepatan:int = 5;` merupakan deklarasi kecepatan burung. Nilai dari variabel ini bisa diubah secara dinamis ketika kondisi tertentu (baris 5) terpenuhi.

`function terbang(e:Event):void{...}` merupakan fungsi yang dijalankan secara berulang-ulang setelah kode `addEventListener` diaktifkan.

`var ob:Object = e.currentTarget;` merupakan deklarasi variabel yang mengacu pada objek yang diperintah oleh `EventListener`, dimana dalam hal ini adalah movieclip dengan instance name “burung”.

`ob.x += kecepatan;` berarti setiap frame (karena menggunakan 30 fps, maka setiap 1/30 detik) posisi x dari burung akan ditambah dengan variabel kecepatan.

`if (ob.x > 600 || ob.x < 100) { .. }` adalah kondisi dimana ketika posisi burung terlalu ke kanan atau (`||`) terlalu ke kiri, maka variabel kecepatan akan dikalikan -1;

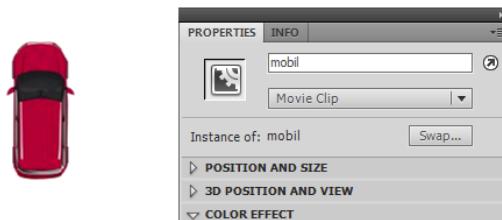
`kecepatan *= -1;` berarti ketika kondisi di atas terpenuhi kecepatan akan dibalik. Jika semula adalah 5 maka akan menjadi -5, dan ketika kecepatan -5 maka burung akan bergerak ke kiri sampai kordinat `x < 100` dan kondisi if terpenuhi lagi sehingga kecepatan menjadi $-5 \times -1 = 5$; dan burung bergerak ke kanan kembali.

`ob.scaleX = kecepatan/Math.abs(kecepatan);` untuk merubah arah burung dengan menggunakan skala X. Ketika kecepatan bernilai -5, maka `ob.scaleX` akan bernilai -5 dibagi nilai absolut kecepatan (5), sehingga hasilnya adalah -1 dan burung akan berubah arah (flip horizontal)

Keyboard Event

Sebagaimana MouseEvent, Keyboard Event adalah sebuah event yang terjadi ketika ada masukan dari keyboard. Contoh penggunaan Keyboard Event adalah pada saat menggerakkan sebuah objek dengan menggunakan tombol panah, misalnya ketika membuat simulasi gerakan mobil. Untuk lebih jelasnya perhatikan contoh berikut :

1. Buatlah sebuah file baru. Selanjutnya buatlah sebuah gambar mobil tampak atas dan menghadap ke atas, atau anda dapat mengimport gambar siap pakai dengan memilih menu **File > Import > Import to Stage**.
2. Seleksi gambar mobil tersebut, kemudian **Convert to Symbol** menjadi movieclip "mobil". Agar dapat diakses oleh kode, tambahkan **instance name "mobil"**.



Gambar 138. Movieclip mobil.

3. Buatlah sebuah layer baru (layer 2). Klik frame 1 layer 2 buka panel Action (F9) dan ketikan kode berikut :

```
1. var tombol_atas:Boolean=false;
2. var tombol_bawah:Boolean=false;
3. var tombol_kiri:Boolean=false;
4. var tombol_kanan:Boolean=false;
5. mobil.kecepatan = 0;
6.
7.
8. function keyDownFunction(event:KeyboardEvent):void {
9.     if (event.keyCode == 38) {
10.         tombol_atas=true;
11.     } else if (event.keyCode == 39) {
12.         tombol_kanan=true;
13.     } else if (event.keyCode == 40) {
14.         tombol_bawah=true;
15.     } else if (event.keyCode == 37) {
16.         tombol_kiri=true;
17.     }
18. }
19. function keyUpFunction(event:KeyboardEvent) {
20.     if (event.keyCode == 38) {
21.         tombol_atas=false;
22.     } else if (event.keyCode == 39) {
23.         tombol_kanan=false;
24.     } else if (event.keyCode == 40) {
25.         tombol_bawah=false;
26.     } else if (event.keyCode == 37) {
27.         tombol_kiri=false;
28.     }
29. }
30.
31. function gerakMobil(e:Event):void{
32.     var ob:Object = e.currentTarget;
33.     ob.x += ob.kecepatan * Math.sin(ob.rotation * Math.PI
34. / 180);
35.     ob.y -= ob.kecepatan * Math.cos(ob.rotation * Math.PI
36. / 180);
37.     //mengatur berdasarkan input tombol
38.     if (tombol_atas) {
39.         if (ob.kecepatan < 10){
40.             ob.kecepatan += 0.2;
41.         }
42.     }else if (tombol_bawah) {
43.         ob.kecepatan *= 0.8;
44.     }else{
45.         ob.kecepatan *= 0.95;
46.     }
47.     if (tombol_kanan){
48.         ob.rotation+=5;
```

```
49. }
50.
51. }
52.
53. stage.addEventListener(KeyboardEvent.KEY_DOWN,
    keyDownFunction);
54. stage.addEventListener(KeyboardEvent.KEY_UP,
    keyUpFunction);
55. mobil.addEventListener(Event.ENTER_FRAME, gerakMobil);
```

4. Jalankan kembali aplikasi dengan menekan **Ctrl+Enter**, dan tekan tombol panah pada keyboard untuk mengatur gerakan mobil.

Penjelasan program :

`var tombol_atas:Boolean=false;` merupakan deklarasi variabel untuk mendeteksi apakah tombol atas ditekan (true) atau dilepas (false).

`if (event.keyCode == 38) { tombol_atas=true; }` Untuk mendeteksi tombol yang ditekan, setelah listener KeyboardEvent aktif maka kode masing-masing tombol keyboard bisa dideteksi. Sebagai contoh tombol panah atas memiliki kode 38.

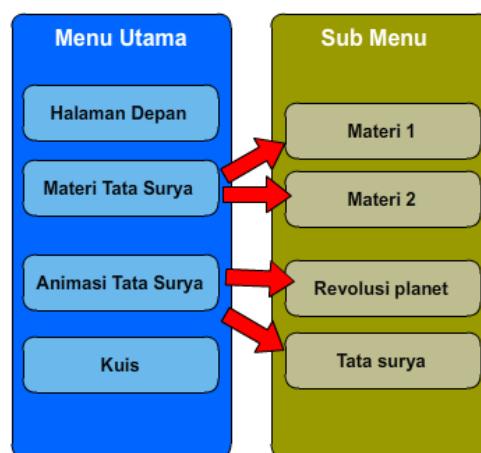
`ob.x += ob.kecepatan * Math.sin(ob.rotation * Math.PI / 180);` merupakan perhitungan trigonometri sederhana sebuah objek berdasarkan rotasi yang dimilikinya.

`stage.addEventListener(KeyboardEvent.KEY_DOWN, keyDownFunction);` merupakan deklarasi KeyboardEvent untuk mendeteksi tombol ditekan.

Bab VIII

Navigasi Halaman Interaktif Multimedia

Setelah memahami beberapa dasar desain multimedia interaktif, pemrograman dasar Actionscript dan dasar interaktivitas, langkah selanjutnya adalah merancang aplikasi multimedia interaktif secara keseluruhan. Sebagai contoh akan dibuat sebuah media pembelajaran interaktif tentang tata surya, maka perlu direncanakan pembagian halaman dan fitur-fitur yang akan ditampilkan. Berikut contoh perencanaan halaman interaktif multimedia pembelajaran tata surya untuk anak usia Sekolah Dasar:



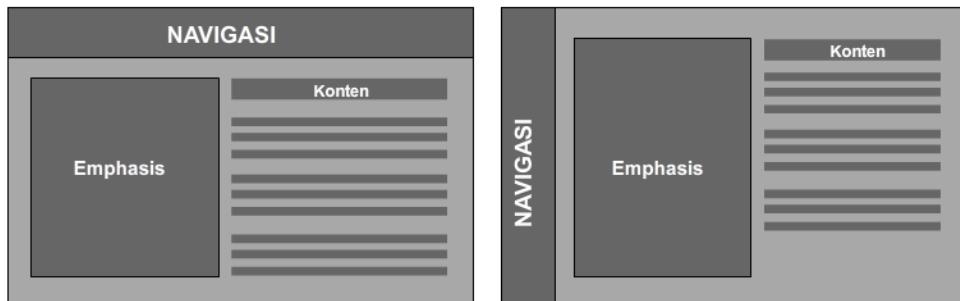
Gambar 139. Rencana halaman interaktif multimedia

Desain Halaman

Setelah perencanaan dibuat, tahapan selanjutnya adalah membuat desain halaman. Pada bab sebelumnya telah dijelaskan prinsip-prinsip desain agar multimedia yang dibuat nantinya mampu menarik pengguna dari sisi tampilan visual.

Desain halaman dalam multimedia juga dapat disebut sebagai tampilan antar muka (user interface). Interface memegang peranan penting karena melalui interface pengguna dapat berinteraksi dengan media. Untuk itu perencanaan interface harus benar-benar sesuai dengan kebutuhan, dalam hal ini dikenal istilah layout. Layout adalah penyusunan dari elemen-elemen desain yang berhubungan ke dalam bidang sehingga membentuk susunan artistik. Hal ini bertujuan untuk menampilkan gambar dan teks yang komunikatif sehingga memudahkan pembaca menerima informasi yang disajikan. Layout yang umum

digunakan untuk desain interface adalah dengan metode grid, yaitu membagi area menjadi beberapa bagian. Secara umum navigasi dapat diletakkan di bagian atas atau samping kiri, dan konten diletakkan di tengah area.



Gambar 140. Layout interface dengan sistem grid

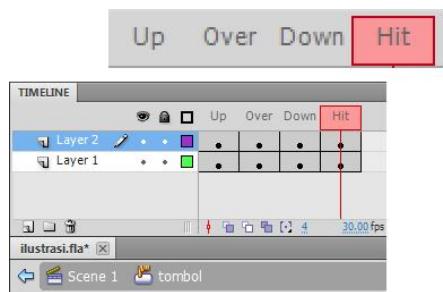
Sebagai contoh akan digunakan layout dengan navigasi di atas. Selanjutnya konten yang dibuat harus disesuaikan dengan target audiense (pengguna). Sebagai contoh, karena media yang akan dibuat adalah pembelajaran tentang tata surya untuk anak usia dasar, maka tampilan yang dibuat bersifat anak-anak dan mengandung elemen yang terkait dengan tata surya, yang dalam hal ini adalah astronot.



Gambar 141. Contoh layout halaman depan dengan navigasi di atas.

Membuat tombol

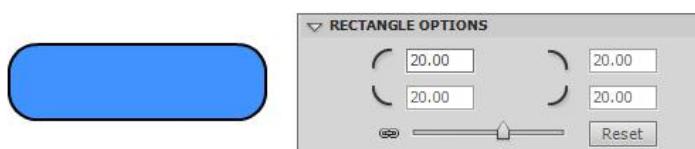
Tombol merupakan salah satu elemen penting dalam multimedia interaktif, untuk itu Flash secara khusus memiliki symbol bertipe **button**. Berbeda dengan symbol bertipe movieclip yang memiliki timeline seperti timeline pada stage, tombol hanya memiliki 4 buah frame yaitu frame Up, Over, Down dan Hit.



Gambar 142. Timeline symbol bertipe button.

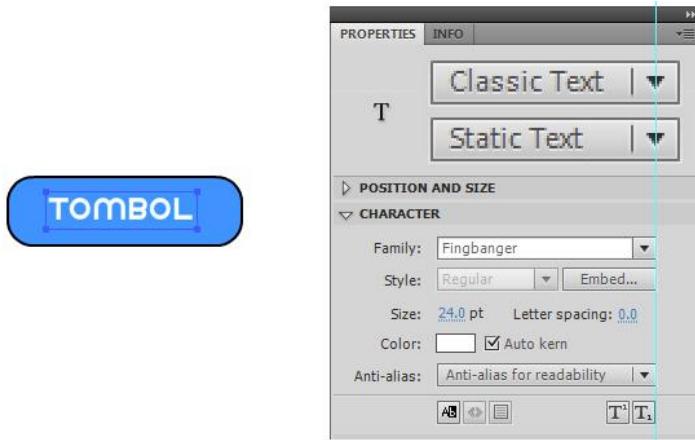
Up adalah kondisi ketika tombol normal, tidak mengalami aktifitas apapun. Over adalah kondisi ketika mouse melintas di atas area tombol. Down adalah kondisi ketika mouse menekan tombol, dan Hit adalah area yang menjadi deteksi atau sensor dari tombol. Untuk lebih jelasnya perhatikan langkah-langkah berikut :

1. Buatlah sebuah file baru. Selanjutnya dengan menggunakan **Rectangle tool** buatlah sebuah kotak dengan sudut lengkung (Rounded rectangle). Untuk melakukannya atur **properties Rectangle option**.



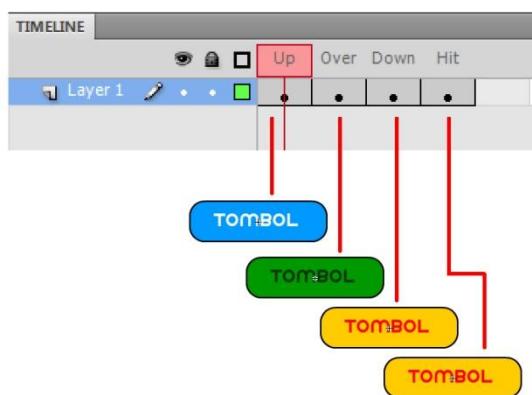
Gambar 143. Rounded rectangle

2. Selanjutnya klik teks tool, pada panel properties pastikan text engine adalah Classic Text dan text type adalah Static Text. Kemudian klik di atas kotak dan ketikkan kata “tombol”.



Gambar 144. Menambahkan teks pada tombol

3. Seleksi kotak dan teks tombol tersebut, kemudian **Convert to Symbol** menjadi **button** dengan nama “tombol”.
4. **Dobel klik button “tombol”** untuk memasuki **mode edit**. Kemudian **klik kanan frame Over** dan pilih **Insert Keyframe**. Selanjutnya ubah warna kotak dan warna teks (bedakan dengan warna pada frame Up).
5. **Klik kanan frame Down** dan pilih **Insert Keyframe**. Selanjutnya ubah kembali warna kotak dan warna teks.
6. **Klik kanan frame Hit** dan pilih **Insert Keyframe**. Untuk frame Hit tidak perlu dilakukan perubahan apapun, karena hanya digunakan sebagai sensor. Bahkan pada contoh ini tanpa frame Hit pun area akan dihitung berdasarkan frame sebelumnya.



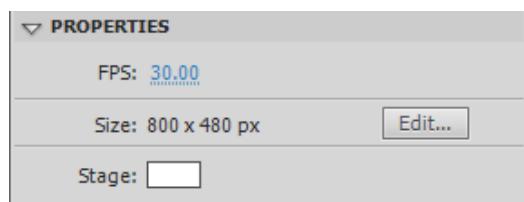
Gambar 145. Pengaturan keyframe button “tombol”

7. Jalankan aplikasi dengan menekan **Ctrl+Enter**. Coba gerakkan mouse ke atas tombol dan klik tombol tersebut, maka akan terjadi perubahan warna sesuai dengan pengaturan frame yang telah dilakukan.

Mengatur Halaman

Proses berikutnya yang sangat penting dan cukup rumit dalam proses pengembangan multimedia pembelajaran interaktif adalah mengatur halaman. Halaman di atur di dalam timeline sedemikian rupa sehingga dapat diakses dengan mudah oleh kode dan memiliki struktur yang memudahkan kita nantinya untuk menambah atau mengubah materi. Ada beberapa teknik yang dapat digunakan, mulai dari teknik menyusun konten secara manual di timele sampai dengan teknik pemrograman seutuhnya (menyusun semua layout dan halaman dengan kode). Pada buku ini akan dijelaskan teknik yang paling mudah yaitu menyusun konten atau materi secara manual. Untuk lebih jelasnya perhatikan contoh berikut :

1. Buatlah sebuah file baru. Selanjutnya atur file dengan membuka **properties**, sehingga file memiliki ukuran standar untuk aplikasi mobile yaitu **800 x 480 pixel** dan **30 FPS**.



Gambar 146. Properties dokumen

2. Buatlah sebuah tampilan untuk halaman pertama dari multimedia dengan menggunakan drawing tool, atau dengan cara **File > Import > Import to Stage**.



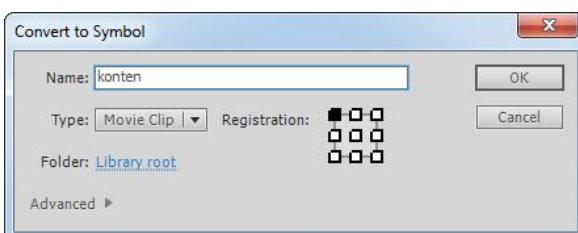
Gambar 147. Contoh halaman awal multimedia

3. Seleksi gambar tersebut, kemudian buka panel **properties**, pastikan posisi gambar berada di kordinat **x = 0; y = 0;** dan ukuran gambar tepat **800x480 pixel**. Jika belum tepat, nilai tersebut dapat diedit secara langsung melalui panel properties.



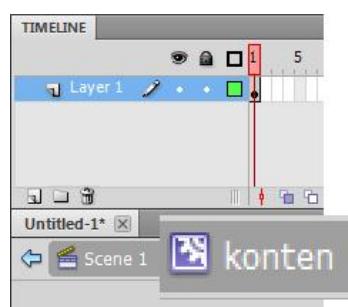
Gambar 148. Position and size properties

4. Seleksi gambar tersebut **dan Convert to Symbol** menjadi movieclip “konten”, sebelum mengklik tombol OK pastikan **titik registrasi** movieclip berada di **kiri atas**.



Gambar 149. Titik registrasi movieclip konten

5. **Dobel klik movieclip konten** untuk masuk ke mode edit.



Gambar 150. Mode edit movieclip konten

6. **Rename** nama **layer 1** menjadi layer **background** (dengan cara dobel klik pada nama layer) , kemudian **buatlah buatlah 5 buah layer baru** dan ubah nama masing-masing layer sesuai dengan urutannya mulai dari atas yaitu layer : **action, label, tombol, sub_halaman** dan **halaman isi**.



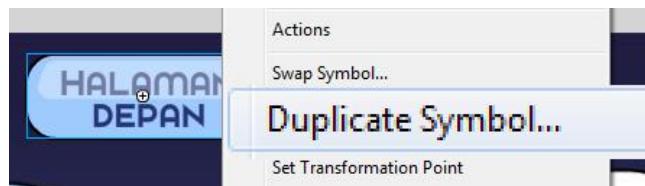
Gambar 151. Susunan layer movieclip konten

7. Klik frame 1 layer tombol kemudian buatlah symbol **button** “tombol_home” sesuai dengan layout yang telah didesain sebelumnya. Letakkan tombol tersebut pada posisi yang sesuai.



Gambar 152. Tombol halaman depan dan posisinya terhadap background

8. Dengan cara yang sama, **buatlah 3 tombol** selanjutnya yaitu tombol “materi”, tombol “animasi” dan “tombol kuis”. Untuk mempertahankan konsistensi dan mempermudah proses pembuatan, dapat digunakan fitur **Duplicate Symbol**. Klik kanan “tombol_home” lalu **Copy** dan **Paste** kan di sebelah kanannya. Selanjutnya klik kanan “tombol_home” hasil copy-paste, lalu pilih **Duplicate Symbol**. Ketikan nama baru, yaitu “tombol materi” dan klik OK. Kemudian dobel klik untuk mengeditnya.

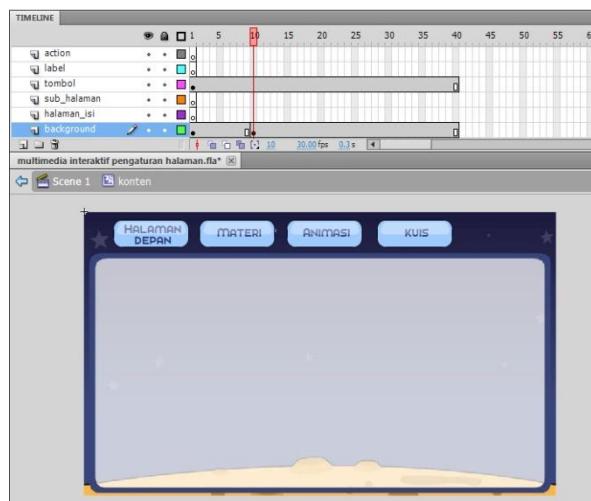


Gambar 153. Duplicate Symbol



Gambar 154. Pengaturan posisi tombol

9. Klik kanan frame 40 layer tombol kemudian pilih **Insert Frame**, sehingga tombol akan tampil sampai dengan frame 40 tanpa perubahan apapun. Hal ini dikarenakan tombol yang ada akan selalu muncul di seluruh halaman.
10. Kemudian klik kanan frame 10 layer background dan pilih **Insert Blank Keyframe**. Selanjutnya buatlah sebuah gambar background atau import gambar siap pakai ke stage. Pastikan background yang digunakan tidak mengganggu tampilan materi yang akan ditambahkan. Selanjutnya klik kanan frame 40 layer background dan pilih **Insert frame** agar background tersebut muncul sampai dengan frame 40.



Gambar 155. Background di frame 10.

11. Klik kanan frame 10 layer halaman isi kemudian pilih **Insert Blank Keyframe**. Pada frame ini ditujukan untuk meletakkan materi. Pada frame ini dapat ditambahkan text, animasi movieclip, maupun video sesuai dengan konsep yang telah dibuat sebelumnya. Untuk sementara, tambahkan sebuah static text "**materi tata surya**" pada stage. Pada bab selanjutnya teks tersebut akan diganti dengan materi yang lebih kompleks.



Gambar 156. Konten sementara untuk materi tata surya

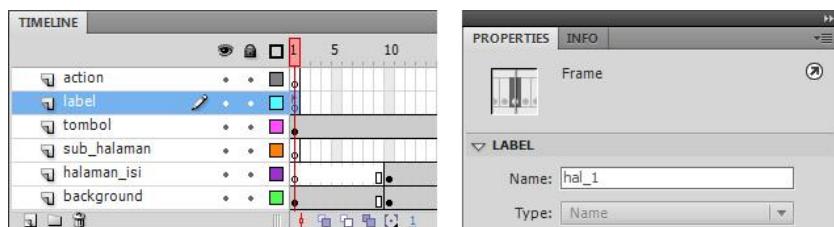
12. Klik kanan frame 20 layer halaman isi kemudian pilih Insert Blank Keyframe. Kemudian tambahkan sebuah static text "animasi tata surya" pada stage.
 13. Klik kanan frame 30 layer halaman isi kemudian pilih Insert Blank Keyframe. Kemudian tambahkan sebuah static text "kuis tata surya" pada stage.
- Penggunaan kelipatan 10 setiap Keyframe dimaksudkan untuk mempermudah anda untuk menambahkan efek transisi dan memudahkan proses editing di kemudian hari.
14. Simpan file.

Menambahkan label

Label dalam flash adalah sebuah tanda dalam bentuk teks (String) yang digunakan untuk mengidentifikasi frame. Sebagai contoh apabila terdapat animasi berdurasi 200 detik, kemudian pada frame 100 ditambahkan label “tengah”, maka untuk melompat ke frame tersebut cukup dengan menggunakan perintah gotoAndPlay(“tengah”);

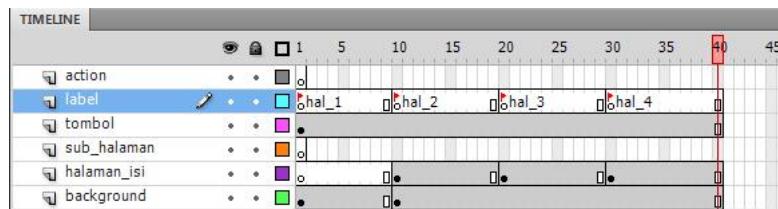
Label berfungsi untuk mempermudah dalam mengidentifikasi frame, sehingga ketika struktur keyframe suatu file berubah akibat penambahan atau pengurangan frame, kode tidak perlu diubah. Untuk lebih jelasnya, lanjutkan file di atas dengan langkah sebagai berikut :

15. Untuk menambahkan **label** klik frame 1 layer label kemudian buka panel **properties**. Ketikan **hal_1** pada kolom Frame label.



Gambar 157. Frame label

16. Klik kanan frame 10 layer label kemudian Insert Blank Keyframe. Lalu buka **properties** dan tambahkan frame label **hal_2**. Lakukan hal yang sama pada frame 20 dan 30 untuk halaman 3 dan halaman 4.



Gambar 158. Label secara keseluruhan

17. Simpan file.

Penambahan Instance Name

Seperti pada penjelasan bab sebelumnya, penambahan instance name diperlukan agar beberapa object bisa diakses oleh actionscript. Dalam multimedia interaktif yang dikembangkan pada bab ini, objek utama yang harus memiliki instance name adalah tombol-tombol dan movieclip konten.

18. Klik **tombol “halaman_depan”**, kemudian buka panel properties dan ketikkan **“tombol_home”** pada kolom **instance name**.



Gambar 159. Instance name tombol_home

19. Dengan cara yang sama tambahkan **instance name “tombol_materi”**, **“tombol_animasi”** dan **“tombol_kuis”**.

20. Keluar dari mode edit movieclip “konten” dengan menekan **Ctrl+E**. Selanjutnya **klik movieclip konten** dan tambahkan **instance name “konten”**.

21. Simpan file.

Penambahan Actionscript untuk Navigasi

Setelah masing-masing elemen tersusun dengan baik dan memiliki instance name, tahapan berikutnya adalah menambahkan kode action script. Jika muncul pertanyaan kenapa seluruh halaman diletakkan di dalam movieclip “konten”, hal tersebut dikarenakan pengaturan objek dalam sebuah movieclip akan lebih mudah dilakukan, dan kode akan dijalankan satu kali saja sehingga tidak membutuhkan memori yang banyak.

Dalam pemrograman terdapat istilah garbage collector, dimana listener-listener yang sudah tidak digunakan harus dihapus. Apabila seluruh konten ditampilkan ke dalam timeline utama (Scene), maka akan muncul peluang kode yang ada di frame 1 (halaman depan) berulang beberapa kali, sehingga akan menambah kebutuhan memori dan memperlambat kerja aplikasi. Dengan memasukkan seluruh konten ke dalam movieclip kode akan lebih mudah diatur, diaktifkan dan dihapus.

Untuk menambahkan kode navigasi, perhatikan langkah-langkah berikut :

22. Lanjutkan file tutorial sebelumnya. Kemudian **buatlah sebuah layer baru** (layer 2).
23. **Klik frame 1 layer 2**, kemudian buka panel **Action** (F9) dan ketikan kode berikut :

```
1. konten.stop();
2. konten.tombol_home.addEventListener(MouseEvent.CLICK,
   aturNavigasi);
3. konten.tombol_materi.addEventListener(MouseEvent.
   CLICK, aturNavigasi);
4. konten.tombol_animasi.addEventListener(MouseEvent.
   CLICK, aturNavigasi);
5. konten.tombol_kuis.addEventListener(MouseEvent.CLICK,
   aturNavigasi);
6.
7. function aturNavigasi(e:MouseEvent):void{
8.     var namaTombol:String = e.currentTarget.name;
9.     if (namaTombol == "tombol_home"){
10.         konten.gotoAndStop("hal_1");
11.     }
12.     if (namaTombol == "tombol_materi"){
13.         konten.gotoAndStop("hal_2");
14.     }
15.     if (namaTombol == "tombol_animasi"){
16.         konten.gotoAndStop("hal_3");
17.     }
18.     if (namaTombol == "tombol_kuis"){
19.         konten.gotoAndStop("hal_4");
20.     }
21. }
```

24. Jalankan aplikasi dengan menekan **Ctrl+Enter**. Uji coba tombol dan navigasinya, pastikan semua bekerja dengan baik. Apabila tidak ada langkah

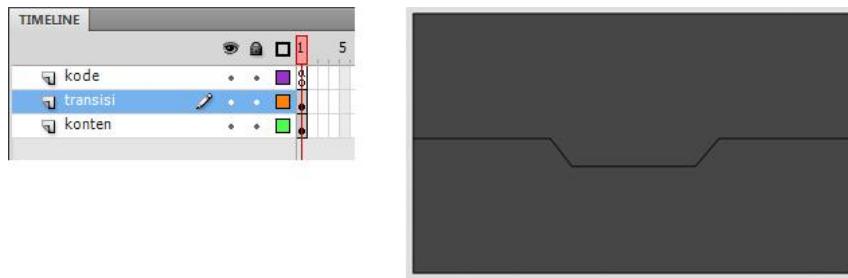
yang salah, maka sistem navigasi dari media pembelajaran interaktif akan berfungsi dengan baik.

Membuat Transisi Halaman

Untuk memperoleh sebuah interktif multimedia yang dinamis, pergantian antar halaman dapat ditambahkan sebuah efek transisi. Efek tersebut bisa berupa animasi motion tween, efek fade in/out, efek zoom in/out dan sebagainya. Pada contoh berikut dijelaskan proses pembuatan transisi halaman dengan menggunakan movieclip. Untuk lebih jelasnya perhatikan langkah berikut :

25. Lanjutkan file di atas. **Buatlah sebuah layer baru di atas layer 1** (layer 3). Kemudian atur kembali nama layer menjadi layer “**konten**”, “**transisi**” dan “**kode**” (berurutan dari bawah ke atas).
26. Klik **frame 1 layer transisi** kemudian buatlah sebuah kotak dengan ukuran yang sama persis dengan ukuran stage, dan posisikan tepat menutupi konten. Selanjutnya buatlah garis horizontal yang membelah kotak tersebut tepat di tengah (digunakan untuk membuat animasi transisi yang menutup bagian atas dan bagian bawah).

Lock layer konten untuk mempermudah proses agar konten tidak terganggu.



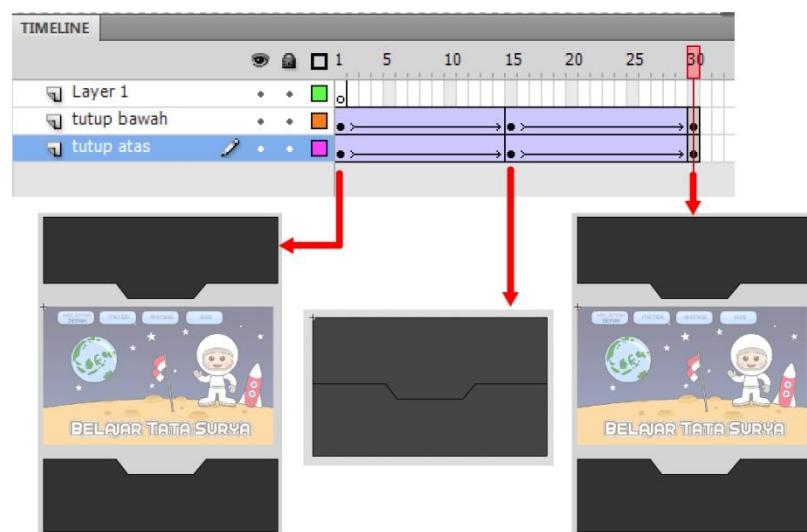
Gambar 160. Mengatur transisi

27. Seleksi gambar di layer transisi tersebut, kemudian **Convert to Symbol** menjadi **movieclip “transisi”**. Selanjutnya dobel klik untuk mengeditnya.
28. Pada mode edit, seleksi bagian atas dari kotak yang terbelah, kemudian **Convert to Symbol** menjadi **movieclip “tutup atas”**.
29. seleksi bagian bawah dari kotak yang terbelah, kemudian **Convert to Symbol** menjadi **movieclip “tutup bawah”**. Hapus outline yang tersisa dari stage, sehingga hanya menyisakan 2 movieclip.



Gambar 161. Movieclip tutup atas dan tutup bawah.

30. Seleksi dua movieclip tersebut, kemudian klik kanan dan pilih **Distribute to Layer**.
31. Selanjutnya buatlah sebuah **animasi tween** membuka dan menutup dengan durasi 30 frame. Pada keyframe 1 dan keyframe 30, posisi penutup sama yaitu membuka penuh. Sedangkan pada keyframe 15 posisi penutup, menutupi layar secara penuh. Perhatikan gambar.



Gambar 162. Animasi tween pada movieclip transisi

32. Keluar dari mode edit movieclip “transisi” dengan menekan **Ctrl+E**.
33. Untuk mengaktifkan transisi, **klik frame 1 layer action** dan ubah kode yang ada menjadi sebagai berikut :

```

1. konten.stop();
2. transisi.stop();
3.
4. konten.tombol_home.addEventListener(MouseEvent.CLICK,
   aturNavigasi);
5. konten.tombol_materi.addEventListener(MouseEvent.CLICK,
   aturNavigasi);

```

```

6. konten.tombol_animasi.addEventListener(MouseEvent.CLICK,
    aturNavigasi);
7. konten.tombol_kuis.addEventListener(MouseEvent.CLICK,
    aturNavigasi);
8.
9. function jalankanTransisi(e:Event):void{
10. if (transisi.currentFrame == 15) {
11.     konten.gotoAndStop(transisi.namaHalaman);
12. }
13. if (transisi.currentFrame == 30) {
14.     transisi.stop();
15.     transisi.removeEventListener(Event.ENTER_FRAME,
        jalankanTransisi);
16. }
17. }
18.
19. function transisiHalaman(namaHalaman:String):void{
20.     transisi.play();
21.     transisi.namaHalaman = namaHalaman;
22.     transisi.addEventListener(Event.ENTER_FRAME,
        jalankanTransisi);
23. }
24.
25. function aturNavigasi(e:MouseEvent):void{
26.     var namaTombol:String = e.currentTarget.name;
27.     if (namaTombol == "tombol_home") {
28.         transisiHalaman("hal_1");
29.     }
30.     if (namaTombol == "tombol_materi") {
31.         transisiHalaman("hal_2");
32.     }
33.     if (namaTombol == "tombol_animasi") {
34.         transisiHalaman("hal_3");
35.     }
36.     if (namaTombol == "tombol_kuis") {
37.         transisiHalaman("hal_4");
38.     }
39. }

```

34. Simpan file dan ujicoba dengan menekan **Ctrl+Enter**. Jika proses dan kode benar, maka transisi antar halaman akan muncul setiap tombol ditekan.

Penjelasan kode

`transisiHalaman("hal_1");` (baris 30) Pada saat tombol home ditekan, maka fungsi `transisiHalaman` akan dipanggil dengan parameter “hal_1”.

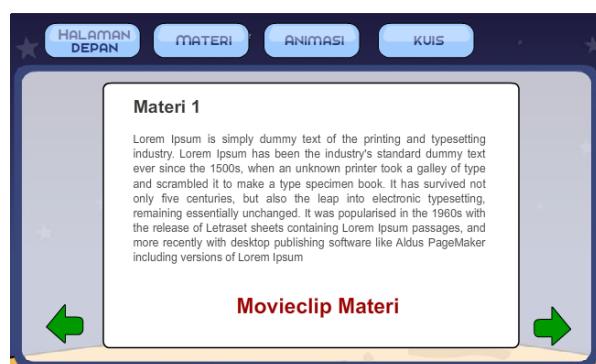
`transisi.namaHalaman = namaHalaman;` (baris 23) maka variabel yang melekat di movieclip `transisi` diubah sesuai dengan parameter tersebut, selanjutnya movieclip `transisi` dimainkan dan di cek oleh fungsi `jalankanTransisi`, ketika frame aktif berada pada posisi

15 (saat tertutup penuh), maka konten diubah menuju ke frame berlabel sesuai dengan namaHalaman.

Sub Menu (sub halaman)

Pada perencanaan multimedia pembelajaran interaktif, menu materi memiliki 2 sub menu yaitu materi 1 dan materi 2. Dalam pengembangan selanjutnya juga dapat dibuat materi 3 dan seterusnya sampai seluruh materi tercover oleh multimedia pembelajaran. Untuk membuat sub halaman, perhatikan langkah berikut :

35. Lanjutkan file di atas.
36. **Dobel klik movieclip konten** untuk mengeditnya, karena akan ditambahkan materi ke dalamnya.
37. **Klik kanan frame 10 layer sub halaman**, kemudian tambahkan **Keyframe**. Selanjutnya buatlah sebuah kotak dan teks “materi 1”. Seleksi kotak dan teks tersebut kemudian **Convert to Symbol** menjadi **movieclip “materi”**.
38. Selanjutnya buatlah **2 buah tombol “panah kanan”** dan **“panah kiri”**. Atur posisinya sesuai gambar.



Gambar 163. Posisi Movieclip materi dan tombol panah.

39. **Dobel klik movieclip materi** untuk mengeditnya. **Klik kanan frame 2** dan tambahkan **KeyFrame**. Edit konten (materi yang ada) menjadi materi selanjutnya. Pada tahapan ini materi yang dimasukkan dapat berupa teks, gambar, animasi movieclip, dan video.



Frame 1

Frame 2

Frame 3

Gambar 164. Pengaturan materi dalam movieclip “materi”

Konten dibuat secara **frame by frame** (seolah-olah satu halaman demi halaman).

Lakukan beberapa kali sampai semua materi yang ingin ditampilkan dalam multimedia interaktif tersampaikan. Prinsip utama dalam membuat materi dalam media layar adalah :

- Keterbacaan teks harus teruji, dan dapat dibaca dengan mudah oleh pengguna, untuk itu perlu digunakan huruf dengan ukuran, ketebalan dan warna yang baik.
- Jumlah kalimat atau paragraf yang ditampilkan dalam 1 halaman tidak boleh terlalu banyak, karena multimedia interaktif memiliki fitur jumlah halaman yang tidak terbatas, sehingga paragraf dan kalimat dapat didistribusikan secara merata.
- Penggunaan gambar atau animasi sebaiknya menggunakan ukuran yang sama dengan ukuran layar (tidak diperbesar / diperkecil lagi). Hal ini dimaksudkan agar gambar dapat ditampilkan dengan sempurna tanpa distorsi.
- Penambahan video dan operasi teks kompleks akan dijelaskan pada bab selanjutnya.

40. Kembali ke mode edit movieclip “konten”, kemudian tambahkan **instance name** “**materi**” pada movieclip, “**tombol_next**” dan “**tombol_prev**” pada tombol panah.

41. Selanjutnya klik kanan frame 10 layer action pilih Insert Blank Keyframe. Selanjutnya buka panel Action (F9) dan ketikkan kode berikut :

```

1. materi.stop();
2.
3. function materiNext(e:MouseEvent):void{
4.     materi.nextFrame();
    
```

```
5. }
6.
7. function materiPrev (e:MouseEvent) :void{
8.     materi.prevFrame ();
9. }
10.
11. tombol_next.addEventListener (MouseEvent.CLICK,
    materiNext);
12. tombol_prev.addEventListener (MouseEvent.CLICK,
    materiPrev);
```

42. Jalankan aplikasi dengan menekan **Ctrl+Enter**. Cobalah klik tombol materi dan klik panah kanan dan kiri. Pastikan semuanya berfungsi dengan baik.

Bab VIII

Operasi Text

Dalam multimedia pembelajaran interaktif terdapat beberapa hal terkait dengan penggunaan teks. Pada umumnya teks disajikan secara statis seperti pada materi pembelajaran, teks dinamis seperti pada kolom soal, kolom waktu dan beberapa variabel yang berubah secara dinamis, serta teks masukan seperti input nama dan input angka. Flash memiliki 3 jenis teks yang dapat dimanfaatkan untuk hal tersebut. Dalam Flash mode teks terbagi menjadi 3 yaitu :

1. Static text

Static text digunakan untuk menuliskan teks yang tidak mengalami perubahan apapun sepanjang aplikasi berjalan. Static text sesuai jika diaplikasi untuk menuliskan materi, judul, nama tombol dan segala teks yang static.

2. Dynamic text

Dynamic text merupakan teks yang dapat diatur value nya. Dengan demikian dynamic text bersifat dinamis tergantung pengaturannya dengan kode. Dynamic teks sesuai jika diaplikasikan untuk teks soal yang berubah, teks waktu, teks score dan sejenisnya.

3. Input text

Input text merupakan jenis masukan dalam bentuk teks, dimana masukan tersebut dapat ditangkap dan diproses oleh kode. Input text sesuai jika diaplikasikan untuk masukan nama pengguna, masukkan nilai-nilai variabel dalam sebuah simulasi dan sejenisnya.

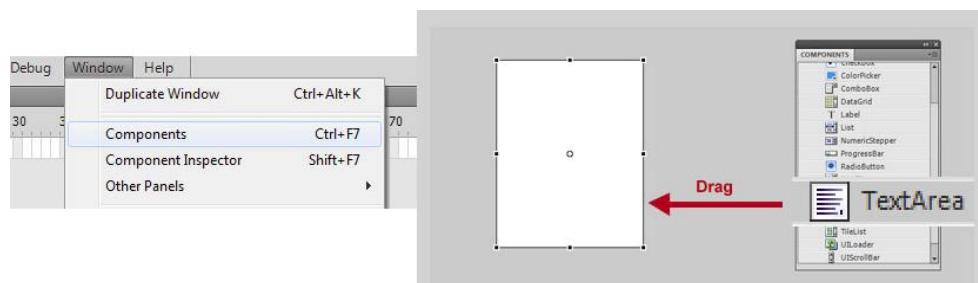
Scrolling Text

Salah satu elemen utama dari sebuah multimedia pembelajaran adalah materi dalam bentuk teks. Akan tetapi sering kali didapati *space* (ruangan) yang terbatas - sedangkan teks yang akan ditampilkan cukup banyak. Sebagai solusi atas permasalahan tersebut adalah menggunakan metode *scrolling text*. Pada dasarnya terdapat 2 cara mudah untuk menambahkan teks scroll, yaitu dengan menggunakan **component text area** dan menggunakan **dynamic text**.

Scrolling Text menggunakan Component

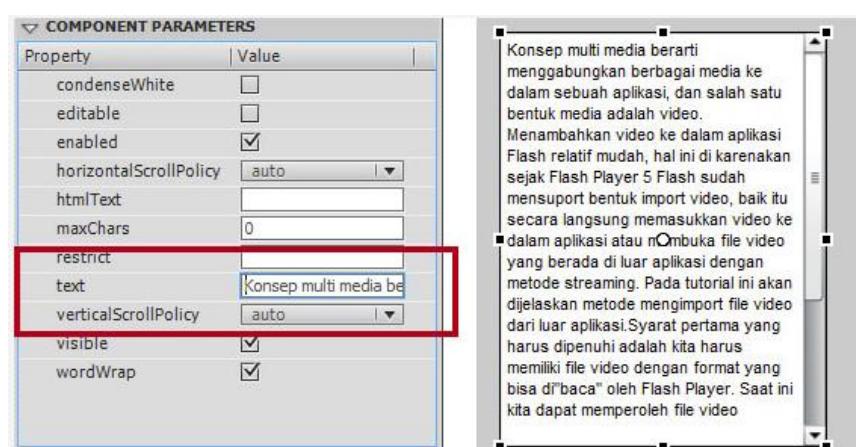
Component adalah sebuah sistem yang telah disiapkan oleh Flash untuk mempermudah elemen-elemen yang umum digunakan oleh pengembang aplikasi. Salah satu component yang dapat digunakan untuk mengolah teks adalah text area. Perhatikan langkah-langkah berikut untuk membuat scrolling teks dengan menggunakan component :

1. Pada file Flash, **klik Keyframe** tempat dimana anda akan menampilkan teks tersebut. Kemudian pilih menu **Window > Component**. Setelah panel Component muncul, drag komponen **Text Area** ke stage. Dengan menggunakan **Transform Tool** atur posisi dan ukurannya.



Gambar 165. Menu components

2. Selanjutnya, **klik TextArea** tersebut kemudian buka panel **properties**. Pada **Component Parameter** ketikan semua teks yang ingin anda tampilkan pada kolom **Text** kemudian tekan tombol **Enter**. Maka teks tersebut akan muncul pada TextArea yang ada di stage, dan apabila baris text nya melebihi ukuran area teks, secara otomatis tombol scroll akan muncul.



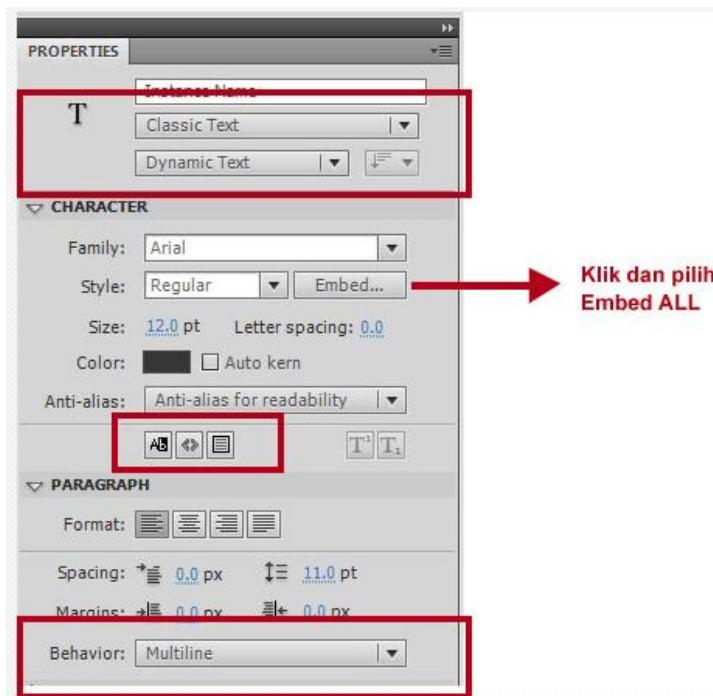
Gambar 166. Componen parameter text area

3. Jalankan aplikasi dengan menekan **Ctrl+Enter**. Text akan muncul, dan scroll langsung dapat difungsikan tanpa penambahan kode apapun.

Scrolling menggunakan Dynamic Text

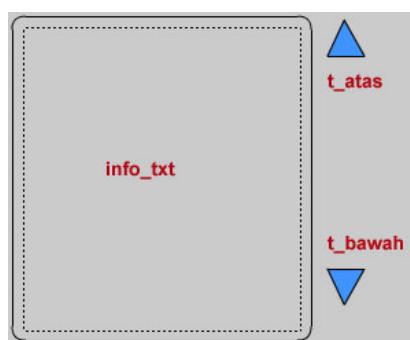
Apabila menginginkan desain yang fleksibel sesuai dengan keinginan, efek scrolling juga dapat dibuat dengan menggunakan dynamic text. Perhatikan langkah-langkah berikut :

1. Buatlah sebuah file baru atau melanjutkan contoh di atas.
2. Klik **Text Tool** kemudian perhatikan panel **properties**. Pada panel **properties**, aturlah opsinya menjadi seperti gambar berikut



Gambar 167. Pengaturan text properties

3. Drag pada stage untuk membuat sebuah area **dynamic text** sesuai dengan kebutuhan anda. Atur posisi, jenis huruf, ukuran huruf dan sebaginya
4. Selanjutnya **buatlah 2 buah tombol** yaitu tombol untuk scroll atas dan tombol untuk scroll bawah. Lihat gambar.



Gambar 168. Pengaturan teks dan tombol

5. Setelah tahapan membuat objek selesai, tahapan selanjutnya adalah memberi **instance name** pada masing-masing objek. Berikan nama **info_txt** pada dynamic text, **t_atas** pada tombol scroll atas dan **t_bawah** pada tombol scroll bawah.
6. **Buatlah sebuah layer baru** (layer 2). **Klik frame 1 layer 2 buka panel Action (F9)** dan ketikan kode berikut :

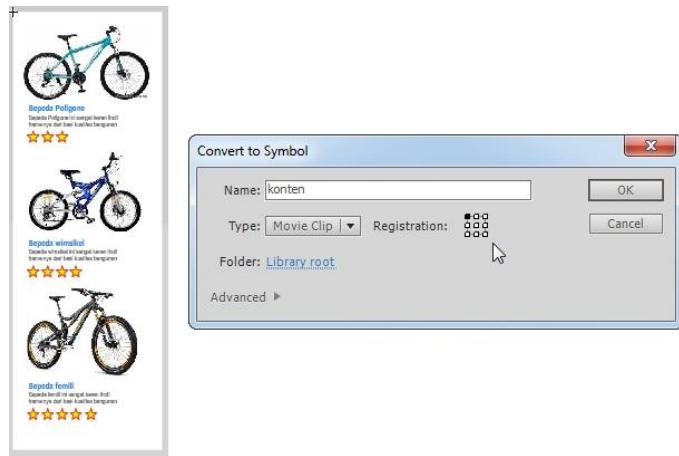
```
1. info_txt.text = "Multimedia adalah, dan seterusnya.....  
    sampai selesai";  
2.  
3. t_atas.addEventListener(MouseEvent.CLICK, scroll_atas);  
4. t_bawah.addEventListener(MouseEvent.CLICK,  
    scroll_bawah);  
5.  
6. function scroll_atas(e:MouseEvent):void{  
7.     info_txt.scrollV--;  
8. }  
9. function scroll_bawah(e:MouseEvent):void{  
10.    info_txt.scrollV++;  
11. }
```

7. Jalankan aplikasi dengan menekan **Ctrl+Enter**. Text akan muncul, dan scroll langsung dapat difungsikan dengan menekan tombol atas dan tombol bawah.

Scrolling text dengan Gambar

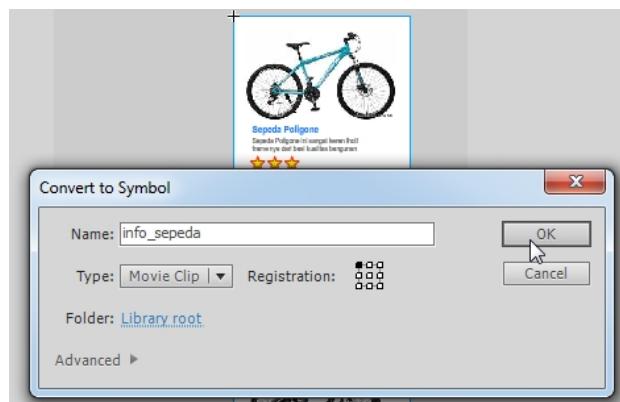
Untuk scrolling konten yang mengandung elemen grafis seperti gambar, baik component maupun dynamic text tidak mendukung, sehingga untuk melakukannya diperlukan sebuah Movieclip untuk menghandle konten yang akan di scroll. Untuk lebih jelasnya ikuti langkah berikut :

1. Siapkan gambar yang akan di tampilkan dalam konten scrolling. Dalam contoh ini akan digunakan 3 buah gambar sepeda.
2. Buatlah sebuah file baru, kemudian import gambar tersebut ke stage dengan memilih menu **File > Import > Import to Stage**, kemudian atur posisi gambar tersebut dan lengkapi dengan **static text** sehingga menjadi isi dari konten yang akan di scrol.
3. Seleksi seluruh elemen yang akan dijadikan konten scrolling tersebut, lalu **Convert to Symbol** menjadi movieClip "konten".



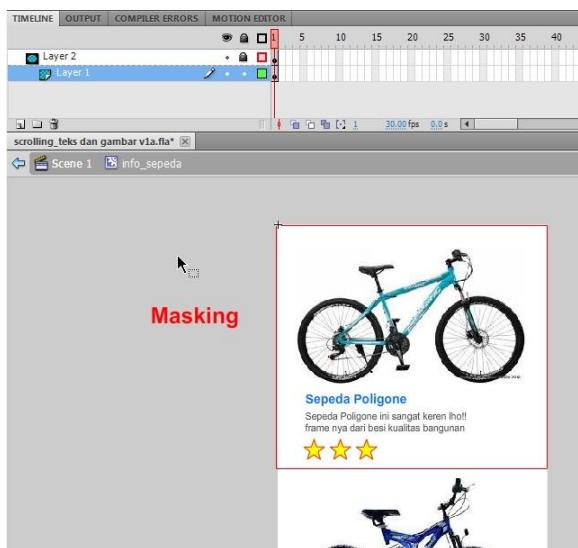
Gambar 169. Movieclip konten scroll

4. Klik movieclip "konten" tersebut buka panel **properties** dan tambahkan **instance name** "**konten**".
5. Seleksi kembali movieClip "**konten**" tersebut kemudian **Convert to Symbol** kembali menjadi movieclip "info_sepeda". Hal ini dimaksukan untuk menambahkan efek masking ke dalam movieclip konten.



Gambar 170. Mengconvert movieclip “info sepeda”

6. **Dobel klik MovieClip "info sepeda"** untuk mengeditnya, **kemudian tambahkan layer baru** dan berikan **masking** agar area yang terlihat menjadi terbatas. Langkah ini dimaksudkan agar konten yang berada diluar area scroll menjadi tertutup oleh masking.



Gambar 171. Penambahan masking

- Kembali ke scene utama dengan menekan **Ctrl+E**, kemudian **buatlah 2 buah tombol** untuk panah atas, dan panah bawah. Kemudian tambahkan **instance name** pada masing-masing objek yaitu **info_sepeda**, **up_btn** dan **down_btn**.



Gambar 172. Pengaturan instance name

- Buatlah sebuah layer baru (layer 2). Klik frame 1 layer 2 buka panel Action (F9) dan ketikan kode berikut :**

```

1. var is_drag:Boolean = false;
2. var scroll_speed:Number = 0;
3.
4. function pencet_mouse(e:MouseEvent):void{
5.     var nama:String = e.currentTarget.name;
6.     is_drag = true;
7.     if (nama == "up_btn") scroll_speed = 10;
8.     if (nama == "down_btn") scroll_speed = -10;
9. }
10. function lepas_mouse(e:MouseEvent):void{
11.     is_drag = false;
12.     scroll_speed = 0;
13. }
14.

```

```

15. function scroll_tengah(e:MouseEvent):void {
16.     info_sepeda.konten.y += e.delta * 3;
17. }
18.
19. function atur_scroll(e:Event):void{
20.     if (is_drag) info_sepeda.konten.y+=scroll_speed;
21.     if (info_sepeda.konten.y > 0) info_sepeda.konten.y =
    0;
22.     if (info_sepeda.konten.y < -info_sepeda.konten.height
    + 375) {
23.         info_sepeda.konten.y = -
    info_sepeda.konten.height + 375;
24.     }
25. }
26.
27. up_btn.addEventListener(MouseEvent.MOUSE_DOWN,
    pincet_mouse);
28. down_btn.addEventListener(MouseEvent.MOUSE_DOWN,
    pincet_mouse);
29. stage.addEventListener(MouseEvent.MOUSE_UP,
    lepas_mouse);
30.
31. info_sepeda.addEventListener(Event.ENTER_FRAME,
    atur_scroll);
32. info_sepeda.addEventListener(MouseEvent.MOUSE_WHEEL,
    scroll_tengah);

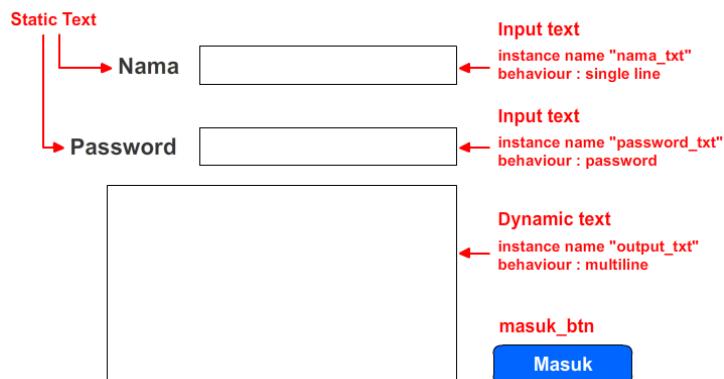
```

- Jalankan aplikasi dengan menekan **Ctrl+Enter**. Uji coba tombol dan pastikan bekerja dengan baik.

Input Text

Input text digunakan untuk membaca masukan berupa teks melalui sebuah teks area. Sebagai contoh dalam form registrasi/pendaftaran, pengguna diminta untuk memasukkan nama, alamat, password, dan sejenisnya. Masukan tersebut dapat dibuat dengan menggunakan input text. Untuk lebih memahami tentang input text, perhatikan contoh berikut :

- Buatlah sebuah file baru. Selanjutnya dengan menggunakan **text tool** buatlah beberapa text area seperti pada gambar berikut :



Gambar 173. Contoh penggunaan teks type

2. Selanjutnya buatlah sebuah **tombol “masuk_btn”**.
3. Agar dapat diakses oleh kode tambahkan **instance name** pada tombol dan teks sesuai pada gambar 173.
4. **Buatlah sebuah layer baru (layer 2). Klik frame 1 layer 2 buka panel Action (F9) dan ketikan kode berikut :**

```

1. function masuk(e:MouseEvent):void{
2.     if (nama_txt.text.length > 4) {
3.         if (password_txt.text == "flash") {
4.             output_txt.text = "Halo "+nama_txt.text+
selamat belajar operasi teks pada flash. Latihan ini
tidak sulit kan?";
5.         }else{
6.             output_txt.text = "Maaf password salah";
7.         }
8.     }else{
9.         output_txt.text = "Maaf nama harus diisi lebih
dari 4 digit";
10.    }
11. }
12.
13. masuk_btn.addEventListener(MouseEvent.CLICK, masuk);

```

5. Jalankan aplikasi dengan menekan **Ctrl+Enter**. Uji coba dengan memasukkan nama dan password bervariasi. Kemudian gunakan password yang benar yaitu "flash". Klik tombol masuk dan pada dynamic text akan muncul hasilnya.

Penjelasan program :

```
if (nama_txt.text.length > 4) { digunakan untuk mendeteksi apakah teks yang diketikkan pada kolom nama panjangnya lebih dari 4 huruf, jika kurang maka akan muncul pesan di baris9.
```

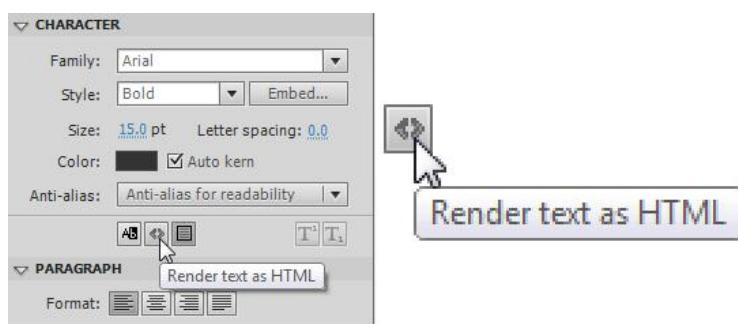
```
if (password_txt.text == "flash") { digunakan untuk memastikan apakah password yang diketikkan dalam kolom password adalah "flash".
```

```
output_txt.text = "..."; digunakan untuk mengubah value dari dynamic text.
```

HTML Text format

Jika diperhatikan pada contoh di atas, tampilan yang muncul pada dynamic text “output_text” selalu sama dari sisi jenis huruf, warna dan ukuran huruf. Sebagai contoh ketika memberikan peringatan bahwa password salah, maka sebaiknya warna teks berubah menjadi merah. Permasalahan yang muncul adalah bagaimana cara merubah teks dalam kondisi tertentu saja, maka solusi yang bisa digunakan adalah HTML text format. Untuk lebih jelasnya perhatikan langkah berikut :

6. Lanjutkan file sebelumnya. Kemudian **klik dynamic text “output_txt”** dan buka panel **properties**. Aktifkan opsi **Render text as HTML**.



Gambar 174. Opsi Render Test as HTML

7. **Klik frame 1 layer 2 buka panel Action (F9)** dan ubah kode menjadi sebagai berikut :

```
1. function masuk(e:MouseEvent):void{
2.     if (nama_txt.text.length > 4) {
3.         if (password_txt.text == "flash"){
4.             output_txt.htmlText = "Halo
5.             "+nama_txt.text+" selamat belajar operasi teks pada
6.             flash. Latihan ini tidak sulit kan?";
7.         }else{
```

```
6.                     output_txt.htmlText = "<b><font color
= '#ff0000'> Maaf password salah</font></b>";
7.                 }
8.             }else{
9.                 output_txt.htmlText = "<b> Maaf nama harus
diisi lebih dari <font color = '#ff0000'><u> 4
digit</u></font></b>";
10.            }
11.        }
12.
13. masuk_btn.addEventListener(MouseEvent.CLICK, masuk);
```

14. Jalankan aplikasi dengan menekan **Ctrl+Enter**. Uji coba dengan memasukkan nama dan password yang salah.

Penjelasan program :

`output_txt.htmlText = "....";` dengan merubah text menjadi **htmlText**, maka dynamic text dapat menampilkan beberapa **<tag>** html seperti bold ****, underline **<u>**, pengaturan huruf ****, breakline **
** dan beberapa tag dasar lainnya.

Bab IX

Audio dan Video

Elemen penting berikutnya dalam multimedia pembelajaran interaktif adalah audio dan video. Keberadaan audio dan video yang tepat mampu menaikkan pemahaman pengguna media, oleh karena itu pengembang multimedia pembelajaran interaktif wajib menambahkan audio dan video ke dalam media yang dibuat.

Menambahkan Suara (Audio)

Suara pada interaktif multimedia bisa berupa background musik, efek suara ketika tombol ditekan, suara dubing dari sebuah karakter, atau suara efek yang lain. Flash mensupport beberapa tipe suara, akan tetapi yang lebih umum untuk digunakan adalah suara berformat WAV dan MP3 (lebih spesifik lagi yaitu suara dengan frekuensi dibawah 44.1). Terdapat 3 cara untuk menambahkan suara ke aplikasi yang kita buat yaitu :

1. Drag and drop symbol suara dari library ke stage.
Cara ini cukup mudah, akan tetapi memiliki kelemahan karena suara akan sulit dikendalikan sehingga sering kali terjadi penumpukan suara.
2. Memasukkan file suara langsung ke dalam aplikasi dan mengaturnya dengan kode.
3. Membuka file suara dari luar aplikasi.

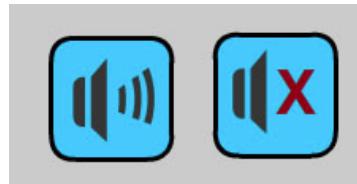
Untuk memahami proses penambahan suara pada aplikasi, ikuti langkah-langkah berikut :

1. Buatlah sebuah file baru. Selanjutnya klik menu **File > Import > Import to Library**. Kemudian **pilih file suara** yang akan anda pakai. Setelah anda menekan tombol **OK** file suara akan masuk ke dalam **library**.



Gambar 175. Mengimport suara ke library.

2. Selanjutnya **buatlah 2 buah tombol**, masing-masing adalah **tombol play** untuk memainkan musik dan **tombol mute** untuk menghentikan musik.

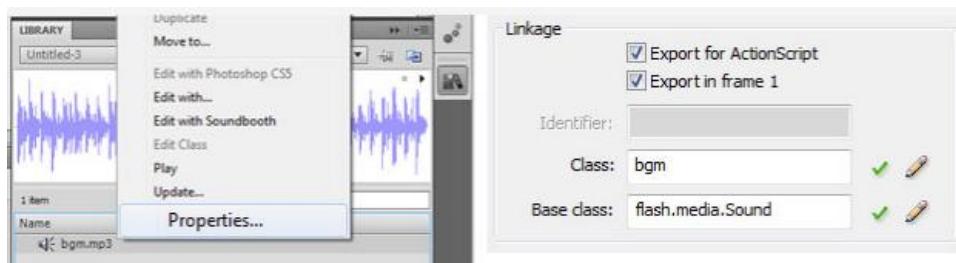


Gambar 176. Tombol play dan mute

3. Tambahkan **instance name** “**t_play**” untuk tombol play dan “**t_mute**” untuk tombol mute.

Untuk objek suara, kita tidak dapat memberikan nama instance melalui panel properties, dikarenakan symbol suara tidak memiliki bentuk fisik, sehingga agar suara bisa diakses menggunakan kode suara harus memiliki **Linkage**.

Untuk menambahkan linkage buka panel **Library** (**window > library**) kemudian **klik kanan symbol suara** dan pilih **Properties**. Klik tombol **advanced**, kemudian klik opsi **Export for Actionscript** dan ketikan **bgm** pada kolom **Class**. Klik **OK** dan akan anda dapatkan **linkage bgm** pada library.



Gambar 177. Mengatur linkage symbol suara

4. **Buatlah sebuah layer baru** (layer 2). **Klik frame 1 layer 2 buka panel Action (F9)** dan ketikan kode berikut :

```

1. var musik:bgm = new bgm(); //membuat objek suara dari
   linkage bgm
2. musik.play(0,999); // memainkan musik loop 999x
3.
4. function atur_musik(e:MouseEvent):void{
5.     var nama_tombol:String = e.currentTarget.name;
6.     if (nama_tombol == "t_play"){
7.         SoundMixer.stopAll();
8.         musik.play(0,999);
9.     }
10.    if (nama_tombol == "t_mute"){
11.        SoundMixer.stopAll(); //mengentikan semua suara
12.    }
13. }
14.
15. t_play.addEventListener(MouseEvent.CLICK, atur_musik);
16. t_mute.addEventListener(MouseEvent.CLICK, atur_musik);

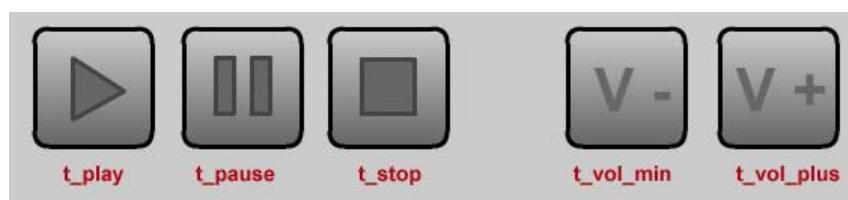
```

- Jalankan aplikasi dengan menekan **Ctrl+Enter**. Maka suara akan terdengar dan dapat diatur dengan ke 2 tombol tersebut.

Memainkan File Suara di Luar Aplikasi

Cara lain dalam menambahkan audio ke aplikasi multimedia pembelajaran interaktif adalah dengan memainkan file suara yang berada di luar aplikasi. Keuntungan metode ini adalah ukuran file aplikasi yang tidak terpengaruh oleh ukuran file suara. Akan tetapi kelemahan dari metode ini adalah file suara harus selalu disertakan bersama aplikasi agar tidak menghasilkan error. Untuk memahami proses membuka suara eksternal berikut tombol pengatur suara seperti tombol play lagu, pause lagu dan stop lagu, ikuti langkah berikut :

- Siapkan file suara bertipe **MP3** atau **WAV** kemudian **copy ke folder yang sama dengan folder aplikasi**, sebagai contoh digunakan file bernama **bgm.mp3**.
- Buatlah sebuah file baru, kemudian buatlah 5 buah tombol pengatur suara, tambahkan **instance name** pada masing-masing tombol tersebut. (Lihat gambar).



Gambar 178. Tombol pengatur suara

3. Buatlah sebuah layer baru (layer 2). Klik frame 1 layer 2 buka panel Action (F9) dan ketikan kode berikut :

```
1. var lagu:Sound = new Sound();
2. lagu.load(new URLRequest("bgm.mp3"));
3.
4. var musik:SoundChannel = new SoundChannel();
5. musik = lagu.play(0,1);
6.
7. var pengatur_suara:SoundTransform = new
   SoundTransform();
8. var suara_aktif:Boolean = true;
9. var posisi:Number = 0;
10.
11. function tombolSuara(e:MouseEvent):void{
12.     var nama_tombol:String = e.currentTarget.name;
13.     if (nama_tombol == "t_play"){
14.         if (! suara_aktif){
15.             musik = lagu.play(posisi,1);
16.             suara_aktif = true;
17.             musik.soundTransform = pengatur_suara;
18.         }
19.     }
20.     if (nama_tombol == "t_pause"){
21.         posisi = musik.position;
22.         musik.stop();
23.         suara_aktif = false;
24.     }
25.     if (nama_tombol == "t_stop"){
26.         posisi = 0;
27.         musik.stop();
28.         suara_aktif = false;
29.     }
30.     if (nama_tombol == "t_vol_min"){
31.         pengatur_suara.volume = pengatur_suara.volume -
   0.2;
32.         if (pengatur_suara.volume <= 0){
33.             pengatur_suara.volume = 0;
34.         }
35.         musik.soundTransform = pengatur_suara;
36.     }
37.     if (nama_tombol == "t_vol_plus"){
38.         pengatur_suara.volume = pengatur_suara.volume +
   0.2;
39.         if (pengatur_suara.volume >= 1){
40.             pengatur_suara.volume = 1;
41.         }
42.         musik.soundTransform = pengatur_suara;
43.     }
44. }
45.
46. t_play.addEventListener(MouseEvent.CLICK, tombolSuara);
47. t_pause.addEventListener(MouseEvent.CLICK, tombolSuara);
```

```
48. t_stop.addEventListener(MouseEvent.CLICK, tombolSuara);  
49. t_vol_min.addEventListener(MouseEvent.CLICK,  
    tombolSuara);  
50. t_vol_plus.addEventListener(MouseEvent.CLICK,  
    tombolSuara);
```

4. Jalankan aplikasi dengan menekan **Ctrl+Enter**. Maka suara akan terdengar dan dapat diatur dengan tombol-tombol yang ada.

Menambahkan Video ke dalam Multimedia Pembelajaran

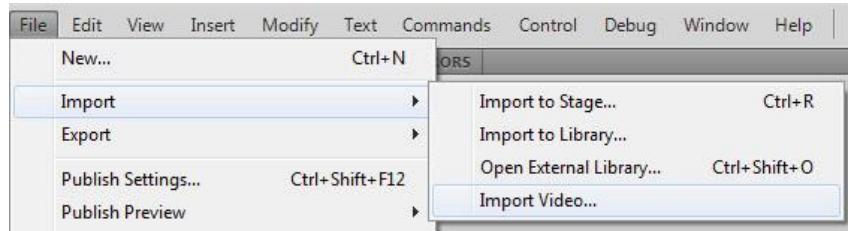
Menambahkan video ke dalam aplikasi Flash relatif mudah, hal ini di karenakan sejak Flash Player 5 Flash sudah mendukung fitur import video, baik itu secara langsung memasukkan video ke dalam aplikasi atau membuka file video yang berada di luar aplikasi dengan metode streaming. Syarat utama yang harus dipenuhi adalah file video yang digunakan memiliki format yang bisa di "baca" oleh Flash Player. Saat ini kita dapat memperoleh file video dengan berbagai format seperti AVI, WMV, 3GP, MOV, MKV, FLV, MP4 dan sebagainya. Akan tetapi Flash hanya bisa membuka file-file jenis tertentu dengan codec tertentu.

Pada umumnya file yang bisa dibaca oleh flash player adalah FLV, AVI, MP4, dan MPG. Cara termudah untuk mendeteksi apakah file video anda bisa dibaca oleh flash adalah dengan melakukan proses import, seperti yang akan dijelaskan berikutnya. Apabila setelah dipublish video tidak muncul, berarti format video masih salah dan harus anda ubah terlebih dahulu menjadi format FLV. Video yang anda unduh dari Youtube misalnya atau situs sejenis biasanya berformat FLV, namun demikian file tersebut biasanya tidak bisa anda import, hal ini dikarenakan codec yang berbeda - sehingga anda harus mengconvert ulang file tersebut.

Untuk mengconvert video menjadi format lain anda bisa menggunakan software video editor seperti Adobe Premiere, Windows Movie Maker, Videopad, dan sejenisnya.

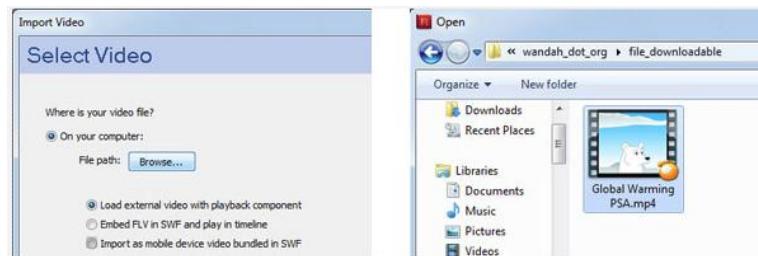
Setelah file siap, pastikan file tersebut berada dalam satu folder yang sama dengan file FLA aplikasi, hal ini akan mempermudah anda dalam menuliskan path (lokasi file video). Selanjutnya ikuti langkah-langkah berikut :

- Buatlah sebuah file baru atau melanjutkan projek yang sedang dikembangkan. Selanjutnya klik frame tempat dimana anda akan meletakkan video tersebut. Kemudian pilih menu **File > Import > Import Video**.



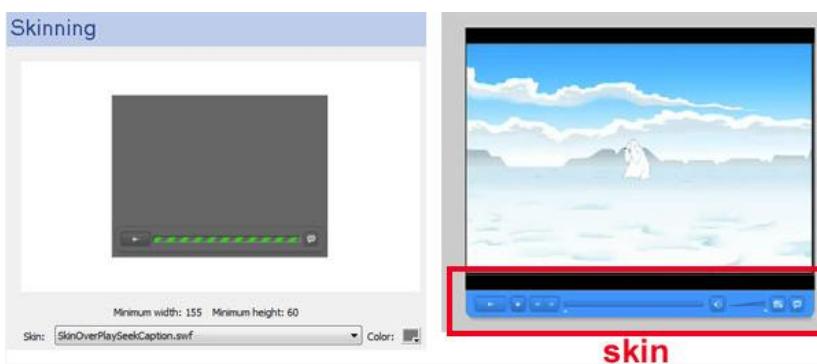
Gambar 179. Import video

- Kemudian panel **Select Video** muncul. Kemudian klik **browse** dan pilih file video yang akan anda import, selanjutnya klik tombol **next**.



Gambar 180. Panel Select video

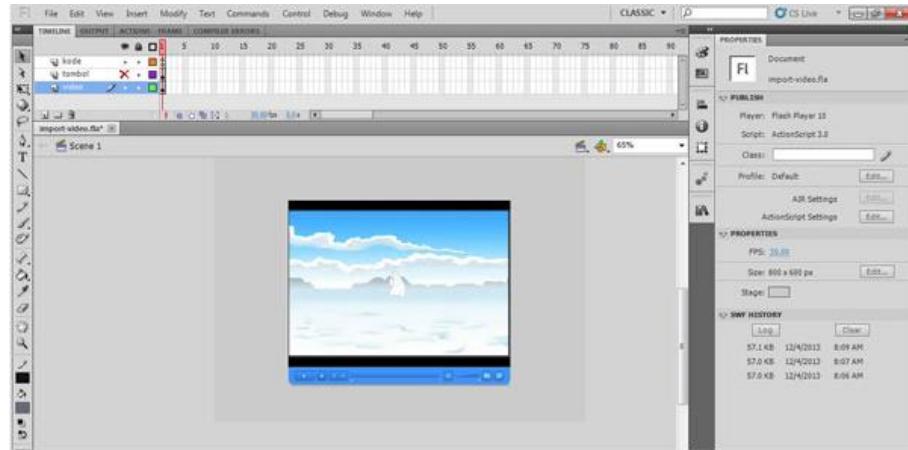
- Kemudian pada bagian **Skinning**, pilihlah **skin** yang anda inginkan. Skin adalah bagian dari video playback component yang dapat digunakan untuk mengontrol video, skin disini memiliki tombol seperti play, pause, stop, dan tombol suara. Tombol-tombol tersebut dapat berfungsi secara otomatis tanpa menambahkan kode. Selanjutnya klik tombol **next**.



Gambar 181. Skinning video

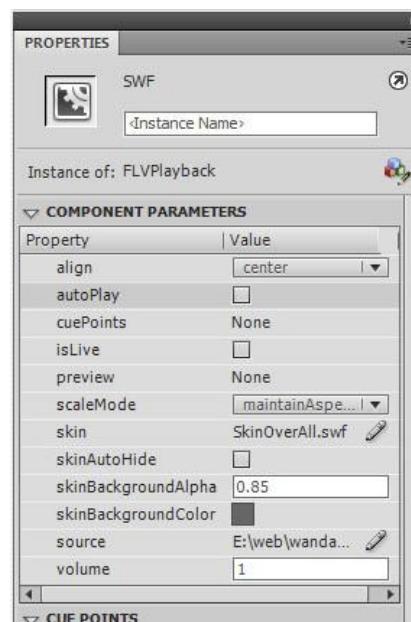
- Lanjutkan proses dengan menekan tombol **Next** sampai dengan tombol **Finish**. Setelah itu video akan muncul di stage, lengkap dengan skin nya.

- Jalankan movie dengan menekan Ctrl+Enter dan test apakah video dapat dimainkan. Apabila video tidak muncul, berarti ada kesalahan (yang paling umum adalah salah format, dan salah folder).



Gambar182. Video yang berhasil di import ke dalam aplikasi

- Jika proses diatas benar, akan anda dapat video tersebut langsung di mainkan. Ada dapat mengubah opsi video dengan membuka panel **Properties** di bagian **component parameter**. Pada panel tersebut terdapat beberapa opsi yang dapat diubah seperti opsi auto play, skin, warna skin, auto hide dsb. Atur opsi tersebut sesuai dengan kebutuhan.



Gambar 183. Pengaturan properties component video

Menambahkan tombol Play alternatif

Apabila anda tidak suka dengan skin yang telah disediakan dan ingin membuat tombol alternatif pengatur video (seperti tombol play dan pause) maka ikutilah langkah-langkah berikut :

7. Klik video, kemudian buka **Properties**. Selanjutnya di bagian **component parameter** pilih **skin none** untuk menghilangkan skin dari video tersebut.
8. buatlah sebuah layer baru di atas layer video, kemudian ubah nama layer menjadi layer **tombol**. **Klik frame 1 layer tombol**, kemudian buatlah sebuah **tombol play**.



Gambar 184.Tombol play alternatif

9. Untuk **tombol stop** atau **pause** kita membutuhkan sebuah tombol yang tidak terlihat. Untuk membuatnya - **Klik layer tombol**, kemudian buatlah sebuah kotak dengan ukuran yang sama dengan ukuran video. Seleksi kotak tersebut dan **Convert to Symbol** menjadi **button** dengan nama “**invisible**”.
10. **Dobel klik tombol invisible**, untuk mengeditnya. Kemudian **drag keyframe UP** menuju frame **HIT**.



Gambar 185. Mengatur tombol invisible

11. Kembali ke scene 1 dengan menekan **Ctrl+E**.
12. Tahapan selanjutnya adalah memberi **instance name** pada masing-masing object. Berikan nama **videoku** untuk video, **t_play** untuk tombol play dan **t_stop** untuk tombol invisible.
13. **Buatlah sebuah layer baru** (layer 2). **Klik frame 1 layer 2** kemudian ketikan kode berikut :

```
1. function klikVideo(e:MouseEvent):void{  
2.     var nama_tombol:String = e.currentTarget.name;  
3.     if (nama_tombol == "t_play") {  
4.         videoku.play();  
5.         t_play.visible = false;  
6.         t_stop.visible = true;  
7.     }  
8.     if (nama_tombol == "t_stop") {  
9.         videoku.stop();  
10.        t_play.visible = true;  
11.        t_stop.visible = false;  
12.    }  
13.}  
14.  
15. t_play.addEventListener(MouseEvent.CLICK, klikVideo);  
16. t_stop.addEventListener(MouseEvent.CLICK, klikVideo);
```

14. Jalankan aplikasi dengan menekan **Ctrl+Enter**. Klik pada video untuk memainkan video dan klik kembali untuk menghentikan sementara (pause)

Bab X

Aplikasi Kuis

Salah satu bentuk kompleksitas dari multimedia pembelajaran interaktif adalah adanya evaluasi. Evaluasi dapat digunakan sebagai salah satu indikator atau tolak ukur untuk menguji kefektifan penyampaian materi. Salah satu bentuk evaluasi yang umum ditemukan adalah kuis. Model kuis dengan soal yang acak dan jawaban acak menjadi sebuah daya tarik tersendiri dalam sebuah media pembelajaran. Akan tetapi para pemula pemrograman biasanya mengalami kesulitan dalam membuat sistem soal acak, jawaban acak dan sistem penilaian. Pada tutorial berikut ini akan dijelaskan secara sederhana tentang pembuatan kuis interaktif dengan sistem pengacakan soal, pengacakan jawaban dan sistem penilaian. Hasil akhir dari aplikasi ini dapat dikembangkan lebih lanjut menjadi sesuatu yang lebih kompleks semacam game Who Want to be Millionaire, Who has biggest brain dan sebagainya. Hasil akhir yang diharapkan kurang lebih sebagai berikut :



Gambar 186. Rencana layout kuis

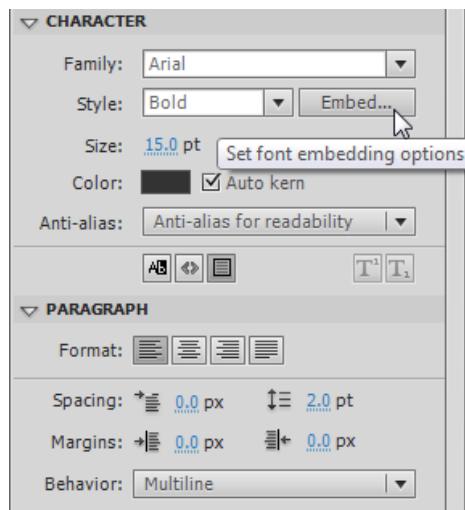
Untuk membuat aplikasi kuis tersebut, perhatikan langkah-langkah berikut :

1. Buatlah sebuah file baru. Selanjutnya buatlah sebuah tampilan untuk aplikasi kuis meliputi background, beberapa **static text** sebagai pemanis tampilan, dan **dynamic text** untuk **soal** dan untuk **score**. Perhatikan gambar :



Gambar 187. Pengaturan awal asset aplikasi kuis

2. Seleksi **dynamic text** untuk soal kemudian buka panel **properties** kemudian atur jenis huruf, matikan opsi **selectable** dan opsi **line type** menjadi **multi line**. Opsi multi line digunakan agar sebuah dynamic text mampu menampilkan beberapa baris teks. Selain itu anda harus menekan tombol embed untuk menyertakan file huruf ke dalam aplikasi, sehingga aplikasi dapat dijalankan di semua tempat tanpa mengalami distorsi huruf. Perhatikan gambar :



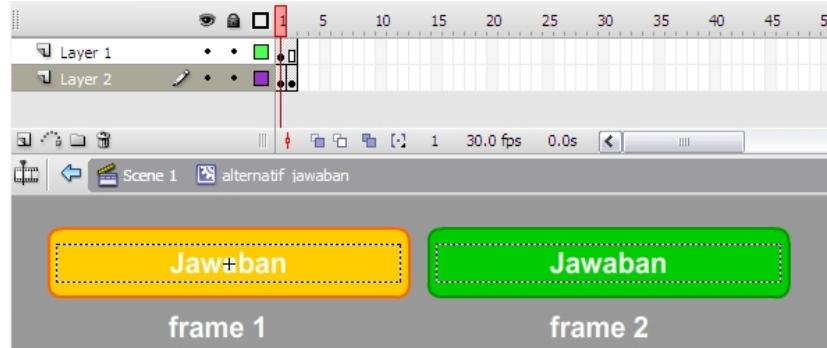
Gambar 188. Pengaturan dynamic text

3. Untuk membuat tombol pilihan jawaban, buatlah sebuah **Movieclip** baru dengan memilih menu **Insert > New Symbol**. Berikan nama “**alternatif jawaban**” dan **registrasi point** berada di tengah, lalu tekan **OK**.

Selanjutnya **buatlah 2 buah Layer**, pada **Layer 2** (atas) buatlah sebuah **dynamic text** untuk menampilkan jawaban. Klik dynamic text tersebut, buka panel **properties** dan tambahkan **instance name "jawaban_txt"**. Pada **Layer 1** (bawah) buatlah **sebuah background** berupa kotak.

Untuk mendapatkan efek **mouse over**, klik **frame 2 layer 1** tambahkan **KeyFrame** (F6), lalu ubah warna background jawaban menjadi warna lain.

Selanjutnya **klik frame 2 layer 2** dan tambahkan **Frame** (F5). (Catatan : Efek mouse over dapat ditampilkan secara otomatis pada symbol bertipe button, akan tetapi button atau tombol tidak dapat menampilkan objek lain seperti dynamic text di dalamnya, sehingga kita harus menggunakan symbol bertipe Movieclip). Perhatikan gambar :



Gambar 189. Susunan movieclip “alternatif jawaban”

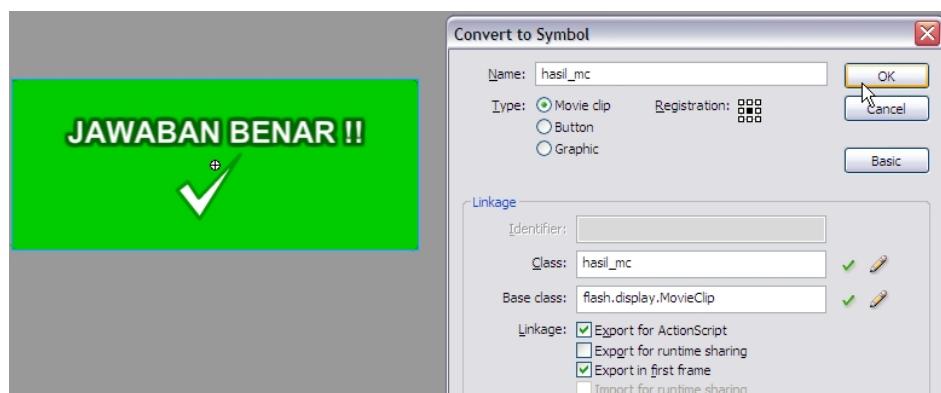
4. Kembali ke Scene 1 dengan menekan **Ctrl+E**, kemudian drag **Movieclip alternatif jawaban** dari **Library** ke **Stage** sebanyak 4 kali. Atur posisinya masing-masing.
5. Agar masing-masing objek dapat diakses oleh kode, maka perlu ditambahkan **instance name**, klik objek, kemudian buka panel **properties**, dan ketikan instance name sesuai dengan gambar berikut:



Gambar 190. Pengaturan instance name

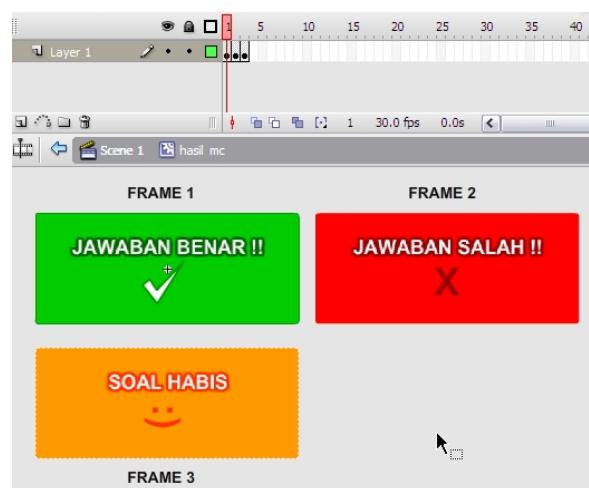
6. Kemudian untuk menampilkan apakah jawaban pemain benar atau salah, digunakan metode **pop up** dengan sedikit animasi. Untuk itu diperlukan sebuah movieclip yang di dalamnya terdapat beberapa pesan pop up.

Untuk membuatnya buatlah sebuah tampilan popup ketika jawaban benar seperti pada gambar, lalu seleksi dan **Convert to Symbol** dengan nama **hasil_mc**, registrasi point tepat di tengah, centang opsi **Export for Actionscript**, lalu tekan **OK**. Perhatikan gambar :



Gambar 191. Pengaturan popup jawaban pada movieclip “hasil_mc”

7. **Edit Movieclip hasil_mc** dengan cara dobel klik movieclip tersebut, kemudian **klik frame 2** dan tambahkan **Keyframe (F6)**, lalu edit konten di frame 2 menjadi tampilan ketika jawaban salah. Dengan cara yang sama pada frame 3 buatlah tampilan ketika permainan berakhir (soal telah habis). Perhatikan gambar :



Gambar 192. Struktur movieclip “hasil_mc”

8. Keluar dari mode edit dengan menekan **Ctrl+E**. Kemudian **buatlah sebuah layer baru** (layer 2). **Klik frame 1 layer 2** kemudian ketikan kode berikut :

```

1. var nilai:Number = 0;
2. var no_soal:Number = 0;
3. var max_soal:Number = 10;
4. var hasil:hasil_mc;
5. // struktur soal [0]= Soal [1-4]= jawaban, Jawaban benar
diletkan no 1
6. var soal:Array = [["Siapakah penemu mesin uap ?", "James
Watt", "Davinci", "Issac Newton", "James bond"],
7.                 ["Akar dari 676 adalah ?", "26", "24",
"16", "34"],
8.                 ["Bagian tumbuhan yang berperan penting
dalam fotosintesis adalah ?", "Klorofil", "Kambium",
"Epidermis", "Kromatin"],
9.                 ["Penulis trilogi Lord of The Ring
adalah?", "JRR Tolkien", "JK Rowling", "J Cameron", "J
Thomas"],
10.                ["Jumlah seluruh sudut segitiga siku-siku
adalah ?", "180", "270", "90", "360"],
11.                ["Provinsi termuda di Indonesia adalah ?",
"Banten", "Bangka belitung", "Batam", "Gorontalo"],
12.                ["Hukum gravitasi ditemukan oleh ?",
"Issac Newton", "Einstein", "T. A. Edison", "Copernicus"],
13.                [" $(\sin 30) \times 8 + (\sin 90) = ?$ ", "5", "8",
"0", "1"],
14.                ["Penemu arus listrik bolak balik adalah ?
", "Nicola Tesla", "T. Alpha Edison", "James Watt", "A.
Graham bell"],
15.                ["Komponen elektronik yang mengubah arus
bolak-balik menjadi searah adalah?", "Diode",
"Transistor", "Resistor", "Trafo"],
16.                ["Planet ke 5 dari Matahari adalah?", "Yupiter",
"Mars", "Uranus", "Saturnus"],
17.                ["Dinosaurus terbesar dalam film Jurasic
World adalah?", "Mousaurus", "Indominus Rex",
"Tyranosaurus", "Spinosaurus"],
18.                ["Gedung tertinggi di dunia sampai dengan
2015 adalah?", "Burj Al Khaifa", "Burj Dubai", "Beijing
Tower", "Petronas"],
19.                ["Pesawat komersial dengan kapasitas
terbesar di dunia adalah?", "Airbus 380", "Boeing 777",
"Beluga", "Antonov"],
20.                ["Juara Dunia MotoGP 2014 adalah?", "Mark
Marquez", "Jorge Lorenzo", "Valentino Rossi", "Dani
Pedrosa"]];
21.
22.
23. var temp_soal:Array;
24. var temp_jawaban:Array;
25.
26. function acak_soal():void{
27.     //mengacak soal
28.     temp_soal = soal.slice(0, soal.length);
29.     for (var i:Number = 0; i < soal.length; i++) {

```

```
30.         var acak:Number =
 31.             Math.floor(Math.random()*soal.length);
32.             var temp:Array = temp_soal[acak];
33.             temp_soal[acak] = temp_soal[i];
34.             temp_soal[i] = temp;
35.     }
36.     function restart():void{
37.         //restart kuis, semua variabel reset
38.         acak_soal();
39.         nilai = 0;
40.         no_soal = 0;
41.         max_soal = 10;
42.         score_txt.text = "0";
43.         restart_btn.visible = false;
44.     }
45.
46.     function tampilan_soal():void{
47.         //tampilan soal
48.         soal_txt.text = temp_soal[no_soal][0];
49.         //acak jawaban
50.         temp_jawaban = temp_soal[no_soal].slice(1, 5);
51.         for (var i:Number = 0; i < temp_jawaban.length; i++){
52.             var acak:Number =
 53.                 Math.floor(Math.random()*temp_jawaban.length);
54.                 var temp:String = temp_jawaban[acak];
55.                 temp_jawaban[acak] = temp_jawaban[i];
56.                 temp_jawaban[i] = temp;
57.         }
58.         //tampilan jawaban
59.         jawaban_1.jawaban_txt.text = temp_jawaban[0];
60.         jawaban_2.jawaban_txt.text = temp_jawaban[1];
61.         jawaban_3.jawaban_txt.text = temp_jawaban[2];
62.         jawaban_4.jawaban_txt.text = temp_jawaban[3];
63.
64.     function setup_awal():void{
65.         restart();
66.         tampilan_soal();
67.         //mengatur jawaban
68.         jawaban_1.stop();
69.         jawaban_2.stop();
70.         jawaban_3.stop();
71.         jawaban_4.stop();
72.         jawaban_1.addEventListener(MouseEvent.CLICK,
 73.             cek_jawaban);
74.         jawaban_2.addEventListener(MouseEvent.CLICK,
 75.             cek_jawaban);
76.         jawaban_3.addEventListener(MouseEvent.CLICK,
 77.             cek_jawaban);
78.         jawaban_4.addEventListener(MouseEvent.CLICK,
 79.             cek_jawaban);
80.         //listener untuk efek tombol
```

```

77.     jawaban_1.addEventListener(MouseEvent.MOUSE_OVER,
    mouse_over);
78.     jawaban_2.addEventListener(MouseEvent.MOUSE_OVER,
    mouse_over);
79.     jawaban_3.addEventListener(MouseEvent.MOUSE_OVER,
    mouse_over);
80.     jawaban_4.addEventListener(MouseEvent.MOUSE_OVER,
    mouse_over);
81.     //mouse out
82.     jawaban_1.addEventListener(MouseEvent.MOUSE_OUT,
    mouse_out);
83.     jawaban_2.addEventListener(MouseEvent.MOUSE_OUT,
    mouse_out);
84.     jawaban_3.addEventListener(MouseEvent.MOUSE_OUT,
    mouse_out);
85.     jawaban_4.addEventListener(MouseEvent.MOUSE_OUT,
    mouse_out);
86.     //tombol restart
87.     restart_btn.addEventListener(MouseEvent.CLICK,
    ulangi);
88. }
89. function mouse_over(e:MouseEvent):void{
90.     e.currentTarget.gotoAndStop(2);
91. }
92.
93. function mouse_out(e:MouseEvent):void{
94.     e.currentTarget.gotoAndStop(1);
95. }
96.
97. function cek_jawaban(e:MouseEvent):void{
98.     var no_jawaban:Number =
    Number(e.currentTarget.name.substr(8))-1;
99.     if (temp_jawaban[no_jawaban] ==
    temp_soal[no_soal][1]){
100.         //jawaban benar
101.         tampilan_hasil(1);
102.         nilai+=10;
103.         score_txt.text = String(nilai);
104.     }else{
105.         //jawaban salah
106.         tampilan_hasil(2);
107.     }
108. }
109.
110.function tampilan_hasil(tp:Number):void{
111.     hasil = new hasil_mc();
112.     hasil.x = 400;
113.     hasil.y = 275;
114.     hasil.gotoAndStop(tp);
115.     hasil.scaleX = 0.2;
116.     hasil.scaleY = 0.2;
117.     hasil.waktu = 0;
118.     hasil.tp = tp;

```

```

119.     hasil.addEventListener(Event.ENTER_FRAME,
120.         gerak_hasil);
121. }
122.
123.function gerak_hasil(e:Event):void{
124.     var ob:Object = e.currentTarget;
125.     if (ob.scaleX < 1){
126.         ob.scaleX+=0.1;
127.         ob.scaleY+=0.1;
128.     }
129.     if (ob.waktu > -1 && ob.tp < 3){
130.         ob.waktu++;
131.         if (ob.waktu > 60){
132.             ob.waktu = -1;
133.             //tambah no soal
134.             no_soal++;
135.             ob.removeEventListener(Event.ENTER_FRAME,
136.                 gerak_hasil);
137.             removeChild(DisplayObject(ob));
138.             tampilan_soal();
139.         }else{
140.             tampilan_hasil(3);
141.             restart_btn.visible = true;
142.         }
143.     }
144. }
145. }
146.function ulangi(e:MouseEvent):void{
147.     hasil.removeEventListener(Event.ENTER_FRAME,
148.         gerak_hasil);
149.     removeChild(DisplayObject(hasil));
150.     tampilan_soal();
151. }
152. setup_awal();

```

9. Jalankan aplikasi dengan menekan **Ctrl+Enter**. Ujicoba kuis mulai dari awal sampai akhir, dan restart kembali. Jika langkah benar, maka akan dihasilkan kuis yang dinamis dari sisi pengacakan kuis, jawaban, dan penilaian.

Untuk pengembangan selanjutnya kuis tersebut dapat ditambahkan waktu dan variasi soal dengan menggunakan XML sebagai database soal.

Membuka data XML

XML (eXtended Markup Language) adalah sebuah bahasa markup yang dapat digunakan untuk keperluan penyimpanan data secara ringkas dengan format yang dibakukan oleh W3C (format standar untuk pengkodean web / HTML). XML memiliki format yang sangat sederhana dan dapat diakses oleh beberapa aplikasi termasuk oleh kode Actionscript. Contoh dari XML sederhaa adalah sebagai berikut:

```
1  <datasoal>
2    <soal>
3      <tanya>Siapakah penemu mesin uap ?</tanya>
4      <jawab1>James Watt</jawab1>
5      <jawab2>Davinci</jawab2>
6      <jawab3>Issac Newton</jawab3>
7      <jawab4>James bond</jawab4>
8    </soal>
9  </datasoal>
10
```

Gambar 193. Contoh XML

Pada contoh XML di atas, terdapat beberapa tag < > yang dapat didefinisikan secara bebas. Tag pertama yaitu <datasoal> berisi tag <soal>, selanjutnya di dalam tag soal tersebut terdapat beberapa tag seperti tag <tanya> dan tag <jawab>. Setiap tag disebut sebagai node dan nilai dari masing-masing tag disebut sebagai node value. Untuk menuliskan kode XML tersebut dapat dilakukan dengan text editor sederhana seperti **Notepad++**.

Untuk memahami cara penggunaan XML perhatikan contoh berikut :

1. Buatlah sebuah file XML dengan menggunakan aplikasi Notepad seperti pada contoh berikut. Tambahkan data soal sesuai kebutuhan. Kemudian simpan dengan nama dataSoal.XML.

```
<datasoal>
  <soal>
    <tanya>Siapakah penemu mesin uap ?</tanya>
    <jawab1>James Watt</jawab1>
    <jawab2>Davinci</jawab2>
    <jawab3>Issac Newton</jawab3>
    <jawab4>James bond</jawab4>
  </soal>
  <soal>
    <tanya>Provinsi termuda adalah ?</tanya>
```

```

<jawab1>Banten</jawab1>
<jawab2>Bangka belitung</jawab2>
<jawab3>Batam</jawab3>
<jawab4>Gorontalo</jawab4>
</soal>
dst.....
<dataSoal>
```

2. Selanjutnya buat file baru Flash. Klik **frame 1 layer 1**, kemudian buka panel **Action (F9)** dan ketikan kode berikut :

```

1. var databaseSoal:XML;
2. var loader:URLLoader = new URLLoader();
3. loader.load(new URLRequest("dataSoal.xml"));
4. loader.addEventListener(Event.COMPLETE, olahData);
5.
6. function olahData(e:Event):void {
7.     databaseSoal = new XML(e.target.data);
8.     for (var i:int = 0; i < databaseSoal.*.length(); i++) {
9.         trace((i+1) + ". soal = " +
databaseSoal.soal[i].tanya)
10.        trace ("a. "+databaseSoal.soal[i].jawab1);
11.        trace ("b. "+databaseSoal.soal[i].jawab2);
12.        trace ("c. "+databaseSoal.soal[i].jawab3);
13.        trace ("d. "+databaseSoal.soal[i].jawab4);
14.    };
15. }
```

3. Simpan file pada folder yang sama dengan tempat file XML, kemudian jalankan dengan menekan **Ctrl+Enter**. Maka akan didapati soal dan jawaban tampil pada output panel.

Mengintegrasikan XML ke dalam aplikasi kuis

Setelah memahami proses membuka file XML dan mengolah data yang ada, maka metode tersebut dapat diintegrasikan ke dalam aplikasi kuis yang telah dibuat sebelumnya dengan metode Array. Untuk melakukannya, ubah kode pada aplikasi kuis di atas, menjadi sebagai berikut :

```

1. var nilai:Number = 0;
2. var no_soal:Number = 0;
3. var max_soal:Number = 10;
4. var hasil:hasil_mc;
5. var soal:Array = [];
6.
7. var temp_soal:Array;
```

```

8. var temp_jawaban:Array;
9.
10. var databaseSoal:XML;
11. var loader:URLLoader = new URLLoader();
12. loader.load(new URLRequest("dataSoal.xml"));
13. loader.addEventListener(Event.COMPLETE, olahData);
14.
15. function olahData(e:Event):void {
16.     databaseSoal = new XML(e.target.data);
17.     for (var i:int = 0; i < databaseSoal.*.length(); i++) {
18.         var dataSoal:Array = new Array();
19.         dataSoal.push(databaseSoal.soal[i].tanya);
20.         dataSoal.push(databaseSoal.soal[i].jawab1);
21.         dataSoal.push(databaseSoal.soal[i].jawab2);
22.         dataSoal.push(databaseSoal.soal[i].jawab3);
23.         dataSoal.push(databaseSoal.soal[i].jawab4);
24.         soal.push(dataSoal);
25.     };
26.     setup_awal();
27. }
28.
29. function acak_soal():void{
30.     //mengacak soal
31.     temp_soal = soal.slice(0, soal.length);
32.     for (var i:Number = 0; i < soal.length; i++){
33.         var acak:Number =
            Math.floor(Math.random()*soal.length);
34.         var temp:Array = temp_soal[acak];
35.         temp_soal[acak] = temp_soal[i];
36.         temp_soal[i] = temp;
37.     }
38. }
39. function restart():void{
40.     //restart kuis, semua variabel dikembalikan ke posisi
        semula
41.     acak_soal();
42.     nilai = 0;
43.     no_soal = 0;
44.     max_soal = 10;
45.     score_txt.text = "0";
46.     restart_btn.visible = false;
47. }
48.
49. function tampilkan_soal():void{
50.     //tampilkan soal
51.     soal_txt.text = temp_soal[no_soal][0];
52.     //acak jawaban
53.     temp_jawaban = temp_soal[no_soal].slice(1, 5);
54.     for (var i:Number = 0; i < temp_jawaban.length; i++){
55.         var acak:Number =
            Math.floor(Math.random()*temp_jawaban.length);
56.         var temp:String = temp_jawaban[acak];
57.         temp_jawaban[acak] = temp_jawaban[i];
58.         temp_jawaban[i] = temp;

```

```
59.    }
60.    //tampilkan jawaban
61.    jawaban_1.jawaban_txt.text = temp_jawaban[0];
62.    jawaban_2.jawaban_txt.text = temp_jawaban[1];
63.    jawaban_3.jawaban_txt.text = temp_jawaban[2];
64.    jawaban_4.jawaban_txt.text = temp_jawaban[3];
65. }
66.
67. function setup_awal():void{
68.     restart();
69.     tampilan_soal();
70.     //mengatur jawaban
71.     jawaban_1.stop();
72.     jawaban_2.stop();
73.     jawaban_3.stop();
74.     jawaban_4.stop();
75.     jawaban_1.addEventListener(MouseEvent.CLICK,
    cek_jawaban);
76.     jawaban_2.addEventListener(MouseEvent.CLICK,
    cek_jawaban);
77.     jawaban_3.addEventListener(MouseEvent.CLICK,
    cek_jawaban);
78.     jawaban_4.addEventListener(MouseEvent.CLICK,
    cek_jawaban);
79.     //listener untuk efek tombol
80.     jawaban_1.addEventListener(MouseEvent.MOUSE_OVER,
    mouse_over);
81.     jawaban_2.addEventListener(MouseEvent.MOUSE_OVER,
    mouse_over);
82.     jawaban_3.addEventListener(MouseEvent.MOUSE_OVER,
    mouse_over);
83.     jawaban_4.addEventListener(MouseEvent.MOUSE_OVER,
    mouse_over);
84.     //mouse out
85.     jawaban_1.addEventListener(MouseEvent.MOUSE_OUT,
    mouse_out);
86.     jawaban_2.addEventListener(MouseEvent.MOUSE_OUT,
    mouse_out);
87.     jawaban_3.addEventListener(MouseEvent.MOUSE_OUT,
    mouse_out);
88.     jawaban_4.addEventListener(MouseEvent.MOUSE_OUT,
    mouse_out);
89.     //tombol restart
90.     restart_btn.addEventListener(MouseEvent.CLICK,
    ulangi);
91. }
92.
93. function mouse_over(e:MouseEvent):void{
94.     e.currentTarget.gotoAndStop(2);
95. }
96.
97. function mouse_out(e:MouseEvent):void{
98.     e.currentTarget.gotoAndStop(1);
99. }
```

```

100.
101.function cek_jawaban(e:MouseEvent):void{
102.    var no_jawaban:Number =
103.        Number(e.currentTarget.name.substr(8))-1;
104.    if (temp_jawaban[no_jawaban] ==
105.        temp_soal[no_soal][1]){
106.            //jawaban benar
107.            tampilan_hasil(1);
108.            nilai+=10;
109.            score_txt.text = String(nilai);
110.        }else{
111.            //jawaban salah
112.            tampilan_hasil(2);
113.        }
114.function tampilan_hasil(tp:Number):void{
115.    hasil = new hasil_mc();
116.    hasil.x = 400;
117.    hasil.y = 275;
118.    hasil.gotoAndStop(tp);
119.    hasil.scaleX = 0.2;
120.    hasil.scaleY = 0.2;
121.    hasil.waktu = 0;
122.    hasil.tp = tp;
123.    hasil.addEventListener(Event.ENTER_FRAME,
124.        gerak_hasil);
125.}
126.
127.function gerak_hasil(e:Event):void{
128.    var ob:Object = e.currentTarget;
129.    if (ob.scaleX < 1){
130.        ob.scaleX+=0.1;
131.        ob.scaleY+=0.1;
132.    }
133.    if (ob.waktu > -1 && ob.tp < 3){
134.        ob.waktu++;
135.        if (ob.waktu > 60){
136.            ob.waktu = -1;
137.            //tambah no soal
138.            no_soal++;
139.            ob.removeEventListener(Event.ENTER_FRAME,
140.                gerak_hasil);
141.            removeChild(DisplayObject(ob));
142.            tampilan_soal();
143.        }else{
144.            tampilan_hasil(3);
145.            restart_btn.visible = true;
146.        }
147.    }
148.}

```

```
149. }
150.
151. function ulangi(e:MouseEvent):void{
152.     hasil.removeEventListener(Event.ENTER_FRAME,
153.         gerak_hasil);
154.     restart();
155.     tampilkan_soal();
156. }
```

Penjelasan program :

Jika sebelumnya format soal diatur dalam variabel **soal** yang bertipe `Array`, maka pada contoh ini variabel **soal** bernilai `Array` kosong ([]). Selanjutnya dengan fungsi `olahData`, file XML yang berisi data base soal dibuka. Dengan menggunakan operasi berulang `for loops`, data dari tiap-tiap node XML dimasukkan ke dalam variabel `soal` dengan menggunakan metode `array push`. Dengan cara ini variabel `soal` akan terbentuk secara otomatis, sesuai dengan data XML yang terbaca. Setelah proses pembentukan variabel `soal` selesai, maka fungsi `setup_awal()` baru dijalankan.

Bab XI

Candy Eye (Pemanis Tampilan)

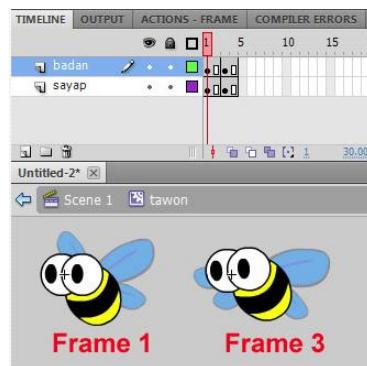
Multimedia interaktif sebagai produk desain harus memenuhi prinsip AIDA (Attention – Interest – Desire – Action), dimana kesan pertama dari aplikasi harus baik sehingga menarik pengguna untuk menjelajahi media lebih lanjut. Salah satu upaya untuk memberikan kesan pertama yang baik adalah dengan memberikan pemanis tampilan atau diistilahkan dengan *candy eye*. Beberapa efek sederhana seperti kursor mouse bergambar, percikan bintang, efek zoom dan sebagainya akan menambah daya tarik multimedia pembelajaran.

Mengubah kursor mouse

Mouse trail atau objek mengikuti mouse atau mengganti kursor mouse dengan gambar lain sangat menarik untuk ditambahkan dalam sebuah aplikasi multimedia. Pada dasarnya untuk mengubah kursor mouse, kita hanya membutuhkan sebuah objek (bisa berupa movieclip) dan mengubah kordinat layar objek tersebut mengikuti kordinat mouse.

Untuk mendeteksi posisi mouse dengan AS 3, kita dapat menggunakan kode mouseX dan mouseY. Setelah itu cukup ditambahkan sebuah listener yang dijalankan secara terus menerus - yang mengubah posisi objek agar sesuai dengan posisi mouse. Untuk lebih jelasnya ikuti langkah berikut:

1. Buatlah sebuah file baru. Selanjutnya buatlah **movieclip** baru yang akan digunakan sebagai kursor mouse dengan nama **tawon**.



Gambar 194. Animasi frame by frame movieclip tawon

2. Kemudian **klik movieclip tawon** tersebut, buka panel **properties** dan ketikan **tawon** pada kolom **instance name**.
3. **Buatlah sebuah layer baru** (layer 2). **Klik frame 1 layer 2** kemudian ketikan kode berikut :

```

1. //menyembunyikan kursor mouse
2. Mouse.hide();
3.
4. function ikut_mouse(e:Event):void{
5.     tawon.x = mouseX;
6.     tawon.y = mouseY;
7. }
8.
9. tawon.addEventListener(Event.ENTER_FRAME, ikut_mouse);

```

4. Jalankan aplikasi dengan menekan **Ctrl+Enter**, maka kursor mouse akan digantikan oleh movieclip tawon.
5. Untuk membuat gerakan tawon lebih natural dapat ditambahkan perlambatan, sehingga gerakan tawon menjadi lebih lembut dalam mengikuti kursor mouse. Untuk melakukannya, ubah kode menjadi sebagai berikut :

```

1. var perlambatan:Number = 20;
2.
3. function ikut_mouse(e:Event):void{
4.     tawon.x+= (mouseX-tawon.x)/perlambatan;
5.     tawon.y+= (mouseY-tawon.y)/perlambatan;
6. }
7.
8. tawon.addEventListener(Event.ENTER_FRAME, ikut_mouse);

```

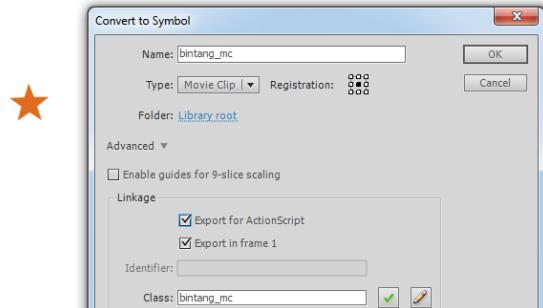
6. Jalankan kembali aplikasi dengan menekan **Ctrl+Enter**.

Efek Starburst (Percikan bintang)

Salah satu efek yang populer dipakai dalam beberapa aplikasi multimedia adalah efek percikan bintang (starburst), contoh ketika pengguna selesai melakukan aktifitas tertentu, ketika nilai evaluasi muncul, dan sebagainya. Untuk membuat efek ini perhatikan langkah berikut :

1. Buatlah file baru. Selanjutnya dengan menggunakan **Polystar Tool**, buatlah sebuah bintang dengan ukuran yang tidak terlalu besar (40 x 40pixel).

2. Seleksi bintang tersebut kemudian **Convert to Symbol** menjadi movieclip “**bintang_mc**”, registrasi point tepat di tengah dan centang opsi **Export for Actionscript** dengan nama class “**bintang_mc**”.



Gambar 195. Movieclip “bintang_mc”

3. **Hapus movieclip** tersebut dari layar. Selanjutnya **klik frame 1 layer 1**, buka panel **Action** dan ketikan kode berikut :

```

1. function gerakBintang(e:Event):void{
2.     var ob:Object = e.currentTarget;
3.     ob.x += ob.kecX;
4.     ob.y += ob.kecY;
5.     ob.kecY++;
6.     ob.waktu--;
7.     ob.scaleX *= 0.95;
8.     ob.scaleY *= 0.95;
9.     if (ob.waktu < 0){
10.         ob.removeEventListener(Event.ENTER_FRAME,
11.             gerakBintang);
12.         removeChild(DisplayObject(ob));
13.     }
14.
15. function tambahBintang(px:int, py:int):void{
16.     var bintang:bintang_mc = new bintang_mc();
17.     bintang.x = px;
18.     bintang.y = py;
19.     bintang.kecX = Math.random()*10 - Math.random()*10;
20.     bintang.kecY = -Math.random()*15;
21.     bintang.waktu = 20 + Math.random()*20;
22.     bintang.scaleX = 0.2 + Math.random();
23.     bintang.scaleY = bintang.scaleX;
24.     bintang.addEventListener(Event.ENTER_FRAME,
25.         gerakBintang);
26. }
27.
28. function starBurst(e:MouseEvent):void{
29.     for (var i:int = 0; i < 20; i++){
30.         tambahBintang(mouseX, mouseY);
31.     }

```

```

32. }
33.
34. stage.addEventListener(MouseEvent.CLICK, starBurst);

```

- Jalankan aplikasi dengan menekan **Ctrl+Enter**. Klik di area stage untuk menampilkan efek percikan bintang.



Gambar 196. Hasil efek percikan bintang.

Menampilkan Waktu

Sering kita temui dalam sebuah aplikasi atau widget sebuah blog yang menampilkan waktu secara *realtime*. Waktu dalam aplikasi terkadang juga dapat berfungsi sebagai pengingat, time counter, limitasi penggunaan software dan sebagainya. Untuk menampilkan waktu dengan kode Actionscript sebenarnya cukup mudah, yaitu dengan menggunakan class **Date**. Untuk lebih jelasnya ikuti langkah berikut :

- Buatlah file baru. Selanjutnya dengan menggunakan **Text Tool** buatlah sebuah **Dynamic Text**. Selanjutnya buka panel **properties**, berikan **instance name** “**waktu_txt**” dan atur beberapa parameter seperti ukuran font, jenis font, warna, paragraf dan opsi selectable. Jangan lupa untuk **mengembed** font agar tampil baik di semua platform.
- Buatlah sebuah layer baru** (layer 2). **Klik frame 1 layer 2** kemudian ketikan kode berikut :

```

1. var nama_hari:Array = ["Minggu", "Senin", "Selasa",
   "Rabu", "Kamis", "Jumat", "Sabtu"];
2. var nama_bulan:Array = ["Jan", "Feb", "Mar", "Apr",
   "Mei", "Jun", "Jul", "Ags", "Sep", "Okt", "Nov", "Des"];
3. var waktu:Date = new Date();
4.
5. function update_waktu(e:Event):void{
6.   var waktu_sekarang:String = "";
7.   var waktu:Date = new Date();
8.   waktu_sekarang = nama_hari[waktu.getDay()] +",
   "+waktu.getDate() + " " +nama_bulan[waktu.getMonth()] +
   "+waktu.getFullYear() +".

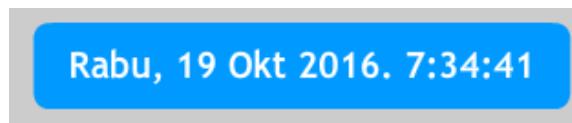
```

```

    "+waktu.getHours () +":"+waktu.getMinutes () +":"+waktu.getSeconds ());
9.     waktu_txt.text = waktu_sekarang;
10. }
11.
12. addEventListener(Event.ENTER_FRAME, update_waktu);

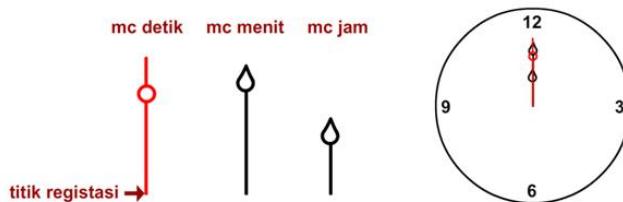
```

3. Jalankan aplikasi dengan menekan **Ctrl+Enter**. Maka akan didapati tampilan waktu secara realtime berdasarkan waktu aktif komputer.



Gambar 197. Tampilan waktu

4. Untuk menambahkan gambar jam klasik dengan jarum, **buatlah 3 buah movieclip** jarum (detik, menit dan jam) dan 1 background lingkaran. Tambahkan **instance name** “**detik**”, “**menit**” dan “**jam**”, kemudian atur posisinya sesuai gambar.



Gambar 198. Pengaturan jarum jam

5. Ubah kode pada frame 1 layer 2 menjadi sebagai berikut :

```

1. var nama_hari:Array = ["Minggu", "Senin", "Selasa",
   "Rabu", "Kamis", "Jumat", "Sabtu"];
2. var nama_bulan:Array = ["Jan", "Feb", "Mar", "Apr",
   "Mei", "Jun", "Jul", "Ags", "Sep", "Okt", "Nov", "Des"];
3. var waktu:Date = new Date();
4.
5. function update_waktu(e:Event):void{
6.     var waktu_sekarang:String = "";
7.     var waktu:Date = new Date();
8.     waktu_sekarang = nama_hari[waktu.getDay()] +",
   "+waktu.getDate() + " "+nama_bulan[waktu.getMonth()] +"
   "+waktu.getFullYear() +".
   "+waktu.getHours() + ":" +waktu.getMinutes() + ":" +waktu.getSeconds();
9.     waktu_txt.text = waktu_sekarang;
10.    //menampilkan waktu pada jarum jam
11.    detik.rotation = waktu.getSeconds()*6;

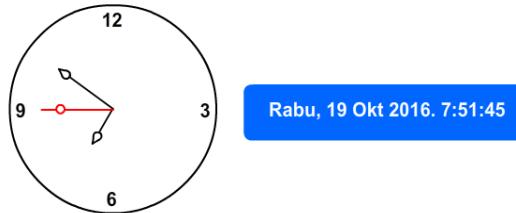
```

```

12. menit.rotation = waktu.getMinutes() *6;
13. jam.rotation = waktu.getHours() *30;
14. }
15.
16. addEventListener(Event.ENTER_FRAME, update_waktu);

```

6. Jalankan aplikasi dengan menekan **Ctrl+Enter**. Maka akan didapati tampilan jarum jam yang tepat.



Gambar 199. Tampilan jarum jam

Pada contoh diatas digunakan listener `ENTER_FRAME`, agar fungsi `update_waktu` dijalankan secara terus menerus. Di dalam fungsi tersebut terdapat kode `var waktu:Date = new Date();` sehingga aplikasi secara terus menerus mengambil data waktu yang aktif pada komputer pengguna, kemudian beberapa kode seperti `getDay`, `getDate` dan sebagainya digunakan untuk mengakses detail dari class `Date`.

Pada baris awal terdapat variabel bertipe **array** yaitu `nama_hari` dan `nama_bulan`, variabel ini diperlukan karena kode `getDay()` dan `getMonth()` menghasilkan nilai bertipe `Number` antara 0 - 7 untuk `day` dan 0-11 untuk `month`. Dengan variabel **array**, kita dapat menampilkan nama hari/bulan berdasarkan nilai yang dihasilkan, sebagai contoh `getMonth()` menghasilkan nilai 11, maka yang akan muncul adalah `nama_bulan[11]` yaitu "Des".

Membuka file SWF Eksternal

Pada beberapa kasus tertentu, diperlukan akses file SWF dari luar aplikasi. Sebagai contoh untuk menambahkan intro (animasi di awal) aplikasi, membuka materi berupa animasi SWF dan sejenisnya. Perhatikan langkah-langkah berikut, dimana aplikasi akan membuka sebuah file swf animasi bumi dan bulan yang telah dibuat sebelumnya.

1. Buatlah sebuah file baru. Selanjutnya pastikan file yang akan dibuka berada di folder yang sama.
2. **Buatlah 2 buah tombol** untuk membuka dan menutup file, kemudian tambahkan **instance name “tombol_buka”** dan **“tombol_tutup”**. Letakkan di bagian kiri bawah layar.
3. **Buatlah sebuah layer baru** (layer 2). **Klik frame 1 layer 2** kemudian ketikan kode berikut :

```
1. var nama_file:String = "animasi_bumi_bulan.swf";
2. var loader:MovieClip = new MovieClip();
3. var terbuka:Boolean = false;
4. addChild(loader);
5.
6. function bukaFile(e:MouseEvent):void{
7.     if (!terbuka){
8.         terbuka = true;
9.         var fileSWF:Loader = new Loader();
10.        var link:URLRequest = new URLRequest(nama_file);
11.        fileSWF.load(link);
12.        loader.addChild(fileSWF);
13.    }
14. }
15.
16. function tutupFile(e:MouseEvent):void{
17.     removeChild(loader);
18.     loader = new MovieClip();
19.     addChild(loader);
20.     terbuka = false;
21. }
22.
23. tombol_buka.addEventListener(MouseEvent.CLICK,
    bukaFile);
24. tombol_tutup.addEventListener(MouseEvent.CLICK,
    tutupFile);
```

4. Simpan, kemudian jalankan aplikasi dengan menekan **Ctrl+Enter**. Lalu coba klik tombol buka dan tombol tutup, jika proses benar maka file SWF yang berada di luar aplikasi dapat dibuka dan ditutup.



Gambar 200. Hasil aplikasi membuka file SWF.

Penambahan efek visual untuk mempercantik tampilan multimedia pembelajaran masih dapat dikembangkan lagi seiring berkembangnya pemahaman kita terhadap penggunaan kode.

Bab XII

Publishing

Langkah terakhir dalam proses pengembangan media pembelajaran interaktif adalah publishing. Publishing berarti menjadikan proyek yang kita buat menjadi aplikasi yang dapat dijalankan di komputer pengguna. Flash versi terbaru mendukung beberapa output diantaranya adalah :

1. SWF (Desktop dan Web Browser)

File bertipe SWF merupakan file output standart yang dihasilkan oleh aplikasi flash. File bertipe SWF dapat dijalankan langsung pada PC Desktop apabila memiliki aplikasi Flash Player (yang dapat di download di situs Adobe), atau dijalankan di browser yang telah terinstall plugin flash player.

2. Windows Projector (.exe Desktop)

Format Windows projector akan menghasilkan file bertipe EXE yang dapat dijalankan secara langsung oleh PC Desktop, meskipun komputer pengguna tidak memiliki aplikasi Flash player.

3. Adobe AIR (Desktop installer)

Format Adobe AIR hampir sama dengan format Windows Projector (.exe). Dengan format Adobe AIR hasil akhir akan berekstensi .air dan dapat diinstall di desktop sebagaimana aplikasi berekstensi exe.

4. HTML 5 (Web Browser)

Standart baru HTML yaitu HTML 5 memiliki fitur baru yaitu canvas. Mempublish flash dengan output HTML 5 akan menghasilkan file HTML yang menghandle aplikasi flash dalam format HTML canvas.

5. AIR for Android (APK)

Format baru AIR for Android ditujukan untuk pengguna mobilephone berbasis android.

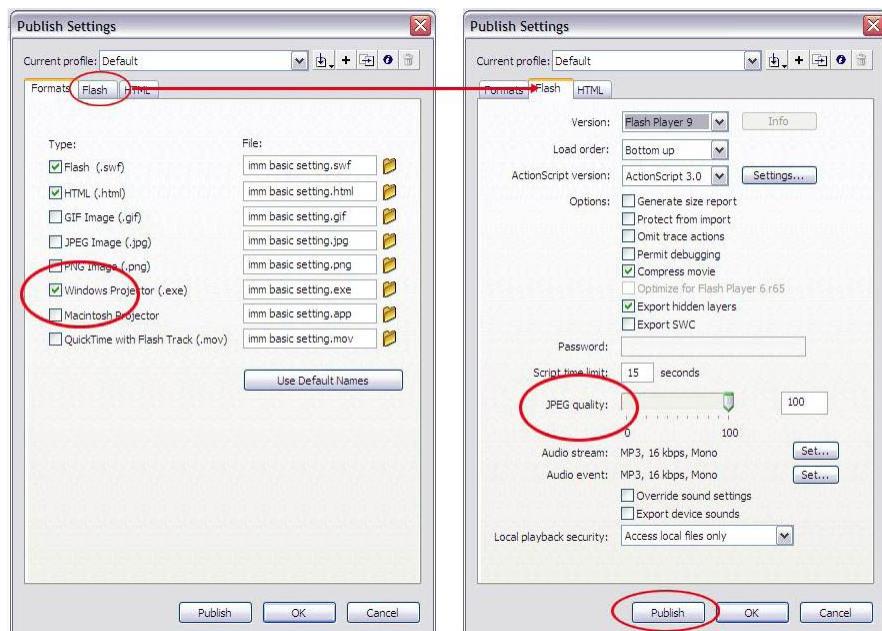
6. AIR for iOS (IPA)

Format baru AIR for iOS akan menghasilkan file berekstensi IPA dan ditujukan untuk pengguna mobilephone berbasis iOS.

Mempublish Aplikasi untuk PC Desktop

Untuk menghasilkan file bertipe exe untuk media pembelajaran interaktif yang dijalankan di PC Desktop, perhatikan langkah-langkah berikut :

1. Bukalah projek yang akan dipublish, pastikan semua kode telah bekerja dengan baik. Pilih menu **File>Publish Setting**. Agar menghasilkan file bertipe exe, centang opsi **Windows Projector (.exe)**. Kemudian klik sub option **Flash** lalu naikkan opsi **JPEG quality** menjadi **100** agar gambar tidak terdistorsi.



Gambar 201. Publish settings

2. Tekan tombol **Publish**, setelah proses publishing selesai, maka akan menghasilkan 3 buah file yaitu file bertipe EXE, SWF dan HTML. Buka Windows explorer anda dan jalankan file yang bertipe **EXE** (Aplication). Jika proses yang anda lakukan benar, saat ini movie anda sudah menjadi sebuah Interactive Multi Media yang bisa dijalankan di sebagian besar komputer .

| Name | Date modified | Type | Size |
|---|--------------------|----------|-----------|
|  multimedia interaktif sub halaman.exe | 10/18/2016 3:59 PM | EXE File | 10,309 KB |
|  multimedia interaktif sub halaman.swf | 10/18/2016 3:59 PM | SWF File | 123 KB |

Gambar 202. Hasil proses publishing bertipe exe.

Pada dasarnya proses publishing tidak berhenti sampai disini, terdapat beberapa hal yang harus ditambahkan dan diperhatikan, diantaranya adalah :

1. Tampilan media pembelajaran interaktif yang baik adalah tampil **fullscreen**. Untuk menjadikan fullscreen, pada awal kode tambahkan kode berikut :

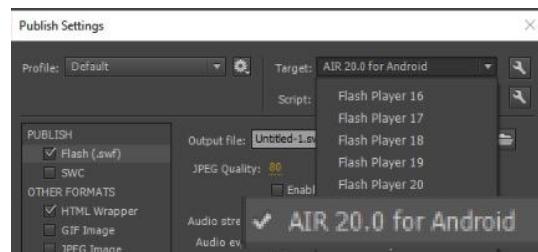
```
stage.displayState = StageDisplayState.FULL_SCREEN;
```
2. Penambahan intro, tombol keluar, dan beberapa pemanis tampilan seperti animasi, kursor yang kustom dan sebagainya dapat menambah daya tarik media.
3. Aplikasi dapat dipublish ulang menggunakan software-software tambahan untuk menambah fitur seperti mencegah klik kanan, menganti icon, membuat installer dan sebagainya. Salah satu aplikasi yang dapat dipakai adalah aplikasi MDM Zinc.
4. Konten aplikasi dapat menampilkan objek 3 dimensional. Untuk menambahkan fitur 3 Dimensi digunakan aplikasi tambahan seperti Flare3D atau Away 3D dan akan dibahas di buku lain.

Mempublish menjadi Aplikasi Mobile Android

Pengguna mobile phone berbasi Android di Indonesia sangatlah besar. Hampir 50 juta pengguna di tahun 2016 merupakan sebuah peluang untuk mempublikasikan media pembelajaran melalui platform Android. Pada dasarnya mempublish projek Flash menjadi aplikasi Android tidak terlalu sulit, hanya saja fitur ini dimiliki oleh aplikasi Flash terbaru seperti Adobe CS6 ke atas dan versi terbaru yaitu Adobe Animate CC.

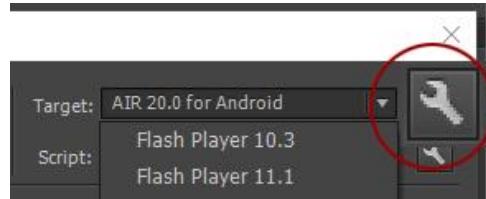
Pada contoh berikut, projek dipublish dengan menggunakan aplikasi Adobe Animate CC. Untuk lebih jelasnya perhatikan langkah berikut :

1. Bukaalah projek yang akan dipublish, pastikan semua kode telah bekerja dengan baik. Pilih menu **File>Publish Setting**. Pada opsi **target** pilih **AIR for Android**.



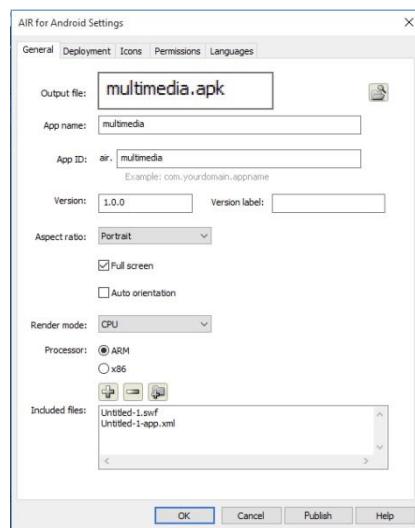
Gambar 203. Target AIR for Android

- Sebelum menekan tombol Publish, terdapat beberapa setting yang harus diatur terkait dengan kebutuhan spesifik untuk mobile phone. Untuk itu klik ikon kunci pas di pojok kanan atas.



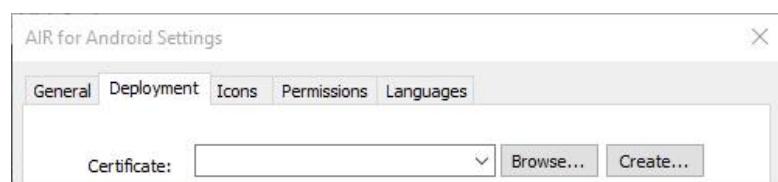
Gambar 204. Tombol setting

- Pada **General Setting**, ketikan nama pada kolom Outputfile. Atur juga Aspect ratio untuk menentukan aplikasi berjalan secara landscape atau portrait. Centang opsi Fullscreen agar aplikasi tampil penuh.



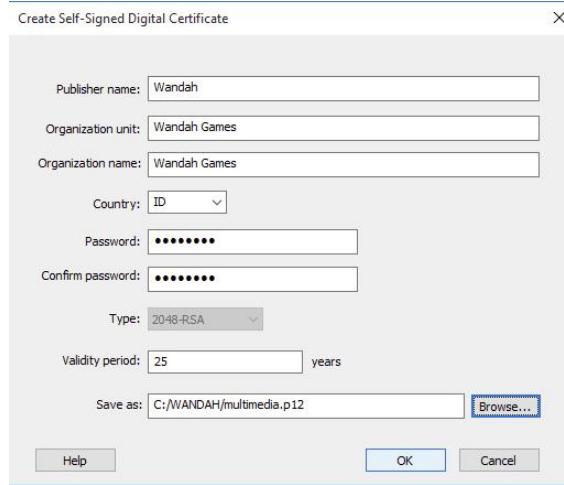
Gambar 205. General Setting

- Selanjutnya dalam setiap pembuatan aplikasi berbasis Android selalu diperlukan file **sertifikat** (.p12). Untuk itu aktifkan tab **Deployment** dan klik tombol **create** untuk membuat certificate.



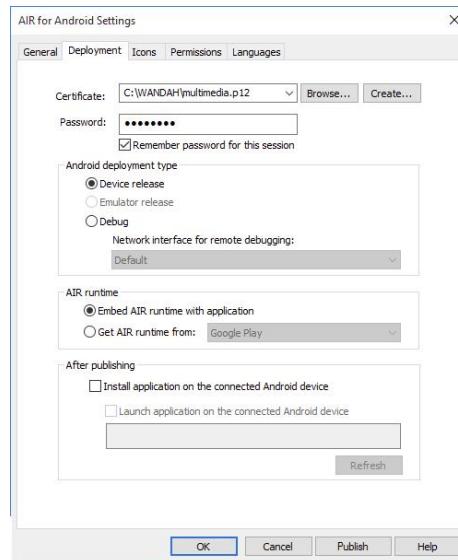
Gambar 206. Create certificate

- Isilah semua kolom isian. Isikan **nama** dan **password**. Selanjutnya klik **OK** dan file berekstensi **.p12** akan terbentuk ditandai dengan pesan sukses.



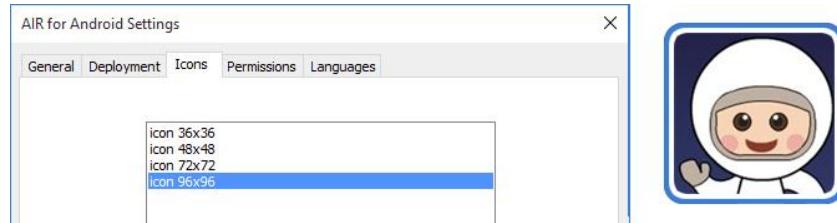
Gambar 207. Panel Create Certificate

6. Setelah itu pada panel Deployment kolom certificate akan terisi oleh sertifikat yang barusaja dibuat. Pada opsi Android deployment type aktifkan opsi Device release. Pada opsi AIR runtime aktifkan opsi Embed AIR runtime, agar pengguna langsung dapat menjalankan aplikasi tanpa menginstall Adobe AIR terlebih dahulu di mobile phonenya.



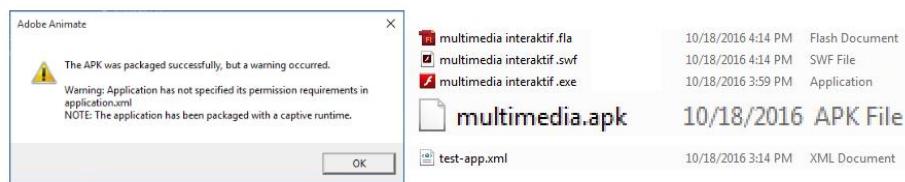
Gambar 208. Deployment setting

7. Pada panel selanjutnya siapkan file untuk **icon** aplikasi dengan beberapa ukuran yang telah ditentukan, yaitu : 36x36 pixel, 48x48 pixel, 72x72 pixel dan 96x96 pixel. Buka masing-masing file dengan menekan tombol **browse**.



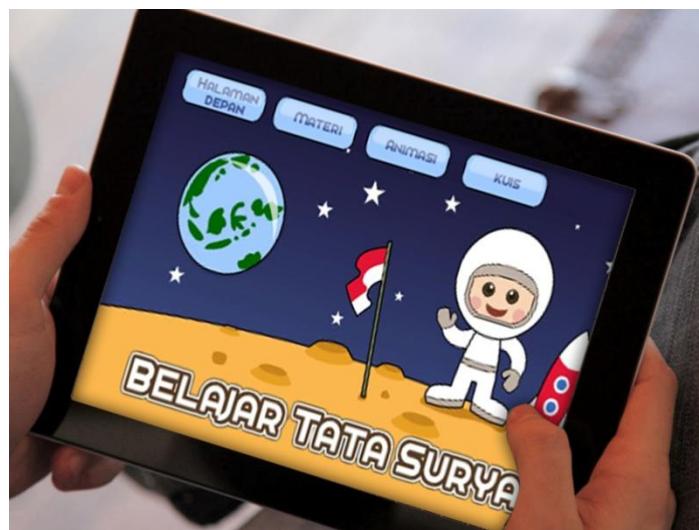
Gambar 209. Icon aplikasi

8. Untuk panel selanjutnya adalah mengatur ijin penggunaan perangkat. Untuk saat ini tidak ada fitur khusus yang membutuhkan ijin, sehingga 2 panel tersebut dapat dilewati. Kemudian klik **publish** dan proses pembentukan file APK akan berlangsung. Dalam beberapa saat kemudian file APK akan tersedia di folder tempat bekerja.



Gambar 210. File APK hasil proses publishing

9. **Copy** file **APK** tersebut ke perangkat mobile berbasis Android, **install** dan **jalankan**. Apabila tidak ada kesalahan, secara umum aplikasi akan bekerja dengan baik di berbagai perangkat dengan berbagai ukuran layar.



Gambar 211. Ujicoba media pembelajaran dengan perangkat mobile berbasis Android

Bab XIII

Penutup

Multimedia Pembelajaran Interaktif merupakan salah satu media yang dapat digunakan untuk menyampaikan materi pembelajaran kepada peserta didik dengan sangat efektif dan efisien. Keunggulan utama media pembelajaran interaktif yaitu interaktivitas itu sendiri membuka berbagai peluang interaksi antara pengguna dengan media. Namun demikian untuk membentuk interaktivitas yang baik diperlukan pengetahuan yang baik tentang desain antar muka dan teknik pemrograman, dua hal yang menjadi kelemahan bagi sebagian besar tenaga pengajar di Indonesia.

Melalui buku ini, penulis mencoba mendeskripsikan proses pembuatan multimedia pembelajaran interaktif mulai dari desain hingga teknik pemrograman. Meskipun dalam buku ini telah dibahas beberapa teknik dasar terkait proses pembuatan multimedia pembelajaran interaktif, akan tetapi pengembangan yang sesungguhnya dikembalikan kepada masing-masing pembaca.

Multimedia pembelajaran interaktif yang dihasilkan dapat digunakan secara personal, maupun dipublikasikan untuk kepentingan komersial. Secara personal penggunaan multimedia pembelajaran interaktif akan sangat membantu peserta didik dalam memahami suatu materi dengan cara baru yang menyenangkan. Secara komersial, pasar multimedia pembelajaran interaktif masih terbuka luas dengan banyaknya pemilik perangkat yang mampu menjalankan aplikasi dan masih sedikit pengembang tanah air yang melirik bidang multimedia pembelajaran interaktif secara serius.

Semoga buku ini memberikan wawasan dan manfaat bagi pembacanya, sekaligus sebagai bahan acuan dalam pengembangan multimedia pembelajaran interaktif. Pada akhirnya penulis ingin berterimakasih kepada seluruh pembaca dan semoga buku ini ditindak lanjuti dengan banyaknya multimedia pembelajaran interaktif yang berkualitas.

Daftar pustaka

- Arsyad, Azhar. 2006. Media pembelajaran. Jakarta : Raja Grafindo Persada.
- Gagné, R.M., Briggs, L.J., & Wager, W.W. 1992. Principles of Instructional Design (4th ed.). Fort Worth, TX: Harcourt Brace Jovanovich College Publishers.
- Hanssen, L dan Jankowski. 1996. Interactivity from the Prespective of Communication Studies. Luton, UK : University of Luton Press.
- Hamalik, Oemar. 1990. Sistem Internship Kependidikan Teori dan Praktek. Bandung: CV Mandarmaju.
- Lister, Martin. 2003. New media : a critical introduction. New York : Routledge.
- Murtiningsih. 2012. Yummyland : mengenal buah dan sayur. Multimedia Interaktif : Jurusan Senirupa Universitas Negeri Semarang.
- Newby, T.J., Stepich, D.A., Lehman, J.D., Russell, J.D.. (2000). Instructional technology for teaching and learning: Designing instruction, integrating computers, and using media. 2nd ed Upper Saddle River, NJ, Merrill.
- Prawiradilaga, Dewi Salma. 2008. Prinsip Desain Pembelajaran. Jakarta: Kencana.
- Safitri, Laras. 2012. Yummyland : mengenal buah dan sayur. Multimedia Interaktif : Jurusan Senirupa Universitas Negeri Semarang.
- Sugandi, Achmad. 2000. Teori Pembelajaran. Bandung : PT. Remaja Rosdakarya.
- Wibawanto, Wandah. 2013. Memprogram Game Flash 3D itu Mudah. Yogyakarta: Penerbit Andi

Tentang Penulis

Wandah Wibawanto S.Sn., M.Ds. menyelesaikan program S1 Desain Komunikasi Visual di Universitas Negeri Malang pada tahun 2006 dan S2 Magister Desain ITB pada tahun 2012. Mendalami pengembangan game flash sejak tahun 2001 dan mengkhususkan pada pengembangan game browser. Pada tahun 2006 bekerja di sebuah perusahaan asing yaitu FreeOnlineGames.com dan menghasilkan beberapa karya game yang cukup populer, di antaranya adalah Sim Taxi, Gangster Life, The Empire, dan beberapa game lainnya.

Tahun 2008 memutuskan untuk mengembangkan game flash secara independen dan menghasilkan beberapa game flash yang ditujukan untuk beberapa game portal seperti Armorgames, Gameninja, Gamebooks, DailyFreeGames, dan Froyogames.com. Mendapatkan beberapa penghargaan di bidang aplikasi games seperti pemenang INAICTA 2009 kategori digital media, Honorable mention Dictionary Games award, FOG daily game programing challenge dan beberapa lainnya.

Pada tahun 2010 penulis bergabung menjadi dosen jurusan Senirupa Universitas Negeri Semarang dengan mata kuliah khusus Multimedia Interaktif. Beberapa buku yang telah ditulis terkait dengan pengembangan Aplikasi antara lain :

- Membuat Game dengan Flash (2003) Penerbit Andi Yogyo
- Dasar-dasar Pemrograman Flash Game (2006) e-book
- Membuat Flash Game 3D (2013) Penerbit Andi Yogyo dan
- Desain dan Pemrograman Multimedia Interaktif (2017)

Beberapa karya aplikasi, tutorial dan materi perkuliahan dapat ditemukan di situs wandah.com, wandah.org dan froyogames.com

Penulis dapat dihubungi via email di wandah@wandah.com atau wandah@mail.unnes.ac.id

KESULITAN DALAM MENGIKUTI TUTORIAL?

Dapatkan File Tutorial yang ada
dalam buku ini

- 37 Tutorial
- Adobe Flash CS 5 (ke atas)
- Dokumentasi kode dalam bentuk
//comment
- softfile bisa diunduh langsung
- konsultasi online



Rp. 50.000.-



081 805 156 852

Rekening

Mandiri - 144 000 5255 127
BNI - 198 301 2806

wandah@wandah.com

Prosedur pemesanan

- Kirim pesan via WA dengan format
"Tutorial Pembelajaran Flash - nama - email"
- Transfer ke salah satu rekening
- Konfirmasi dengan mengirim bukti transfer
- Link download akan dikirim via WA
(link akan expired setelah 48 jam)



Multimedia Pembelajaran Interaktif merupakan salah satu media yang dapat digunakan untuk menyampaikan materi pembelajaran kepada peserta didik dengan sangat efektif dan efisien. Keunggulan utama media pembelajaran interaktif yaitu interaktivitas itu sendiri membuka berbagai peluang interaksi antara pengguna dengan media. Namun demikian untuk membentuk interaktivitas yang baik diperlukan pengetahuan yang baik tentang desain antar muka dan teknik pemrograman, dua hal yang menjadi kelemahan bagi sebagian besar tenaga pengajar di Indonesia.

Melalui buku ini, penulis mencoba mendeskripsikan proses pembuatan multimedia pembelajaran interaktif mulai dari desain hingga teknik pemrograman

