

BAB II

LANDASAN TEORI

2.1 Pengertian *Tutorial*

Tutorial dalam pembelajaran berbasis komputer sebagaimana diungkapkan Rusman (2008), adalah pembelajaran khusus dengan instruktur yang terqualifikasi dengan menggunakan *software* komputer yang berisi materi pelajaran yang bertujuan untuk memberikan pemahaman secara tuntas (*mastery learning*) kepada siswa mengenai bahan atau materi pelajaran yang sedang dipelajari.

Dalam tutorial, komputer berperan sebagai guru sehingga semua interaksi terjadi antara komputer dengan peserta didik sedangkan guru hanya sebagai fasilitator dan pemantau. Dalam model ini, sebenarnya *software* program komputer menggantikan sistem tutor yang dilakukan oleh guru atau instruktur. Pembelajaran dalam model ini disajikan melalui teks atau grafik yang ditampilkan oleh layar komputer. Kemudian komputer menampilkan pertanyaan sesuai dengan masalah yang disajikan. Dalam merancang interaksi yang berbentuk tutorial, saat atau setelah menyampaikan materi biasanya diikuti dengan pemberian soal atau latihan serta kasus. Jawaban pengguna terhadap soal dan kasus tersebut akan dianalisis oleh komputer dan seketika itu juga komputer akan memberikan respons dan memberikan umpan balik terhadap hasil belajar pengguna. Dalam interaksi tutorial ini, informasi dan pengetahuan yang disajikan sangat

komunikatif, seolah-olah ada pengajar yang berdiri di samping peserta didik yang memberikan pengarahan dan pembimbingan secara langsung kepada peserta didik.

2.2 Jenis – Jenis *Tutorial*

Menurut Arsyad (2011), tutorial memiliki 2 jenis diantaranya adalah :

a) Tutorial Terprogram

Tutorial terprogram merupakan seperangkat tayangan baik statis maupun dinamis yang terlebih dahulu diprogramkan. Secara berturut, seperangkat kecil informasi ditayangkan yang diikuti dengan pertanyaan. Jawaban siswa dianalisis oleh komputer (dibandingkan dengan kemungkinan-kemungkinan jawaban yang telah dirancang oleh si pembuat program atau guru), dan berdasarkan hasil analisis itu umpan balik yang sesuai. Urutan linear dan urutan bercabang digunakan. Penetapan kapan bercabang dimaksudkan untuk penyajian materi pelajaran tambahan berdasarkan hasil analisis perkembangan siswa setelah menyelesaikan beberapa latihan dan tugas.

b) *Tutorial Intellegent*

Berbeda dari tutorial terprogram karena jawaban komputer terhadap pertanyaan siswa dihasilkan oleh intelegensia artifisial (kecerdasan buatan), bukan jawaban-jawaban yang terprogram yang telah disiapkan terlebih dahulu oleh si perancang. Dengan demikian, ada dialog dari

waktu ke waktu antara siswa dengan komputer. Baik siswa maupun komputer dapat bertanya ataupun memberi jawaban.

2.3 Identitas Model *Tutorial*

Menurut Hick dan Hyde (dalam Wena, 2011). *Computer Assisted Instruction* (CAI) adalah suatu pembelajaran yang menggunakan komputer dalam menyampaikan materi ajar dan siswa dapat melakukan aktifitasnya secara langsung dengan berinteraksi melalui komputer.

Terdapat 8 identitas model Tutorial dalam pembelajaran CAI, yaitu :

- a. Pengenalan (*introduction*).
- b. Penyajian informasi (*presentation of information*).
- c. Pertanyaan dan respon (*question and responses*).
- d. Penilaian respon (*judging responses*).
- e. Pemberian feedback terhadap respon (*providing feedback about responses*).
- f. Pembetulan (*remediation*).
- g. Segmen pengaturan pengajaran (*sequencing lesson segment*).
- h. Penutup (*Closing*).

Dalam beberapa bentuk tutorial sebagaimana diungkapkan oleh Hackbarth (1996), selain menampilkan kedelapan identitas tersebut, biasanya model tutorial biasa digabung dengan soal-soal latihan yang ada dalam model *drill*, sehingga diharapkan siswa akan lebih memahami dan belajar secara tuntas.

2.4 Media Pembelajaran

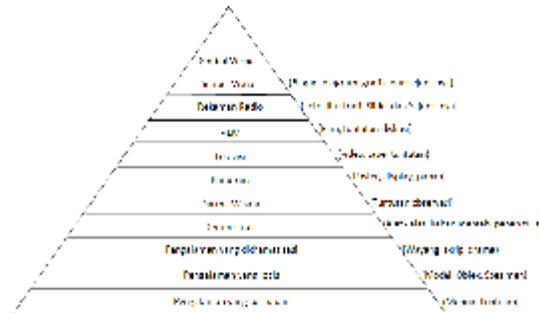
Media pembelajaran menurut Arsyad (2011), adalah alat bantu proses belajar mengajar. Segala sesuatu yang dapat dipergunakan untuk merangsang pikiran, perasaan, perhatian dan kemampuan atau ketrampilan belajar sehingga dapat mendorong terjadinya proses belajar. Batasan ini cukup luas dan mendalam mencakup pengertian sumber, lingkungan, manusia dan metode yang dimanfaatkan untuk tujuan pembelajaran atau pelatihan.

Menurut Sadiman (2003), media pembelajaran adalah segala alat fisik yang dapat menyajikan pesan serta merangsang siswa untuk belajar. Sedikit berbeda dari beberapa batasan di atas yaitu menurut asosiasi pendidikan nasional (*National Education Assosiation atau NEA*) yang dikutip oleh Sadiman menyatakan media merupakan bentuk-bentuk komunikasi baik tercetak maupun audio visual serta peralatannya. Jadi media dapat dicetak sehingga dapat dilihat, didengar dan dibaca.

Pembelajaran merupakan proses komunikasi dan berlangsung dalam suatu sistem, media pembelajaran menempati posisi yang cukup penting sebagai salah satu komponen sistem pembelajaran. Tanpa media, komunikasi tidak akan terjadi dan proses pembelajaran sebagai proses komunikasi juga tidak akan bisa berlangsung secara optimal. Media pembelajaran adalah komponen integral dari sistem pembelajaran.

Dalam usaha media sebagai alat bantu pembelajaran menurut Edgar Dale dikutip oleh Sadiman (2011), menyusun kalsifikasi pengalaman menurut tingkat dari yang paling konkret ke yang paling abstrak. Klasifikasi tersebut kemudian

dikenal kerucut pengalaman (*cone of experience*). Kerucut pengalaman Edgar Dale dapat dilihat pada gambar berikut:



Gambar 2.1 Prinsip Kerucut Pengalaman Edgar Dale

Ada beberapa jenis media pembelajaran Hernawan (2007), diantaranya :

1. Media Visual : grafik, diagram, grafik, bagan, poster, kartun, komik
2. Media Audial : radio, alat perekam, laboratorium bahasa, dan sejenisnya.
3. *Projected still media* : slide, *over head proyektor* (OHP), *in focus* dan sejenisnya.
4. *Projected motion media* : film, televisi, video (VCD, DVD, VTR), komputer dan sejenisnya.

Pada hakikatnya bukan media pembelajaran itu sendiri yang menentukan hasil belajar. Ternyata keberhasilan menggunakan media pembelajaran dalam proses pembelajaran untuk meningkatkan hasil belajar tergantung pada (1) isi pesan, (2) cara menjelaskan pesan, dan (3) karakteristik penerima pesan.

ketiga faktor tersebut mampu disampaikan dalam media pembelajaran tentunya akan memberikan hasil yang maksimal.

2.5 *Game* Edukasi

Menurut Randel (1991), bahwa *game* sangat berpotensi untuk menumbuhkan kembali motivasi belajar anak yang mengalami penurunan. Berdasarkan penelitian yang dilakukan oleh Randel mencatat bahwa pemakaian *game* sangat bermanfaat pada materi-materi yang berhubungan dengan matematika, fisika dan kemampuan berbahasa (seperti studi sosial, biologi dan logika).

Game edukasi menurut Ismail (2006), dalam bukunya *EducationGames*, yaitu suatu kegiatan yang sangat menyenangkan dan dapat merupakan cara atau alat pendidikan yang bersifat mendidik.

Dari uraian diatas dapat disimpulkan bahwa *education games* (permainan edukatif) adalah sebuah permainan yang digunakan dalam proses pembelajaran dan dalam permainan tersebut mengandung unsur mendidik atau nilai-nilai pendidikan.

Game yang memiliki konten pendidikan lebih dikenal dengan istilah *game* edukasi. *Game* berjenis edukasi ini bertujuan untuk memancing minat belajar anak terhadap materi pelajaran sambil bermain, sehingga dengan perasaan senang diharapkan anak bisa lebih mudah memahami materi pelajaran yang disajikan. Jenis ini sebenarnya lebih mengacu kepada isi dan tujuan *game*, bukan jenis yang sesungguhnya.

2.6 Kriteria *Game* Edukasi

Kriteria *game* edukasi Menurut Hurd dan Jenuings (2009), perancangan *game* edukasi yang baik haruslah memenuhi kriteria dari *education game* itu sendiri. Berikut ini adalah beberapa kriteria dari sebuah *education game* menurut Nalendra (2012), yaitu:

1. Nilai Keseluruhan (*Overall Value*)

Nilai keseluruhan dari suatu *game* terpusat pada desain dan panjang durasi *game*. Aplikasi ini dibangun dengan desain yang menarik dan interaktif. Untuk penentuan nilai, aplikasi ini menggunakan fitur nilai.

2. Dapat Digunakan (*Usability*)

Mudah digunakan dan diakses adalah poin penting bagi pembuat *game*. Aplikasi ini merancang sistem dengan *interface* yang *user friendly* sehingga user dengan mudah dapat mengakses aplikasi.

3. Keakuratan (*Accuracy*)

Keakuratan diartikan sebagai bagaimana kesuksesan model atau gambaran sebuah *game* dapat dituangkan ke dalam percobaan atau perancangannya. Perancangan aplikasi ini harus sesuai dengan model *game* pada tahap perencanaan.

4. Kesesuaian (*Compatibility*)

Kesesuaian dapat diartikan bagaimana isi dan desain *game* dapat diadaptasikan terhadap keperluan user dengan baik. Aplikasi ini menyediakan *menu* dan fitur yang diperlukan pengguna untuk membantu pemahaman pengguna dalam menggunakan aplikasi.

5. Relevan (*Relevance*)

Relevan artinya dapat mengaplikasikan isi game ke target *pengguna*. Agar dapat relevan terhadap *user*, sistem harus membimbing mereka dalam pencapaian tujuan pembelajaran. Karena aplikasi ini ditujukan untuk anak – anak, maka desain antarmuka harus sesuai dengan nuansa anak - anak, yaitu menampilkan warna - warna yang ceria.

6. Objektivitas (*Objectivity*)

Objektivitas menentukan tujuan pengguna dan kriteria dari kesuksesan atau kegagalan. Dalam aplikasi ini objektivitas adalah usaha untuk mempelajari hasil dari permainan.

7. Umpan Balik (*Feedback*)

Untuk membantu pemahaman pengguna bahwa permainan (*performance*) mereka sesuai dengan objek game atau tidak, *feedback* harus disediakan. Aplikasi ini menyajikan animasi dan efek suara yang mengindikasikan kesuksesan atau kegagalan permainan.

Menurut sumber jurnal Lakoro (2009) mencatat bahwa “Personalisasi yang ada dalam *game-game* elektronik modern dapat berubah menjadi suatu nilai penting dalam aplikasi komersial masa depan dan melayani dunia satu demi satu dimana otomasi dan sistem cerdas dibuat untuk interaksi pengguna dengan *system* secara seketika (*real time*)”.

2.7 *Mobile Learning*

Dalam jurnal Riyanto (2006), *mobile learning* memungkinkan pembelajar dapat mengakses materi pembelajaran, arahan dan aplikasi yang berkaitan dengan materi pembelajaran kapan-pun dan dimana-pun. Walaupun sekarang *mobile learning* masih berada pada tahap awal pengembangan dan para peneliti masih mengeksplorasi setiap aspek, *mobile learning* akan menjadi cukup pesat dalam jangka waktu dekat dan menurut Crawford (2003), lingkungan yang penuh dengan pembelajaran dapat diciptakan melalui *network device mobile learning*.

Ada beberapa hal yang harus diperhatikan dalam pembuatan *game* edukasi (Kenty, 2006):

1. Sebuah *game* mempunyai keaslian konteks mendalam (*historical, contemporary, hard science-fiction*).
2. Permainan *game* mempunyai keaslian konteks dalam pendidikan.
3. Keberhasilan bergantung pada pilihan dan keputusan yang cerdas.
4. Kegagalan dapat terjadi dan mengajari pemain ketika hal ini terjadi.
Memungkinkan untuk kalah dalam permainan
5. Petunjuk cara bermain harus jelas dan mudah dipahami.
6. Berbagai cara untuk memenangkan permainan.
7. Model umpan baliknya pendek sehingga siswa dapat dengan cepat untuk melihat bagaimana sebuah keputusan yang dibuat dapat mempengaruhi seluruh gambaran yang lebih besar.
8. *Game* menjadi meningkatkan tingkat tantangan dan kesulitannya.

2.8 *Mobile Operating System*

Suatu sistem operasi yang mengontrol sistem dan kinerja pada barang elektronik *mobile*, mirip dengan fungsi windows, mac os x, dan linux pada *desktop* atau *notebook* tetapi lebih sederhana. Biasanya penggunaannya hadir di ponsel pintar, PDA, tablet komputer, dan PMP.

Beberapa sistem operasi dalam *mobile* diantaranya:

1. Sistem operasi android dari google inc. (*open source, Apache*), sistem operasi yang berbasis linux dan banyak digunakan oleh para pembuat atau pabrikan telepon genggam sekarang ini (Samsung, HTC, Motorola, Sony Ericsson, LG, Huawei, dll).
2. Sistem operasi ios dari apple Inc. (*closed source, proprietary*), iOS merupakan sistem operasi buatan apple yang hanya hadir untuk produk-produk buatan apple
3. Sistem operasi symbian dari the symbian foundation (*open public license*), symbian terdiri dari beberapa sistem operasi telepon genggam yang tidak saling mendukung secara penuh beberapa edisi sistem operasi symbian yaitu S60 3rd edition, S60 5th edition, S80, UIQ, MOAP, dan symbian3 atau anna. Yang masih dikembangkan adalah yang berbasis symbian3.
4. Sistem operasi windows phone dari microsoft (*closed source, proprietary*), Windows Phone sebelumnya dinamai *windows phone 7 series* dan diperkenalkan pertama kali pada bulan februari 2010. Seri pengembangan lebih lanjut memiliki kode nama Mango.

2.9 Pengenalan Rambu Rambu Lalu Lintas di Indonesia

Rambu lalu lintas adalah bagian dari perlengkapan jalan yang memuat lambang, huruf, angka, kalimat atau perpaduan di antaranya, yang digunakan untuk memberikan peringatan, larangan, perintah dan petunjuk bagi pemakai jalan. Agar rambu dapat terlihat baik siang ataupun malam atau pada waktu hujan maka bahan harus terbuat dari material retro-reflektif.

Berdasarkan jenis pesan yang disampaikan, rambu lalu lintas dapat dikelompokkan menjadi rambu-rambu sebagai berikut :

1. Rambu Peringatan

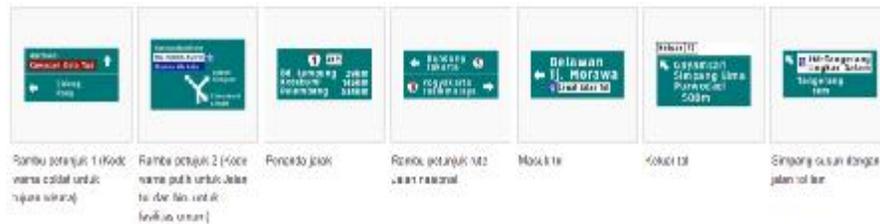
Rambu yang memperingatkan adanya kondisi berbahaya dan berpotensi bahaya agar para pengemudi berhati-hati dalam menjalankan kendaraannya. Contoh: rambu yang menunjukkan adanya lintasan kereta api, atau adanya persimpangan berbahaya bagi para pengemudi. Rambu ini berlatar belakang warna kuning dan tepian bewarna hitam.



Gambar 2.2 Gambar Macam Rambu Rambu Peringatan

2. Rambu Petunjuk

Rambu yang memberikan petunjuk atau keterangan kepada pengemudi atau pemakai jalan lainnya, tentang arah yang harus ditempuh atau letak kota yang akan dituju lengkap dengan nama dan arah letak itu berada.



Gambar 2.3 Gambar Macam Rambu Rambu Petunjuk

3. Rambu Larangan

Rambu ini untuk melarang penggunaan dan pergerakan lalu lintas tertentu. Misalnya:

- Rambu larangan berhenti.
- Rambu larangan membunyikan isyarat suara.
- Semua kendaraan dilarang lewat.



Gambar 2.4 Gambar Macam Rambu Rambu Larangan

4. Rambu Perintah

Rambu ini untuk memerintahkan penggunaan dan pergerakan lalu lintas tertentu. Misalnya:

- Rambu perintah memasuki lajur yang ditunjuk.
- Rambu batas minimum kecepatan.
- Rambu perintah bagi jenis kendaraan tertentu untuk melalui lajur dan/atau jalur tertentu.



Gambar 2.5 Gambar Macam Rambu Rambu Perintah

2.10 Prinsip Dasar Desain

Prinsip dasar desain merupakan pengorganisasian unsur-unsur dasar desain dengan memperhatikan prinsip-prinsip dalam menciptakan dan mengaplikasikan kreativitas. Berikut ini adalah beberapa prinsip dasar desain menurut Sanyoto (2005) :

- a) Keseimbangan (*balance*) adalah karya seni dan desain harus memiliki keseimbangan agar nyaman dipandang dan tidak membuat gelisah. Dalam bidang seni keseimbangan ini tidak dapat diukur tapi dapat

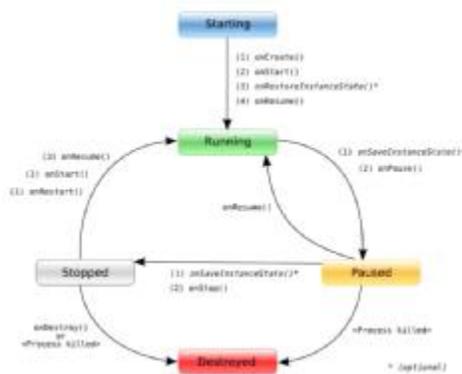
dirasakan, yaitu suatu keadaan dimana semua bagian dalam sebuah karya tidak ada yang saling membebani.

- b) Ritme atau irama (*rhythm*) adalah pengulangan gerak yang teratur dan terus menerus. Dalam bentuk-bentuk alami kita bisa ambil contoh pengulangan gerak pada ombak laut, barisan semut, gerak dedaunan dan lain-lain. Prinsip irama sesungguhnya adalah hubungan pengulangan dari bentuk-bentuk unsur desain.
- c) Kesatuan (*unity*) kesatuan merupakan salah satu prinsip dasar tata rupa yang sangat penting. Tidak adanya kesatuan dalam sebuah karya akan membuat karya tersebut tidak nyaman dipandang.
- d) Dominasi (*domination*) merupakan salah satu prinsip dasar yang harus ada dalam karya seni dan desain. Dalam dunia desain, dominasi sering juga disebut *Center of Interest*, *Focal Point* dan *Eye Catcher*.
- e) Proporsi (*proportion*) Proporsi termasuk prinsip dasar untuk memperoleh keserasian. Pada dasarnya proporsi adalah perbandingan matematis dalam sebuah bidang. Proporsi Agung (*The Golden Mean*) adalah proporsi yang paling populer dan dipakai hingga saat ini dalam karya seni rupa hingga karya arsitektur. Dalam bidang desain proporsi ini dapat kita lihat dalam perbandingan ukuran kertas dan layout halaman.

2.11 Konsep Umum Sistem Android

Perangkat berbasis android hanya mempunyai satu layar *foreground* keadaan normal saat menghidupkan android, yang pertama adalah home. dalam

menjalankan sebuah aplikasi permainan catur, *user interface* (UI) akan menumpuk diatas layar sebelumnya (home). pada saat melihat help-nya catur, maka UI help akan menimpa UI sebelumnya (catur), begitu seterusnya. Semua proses diatas direkam di *application stack* oleh *activity manager system*. Menekan tombol back hanya kembali ke halaman sebelumnya, analoginya mirip dengan browser dimana ketika saat meng-klik tombol back browser akan kembali menampilkan halaman sebelumnya. Setiap *user interface* diwakili oleh *activity class*. Setiap *activity* mempunyai siklus, dapat dilihat pada gambar dibawah berikut.



Gambar 2.6 Siklus *Interface* dalam android

2.12 Siklus Activity

Activity aplikasi android dikelola dengan sistem yang dikenal dengan *activity stack*. ketika suatu *activity start*, *activity* diletakkan pada *stack* yang paling atas dan *activity-activity* yang sudah berjalan berada dibawahnya dan akan terus berada pada posisi atas *stack* sampai muncul *activity* yang baru.

Pada dasarnya *activity* memiliki empat keadaan, yaitu:

1. *Active atau Running*

Jika *activity* berada pada posisi atas *stack*.

2. *Pause*

Jika *activity* tidak dipakai atau dibutuhkan pada suatu saat tertentu, tetapi *activity* itu masih ada atau *visible*, ketika *activity* baru yang ditangani oleh *activity system* yang lama disebut *pause* dan masih berada dalam memori, bisa jadi suatu *activity* yang sudah keadaan *pause* tidak ada dalam memori yang memungkinkan disebabkan oleh keterbatasan memori.

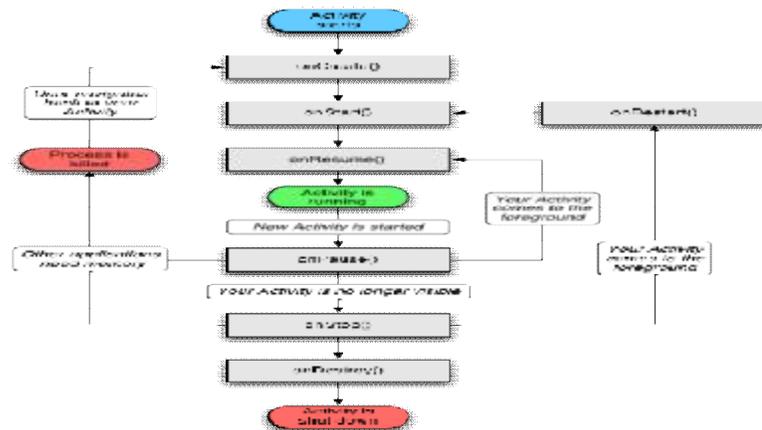
3. *Stopped*

Jika *activity* sudah tidak dipakai dan digantikan oleh *activity* lain, *activity* yang sudah berhenti tidak akan pernah dipanggil lagi, dan secara permanen memori pun tidak menyimpan info mengenai *activity* ini.

4. *Restart*

Jika *activity pause* atau *stopped*, sistem dapat mendrop *activity* ini dari sistem memori, dan ketika *user* membutuhkan *activity* tersebut, *activity* akan kembali keadaan semula, yang artinya *activity* tersebut mengalami perulangan.

Berikut ini adalah gambaran mengenai *activity lifecycle* didalam sistem android.



Gambar 2.7 *Lifecycle Activity* pada Android

Activity menurunkan sifat-sifatnya atau dengan kata lain setelah *startingpoint* melakukan *extends* kepada *activity class*, maka siklus di atas berlaku pula pada *starting point class*. Siklus sesungguhnya berisi dengan masing-masing *state* yang sebenarnya adalah suatu *method* yang dimiliki kelas *activity*.

Berikut penjelasan pada masing-masing *state*:

1. *onCreate()*

Method ini dipanggil ketika *activity* pertama kali dibuat.

2. *onStart()*

Method ini dipanggil ketika sebuah *activity* tampil ke pengguna.

3. *onResume()*

Method ini dipanggil ketika *activity* yang berjalan pada saat itu dihentikan sementara dan *activity* sebelumnya dijalankan kembali. (Hasil dari *method OnRestart()*).

4. *onFreeze()*

Method ini dipanggil ketika *activity* berada dalam keadaan *freeze* atau tidak merespon akibat sedang sibuk mengerjakan task tertentu pada sistem.

5. *onPause()*

Method ini dipanggil ketika *activity* di hentikan sementara dan berikutnya ketika dijalankan kembali akan berada dalam posisi aktif dan memanggil method *OnResume()*.

6. *onStop()*

Method ini dipanggil ketika *activity* tidak lagi tampak kepada *user interface*.

7. *onDestroy()*

Method ini dipanggil sebelum *activity* dihancurkan (*destroy*) oleh sistem baik secara manual maupun untuk kepentingan pelonggaran memori.

8. *onRestart()*

Method ini dipanggil ketika *activity* dijalankan kembali setelah *state stop*.

2.13 Android SDK Manager

Android sdk merupakan *tools* bagi para programmer yang ingin mengembangkan aplikasi berbasis google android. Android sdk mencakup seperangkat alat pengembangan yang komprehensif. Android sdk terdiri dari *debugger*, *libraries*, *handset emulator*, dokumentasi, contoh kode, dan *tutorial*.

Saat ini android sudah mendukung arsitektur x86 pada linux (distribusi linux apapun untuk *desktop* modern), mac os x 10.4.8 atau lebih, windows xp atau vista. persyaratan mencakup jdk, apache ant dan python 2.2 atau yang lebih baru. (IDE) yang didukung secara resmi adalah eclipse 3.2 atau lebih dengan menggunakan *plug-in android development tools* (ADT), dengan ini pengembang dapat menggunakan teks editor untuk mengedit file java dan xml serta menggunakan peralatan *command line* untuk menciptakan, membangun, melakukan *debug* aplikasi android dan pengendalian perangkat android (dengan kasus, *reboot*, menginstal paket perangkat lunak dengan jarak jauh).

2.14 Eclipse

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan disemua *platform* (*platform-independent*).

Berikut ini adalah sifat dari Eclipse:

1. *Multi-platform*

Target sistem operasi eclipse adalah microsoft windows, linux, solaris, aix, hp-ux dan mac os x.

2. *Multi-language*

Eclipse dikembangkan dengan bahasa pemrograman java, akan tetapi eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, dan lain sebagainya.

3. *Multi-role*

Selain sebagai IDE untuk pengembangan aplikasi, eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, tes perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.

2.15 Halaman Utama Eclipse

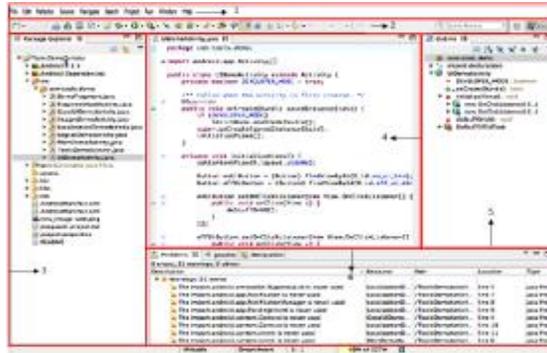
Halaman utama adalah tampilan yang pertama kali muncul ketika kita mengakses eclipse. Cara untuk mengakses eclipse pertama kali yaitu double klik pada ikon yang ada di desktop atau lihat dari daftar program. Di bawah ini adalah tampilan halaman utama dari eclipse.



Gambar 2.8 Halaman Awal Tampilan Program Eclipse

2.15.1 Jendela Utama Program Eclipse

Jendela utama merupakan awal dari pembuatan program, pembuatannya dilakukan dalam *workspace* yang didukung oleh alat lainnya. Seperti yang pernah dijelaskan dalam sebuah tulisan.



Gambar 2.9 Jendela Utama Program Eclipse

Keterangan gambar :

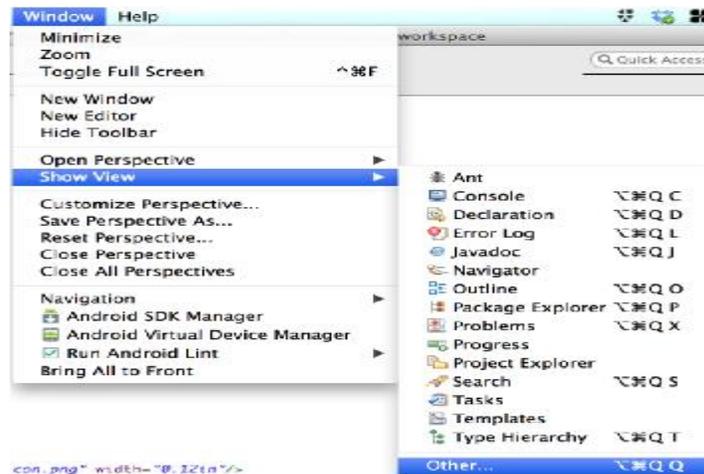
1. *Menu bar* adalah kumpulan yang terdiri atas dasar menu-menu yang digolongkan dalam suatu kategori. Misalnya menu *file* terdiri atas perintah *new*, *open*, *browse*, *save*, *save as*, *import*, *export* dan lain- lain.
2. *Tool Icon* atau *Toolbox* adalah kumpulan tools yang sering digunakan untuk melakukan eksekusi, mengecek kode, ekspor file, memodifikasi dan mengatur elemen elemen pada layar kerja.
3. *Package explorer* adalah untuk melihat semua paket kode proyek yang dibuat termasuk komponen. Dapat di tampilkan dengan cara mengklik pada segitiga kecil untuk memperluas atau meminimalkan ukuran jendela package explorer tertentu, yang akan memperluas atau meminimalkan bagian dari pemakaian kode paket yang tepat.

4. *Code editor* adalah area dimana digunakan untuk membuat perubahan kode yang dituliskan. kode warna biasanya untuk menyorot berbagai elemen sintaks dengan benar. Untuk Misalnya, definisi variabel ditampilkan dalam warna biru, dan metode pengubah ditampilkan ungu, dan untuk kode warna merah biasanya menandakan adanya kesalahan dalam penulisan sintaks.
5. *Outline* menampilkan bagian utama dari kode dalam cara yang terstruktur. dengan dapat mengklik pada segitiga untuk memperbesar dan memperkecil bagian tertentu, yang menunjukkan struktur dari dokumen xml saat ini atau *java class* yang diedit dengan editor. dan kemudian dengan cepat menavigasi ke dalam lokasi *file*. Ini adalah metode cepat dengan cara utama navigasi lokasi *file*. Pada windows dan linux, dengan menekan `ctrl + o`, dan pada mac os x, menggunakan `command + o`.
6. *Tab Problem* Tab ini muncul di bagian bawah layar dan menampilkan peringatan apapun atau kesalahan dalam proyek yang dibuat. Jika ada kesalahan dalam penulisan sintaks, maka pesan masalah akan tercantum di sini (merah). Kita juga bisa mengklik pada salah satu pesan yang muncul di sini. Ini akan membuka kode dengan kesalahan dan menempatkan kursor pada lokasi di mana ada masalah yang sama.

2.15.2 Alat dan Fitur Tambahan Eclipse

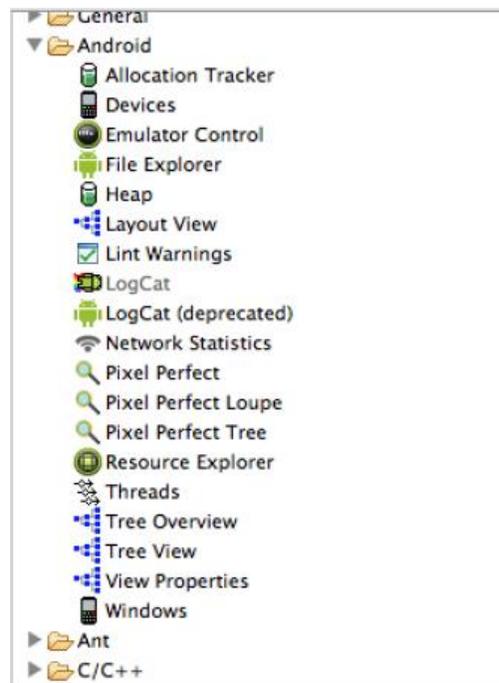
a) *Layout*

Layout standar untuk eclipse kemungkinan besar tidak akan memiliki semua komponen perlu menggunakan beberapa alat tambahan yang tersedia, serta beberapa cara yang berbeda untuk menyesuaikan apa yang ditampilkan.



Gambar 2.10 Alat dan Fitur Tambahan dalam Program Eclipse

dapat menampilkan alat android tambahan dengan memilih *window* → tampilkan *view* → lainnya → android. Ini menampilkan daftar alat di bawah bagian android, seperti yang ditunjukkan pada Gambar diatas. Untuk menambahkan salah satu dari ini untuk tata letak eclipse, pilih dari daftar. Setelah ditempatkan di layar, *drag* dan *drop* tab untuk memindahkannya ke mana pun yang diinginkan.



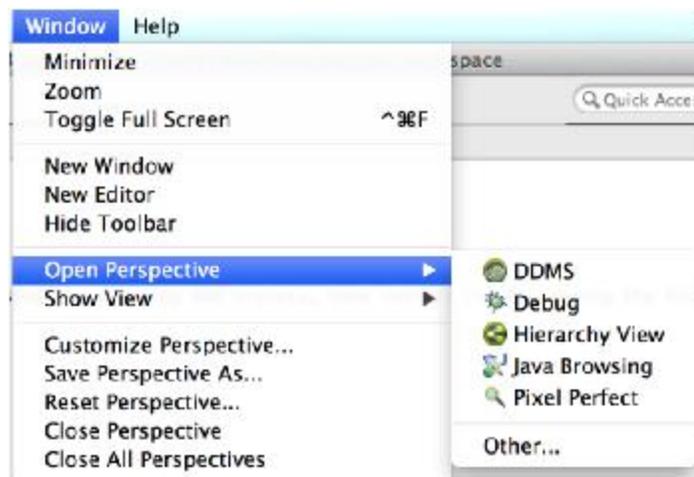
Gambar 2.11 Daftar Alat Tambahan yang ada pada Program Eclipse

b) *Perspective Menu Tools*

Konsep perspektif dalam eclipse adalah kumpulan tab dan alat-alat yang disediakan dalam cara yang telah ditetapkan. android hadir dengan perspektif yang telah dikonfigurasi diantaranya *ddms*, *hirarcy viewer*, dan *pixel perfect view*.

Langkah-langkah untuk membuka perspektif adalah:

1. Pilih *Window* → Buka *Perspective*.
2. Pilih perspektif yang ingin ditampilkan.



Gambar 2.12 Menu Membuka Perspektif dalam Eclipse

c) **Quick Button**



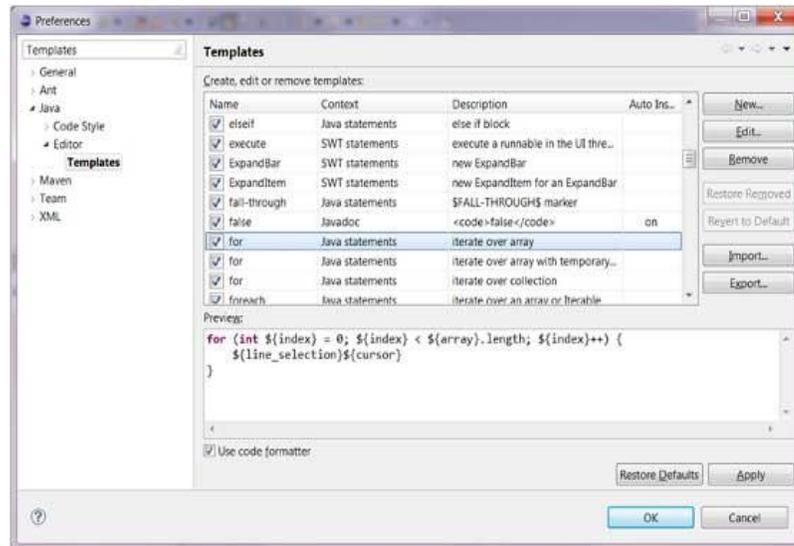
Gambar 2.13 Fitur Quick Button

Ada tombol run kecil di atas baris menu (🎮) yang mudah untuk mengeksekusi kode telah di rancang. Ini tombol jalan pintas untuk menjalankan proyek yang dibuat. Ini memiliki fungsi yang sama yang dieksekusi dengan menerapkan pilihan tombol menu *run as* (yang dapat diakses dengan mengklik kanan pada proyek aplikasi). Tombol ini adalah cara paling sederhana untuk menjalankan aplikasi, karena hanya satu klik tombol (sebagai alternatif navigasi menu). Hal ini penting untuk dicatat bahwa ini bekerja hanya jika kita mengedit *file java* maupun mengedit *file xml*, tidak ada yang terjadi ketika mengklik tombol *run*.

d) **Code Template**

Ini bagian untuk memahami cara yang tepat untuk membuat kelas android. Perlu untuk mengetahui standar penamaan yang tepat, metode yang

diperlukan, bagaimana untuk mengikat berbagai sumber daya secara bersamaan, dan dengan pola lain untuk menciptakan kelas yang tepat.



Gambar 2.14 Jendela Code Template

2.16 Flowchart

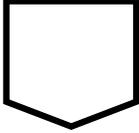
Flowchart merupakan kumpulan dari gambar yang biasanya berupa simbol atau bagian yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya.

Simbol-simbol dalam *flowchart* dibagi menjadi 3 kelompok :

a) *Flow Direction Symbols*

Dipakai untuk menggabungkan antara simbol yang satu dengan simbol lainnya.

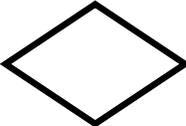
Tabel 2.1 Simbol *Flow Direction*

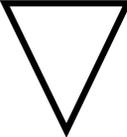
Gambar	Keterangan
	Symbol Off-line Connector (Simbol untuk keluar/masuk prosedur atau proses dalam lembar/halaman yang lain)
	Symbol Connector (simbol untuk keluar/masuk prosedur dalam lembar/halaman yang sama)

b) *Processing symbols*

Digunakan untuk menunjukkan jenis operasi pengolahan dalam suatu prosedur.

Tabel 2.2 Simbol *Processing*

Gambar	Keterangan
	Symbol Process (Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer)
	Symbol Manual Operation (Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer)
	Symbol Decision (Simbol untuk kondisi yang akan menghasilkan beberapa kemungkinan jawaban/aksi)

	<p>Symbol Predefined process (Simbol untuk mempersiapkan penyimpanan yang akan digunakan tempat pengolahan di <i>storage</i>)</p>
	<p>Symbol Off-line Storage (Simbol yang menunjukkan bahwa data di dalam simbol ini akan disimpan)</p>
	<p>Symbol Manual Input (Simbol untuk pemasukan data secara manual <i>on-line keyboard</i>)</p>
	<p>Symbol Keying Operation (Simbol operasi dengan menggunakan mesin yang mempunyai <i>keyboard</i>)</p>
	<p>Symbol Input Output (Merepresentasikan fungsi I/O yang membuat sebuah data dapat diproses (input) atau ditampilkan (output) setelah mengalami eksekusi informasi)</p>
	<p>Symbol Prosedur (Simbol ini berperan sebagai blok pembangun dari suatu program. Prosedur memiliki suatu flowchart yang berdiri sendiri diluar flowchart utama.)</p>

2.17 Penelitian Sebelumnya

Penelitian sebelumnya diperlukan dalam sebuah pembuatan skripsi karena penelitian sebelumnya dapat menjadi acuan untuk pembuatan penelitian berikutnya agar peneliti dapat menghindari adanya penelitian yang berulang-ulang namun sebenarnya sama. Berikut ini adalah beberapa tinjauan penelitian sebelumnya yang digunakan dalam perancangan *game* ini :

Penelitian (1):

Game Shopping Time

Adalah hasil karya dari Arizonanataliya Heryanti dan Galih Hermawan. *Game* ini berbasis dekstop dan software yang digunakan adalah *Adobe Flash (ActionScript 2)*. Permainan tersebut terdiri dari 3 *level* dengan perbedaan dari tiap *level* terletak pada nilai uang jajan, waktu yang diberikan dan pertanyaan yang diberikan. Permainan *shopping time* memiliki *gameplay* sebagai berikut:

- Permainan dimulai dengan layar yang menampilkan kawasan pusat perbelanjaan lengkap dengan toko-tokonya yang berjumlah 6 buah, dan dikatakan selesai jika pemain berhasil menghabiskan uang jajan yang diberikan dengan membelikan barang-barang yang ada dalam waktu yang telah ditentukan agar pemain mendapatkan *score* (nilai).
- Pemain harus menghabiskan uang tadi dengan membelikan barang-barang yang ada di toko-toko yang ada di pusat perbelanjaan.
- Saat pemain meng-klik toko, maka akan muncul kotak yang berisi nama barang dan harga barang.

- Untuk membeli barang tersebut, pemain harus bisa menjawab pertanyaan yang ada.
- Pada setiap toko hanya bisa dibeli satu barang saja.
- Permainan ini terdiri dari 3 *level* yakni:
 - a) *Level 1*, waktu yang diberikan adalah 2 menit dan uang jajan yang diberikan adalah Rp 50.000. Pemain harus menjawab pertanyaan yaitu pengurangan antara uang jajan dengan harga barang yang dibeli.
 - b) *Level 2*, waktu yang diberikan adalah 1,5 menit dan uang jajan yang diberikan adalah Rp 45.000. Pemain harus menjawab pertanyaan yaitu pengurangan antara uang jajan dengan harga barang yang dibeli dan apa warna barang yang dibeli.
 - c) *Level 3*, waktu yang diberikan adalah 1 menit dan uang jajan yang diberikan adalah Rp 37.500. Pemain harus menjawab pertanyaan yaitu pengurangan antara uang jajan dengan harga barang yang dibeli dan apa warna barang yang dibeli dan ada NPC yang akan mencuri barang-barang yang ada di toko.
- Pemain hanya dapat memainkan *level 3* jika telah berhasil melewati *level 1* dan *level 2*.
- Permainan berakhir saat uang pemain habis atau waktu yang disediakan habis.

Tampilan dari *Game Shopping Time* adalah sebagai berikut:



Gambar 2.15 Tampilan halaman utama *shopping time*



Gambar 2.16 Tampilan awal permainan *shopping time*

Penelitian (2):

“Tracing Aksara Jawa” Aplikasi Pembelajaran Mengenal dan Menulis Aksara Jawa Berbasis Android Menggunakan Corona

Adalah hasil karya dari Dewi Jayanti Setiasih, Wahyu Sulistiyo, Parsumo Raharjo. Mahasiswa Politeknik Negeri Semarang. Game ini bertemakan permainan pembelajaran aksara Jawa dikembangkan dengan menggunakan Corona, Aplikasi *Tracing* Aksara Jawa memuat pengguna aplikasi dan juga alur data pada aplikasi. Aplikasi pembelajaran aksara Jawa terdapat satu actor yaitu pengguna yang dapat mengakses keseluruhan aplikasi. Pengguna dapat memilih menu aksara Jawa, menu sandhangan dan memilih menu latihan. Pilihan

mengenal aksara Jawa dan mengenali sandhangan dilengkapi dengan contoh penggunaan aksara Jawa dan sandhangan dalam sebuah kata. Pilihan menulis aksara Jawa dan menulis sandhangan terdapat lembar latihan untuk menulis aksara Jawa dan sandhangan. Selain mengenali aksara Jawa dan sandhangan, pengguna dapat melakukan latihan *tracing* (mengikuti jejak) dengan memilih salah satu dari dua puluh benda yang telah disediakan pada menu latihan dalam bentuk kata. Setelah selesai mengerjakan latihan menulis, pengguna dapat memperoleh nilai yang dalam aplikasi ini berupa bintang dengan nilai tertinggi lima bintang dan yang terendah satu bintang. Dengan dilengkapi menu tersebut diharapkan pengguna dapat memahami aksara Jawa dengan semakin mudah dan cepat. Pembelajaran *tracing* aksara Jawa ini dilengkapi dengan cara baca yang berupa suara manusia dan latar belakang lagu anak-anak Jawa tengah yang dapat menambah pemahaman pengguna tentang aksara Jawa dan menjadikan suasana belajar menjadi lebih menyenangkan. Berikut ini screenshot penampilan program pembelajaran aksara Jawa berbasis android:



Gambar 2.17 Tampilan Halaman Utama *Tracing* Aksara Jawa



Gambar 2.18 Tampilan Menu Utama *Tracing Aksara Jawa*

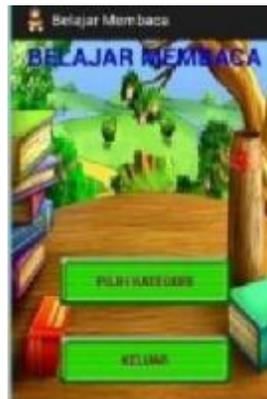
Penelitian (3):

Perancangan Permainan *Game* Edukasi Belajar Membaca Pada Anak Prasekolah Berbasis *Smartphone Android*

Aplikasi ini dirancang oleh Mahasiswa jurusan Teknik Industri Fakultas Teknologi Industri Institut Teknologi Padang dengan atas nama Busran, dan Fitriyah. Aplikasi permainan edukasi yang dirancang untuk anak prasekolah ini memiliki sejenis permainan edukasi yang unik baik dari segi penampilan, desain, dan alur permainannya sangat menarik, dalam permainan ini, dalam halaman muka sudah tersedia beberapa pilihan menu diantaranya :

- a. Judul pilihan kategori.
- b. Kategori Mengenal huruf : menampilkan tampilan halaman kategori satu mengenal huruf A sampai dengan Z secara berurutan.
- c. Kategori Mengenal kata : menampilkan tampilan halaman kategori dua mengenal kata, menyebutkan satu persatu huruf hingga membentuk satu kata.

- d. Kategori Bermain : menampilkan tampilan halaman kategori ketiga, bermain sambil belajar terdapat tiga *level*, setiap *level* memiliki nilai point jika user bermain dengan menjawab benar maka *level* berikutnya akan terbuka dan nilai point akan bertambah tetapi apabila user bermain dengan menjawab salah maka permainan akan mengunci *level* tersebut, pengguna mendapatkan nilai poin.
- e. Tombol Keluar untuk keluar dari aplikasi permainan
- Permainan ini yang bertemakan bobo salah satu karakter dalam game ini tersedia dalam beberapa *level* berdasarkan tingkat kesulitannya beberapa kategori latar tempatnya, berikut penjelasan mengenai gambaran setiap latar memiliki *level* masing masing :



Gambar 2.19 Halaman Menu Utama Belajar Membaca



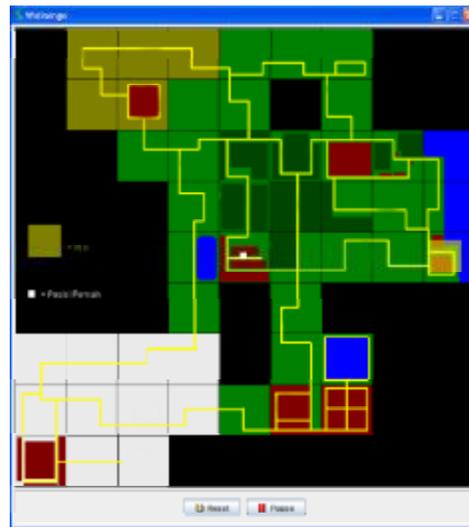
Gambar 2.20 Halaman Menu Pilihan kategori Belajar Membaca

Penelitian (4):

Membangun *Game* Edukasi Sejarah Walisongo

Adalah sebuah *game* yang dibuat oleh Nelly Indriani dan Irwan Setiawan dari Jurusan Teknik Informatika, FTIK UNIKOM. Merupakan *game* edukasi sejarah Walisongo dengan misi mendapatkan sembilan foto dari tokoh Walisongo, dan mengalahkan tiga bos musuh dalam *game*. *Storyline* dari *Game* ini adalah berlatar di sebuah pulau, yang dikisahkan bahwa akhlak umat manusia dimasa depan hancur karena kurangnya contoh-contoh yang memberikan nilai dan norma agama Islam. Untuk itu karakter Pemain kembali ke masa lalu demi mengetahui sejarah Walisongo dan menyelamatkan foto para tokoh Walisongo yang dijadikan bukti bahwa terdapat contoh yang baik untuk umat manusia di masa depan yang berasal dari tanah Jawa. Namun perjalanan player untuk mengetahui sejarah Walisongo dan mendapatkan foto tokoh Walisongo tidak mudah karena dihalangi oleh Zombie yang menyesatkan umat manusia di masa depan. Maka tugas pemain dalam *game* ini adalah mengumpulkan semua foto Walisongo, dan mengalahkan 3

(tiga) bos *zombie*. Misi dalam *game* edukasi sejarah Walisongo ini adalah mendapatkan Sembilan foto dari tokoh Walisongo, dan mengalahkan tiga bos *zombie* atau musuh dalam *game*. Karakter pemain memulai permainan dengan darah 100% dan *score* 0. *Score* dapat bertambah apabila pemain mendapatkan foto dan mengalahkan *zombie*. Tipe *game* berupa Petualangan dimana karakter pemain dapat digerakan oleh pemain untuk bergerak ke kiri, bergerak ke atas, bergerak ke bawah dan menembak. Tampilan dari *game* Walisongo adalah sebagai berikut:



Gambar 2.21 Tampilan Denah Permainan Walisongo

Penelitian (5):

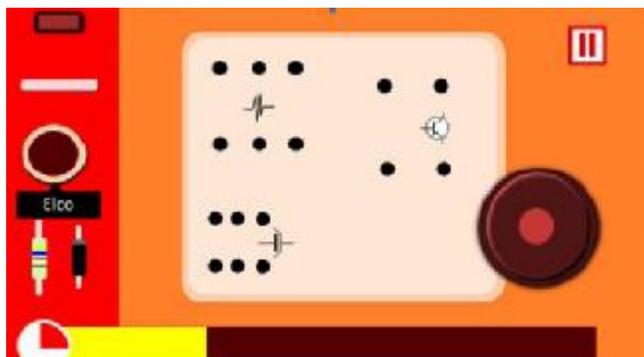
***Game* Edukasi Perakitan *Amplifier* Berbasis Android Untuk User Umum**

Adalah sebuah aplikasi *game* Edukasi Perakitan *Amplifier* Berbasis Android Untuk User Umum. *Game* ini dibuat oleh Mahasiswa Fakultas Teknik Informatika Universitas Muria Kudus yang bernama Fundhi Fanju Hafili, Rizky Sari Mei Maharani, dan anastasya latubessy, dibuat menggunakan *Macromedia*

Flash. Di dalam *Game* yang disediakan mempunyai 2 *level*. Dari pertama *level* yang lebih mudah dan dilanjutkan dengan *level* 2 dengan posisi yang sedikit lebih lanjut dari posisi pertama. *Level* pertama dibuat *game* petualangan untuk menyusun komponen *amplifier*. Sehingga tercapai semua komponen *amplifier* sampai menyelesaikan *level* satu. Nilai dari *level* satu berdasarkan pengumpulan komponen *amplifier*.



Gambar 2.22 Halaman *Level* Petualang Perakitan *Amplifier*



Gambar 2.23 Menyusun Simulasi *Amplifier* seperti Aslinya

Dari kelima penelitian terdahulu di atas, maka ditarik beberapa kesimpulan yang bisa dijadikan sebagai pertimbangan untuk perkembangan penelitian selanjutnya:

1. Dari kelima *game* yang ada, belum ada permainan yang bertema budi pekerti yang bersifat pengujian kemampuan menganalisa. Penyimpanan soal permainan diperlukan agar jika pemain tidak bisa menyelesaikan *game* maka permainan akan berakhir.
2. Beberapa *game* edukasi lebih memperkenalkan tentang sejarah dan materi-materi di sekolah, dan jarang memperkenalkan tentang test kemampuan mengenal rambu rambu lalu lintas di Indonesia.
3. Beberapa *game* di atas visualnya masih terlihat cukup dan menarik. sedangkan desain visual dari sebuah *game* sangat menentukan apakah pemain tertarik ingin memainkan *game* tersebut atau tidak.