

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Membandingkan penelitian yang sudah ada dengan penelitian yang dilakukan Penelitian yang sudah ada:

2.1.1 Aplikasi *Cartoon Alphabet*

Aplikasi *Cartoon Alphabet* adalah aplikasi yang sudah di buat oleh programmer (di-link playstore/ sumber) yang sudah di upload di *playstore market* android aplikasi ini tentang pembelajaran tentang mengenal huruf besar dan kecil disertai gambar dan nama objek gambar sekaligus mengajarkan kepada pengguna untuk bisa membaca. Aplikasi tersedia dua menu *Learn* dan *Practice* dapat di lihat pada Gambar 2.1. Menu pertama yaitu *Learn* berisikan contoh huruf besar dan huruf kecil serta gambar dan nama gambar yang berawalan sesuai huruf depan sebagai contoh dapat di lihat pada Gambar 2.2. Menu ke dua yaitu *Practice* menu berisikan cara menulis huruf *alphabet* dari huruf A sampai dengan Z, untuk melihat tampilan menu ini dapat dilihat Gambar 2.. Kelemahan Aplikasi *Cartoon Alphabet* adalah dalam menu *learn* hanya menunjukkan cara membuat huruf besar tidak menampilkan cara membuat huruf kecil, jadi belum lengkap dalam pengenalan huruf dalam aplikasi ini.



Gambar 2.1 Tampilan Depan *Cartoon Alphabet*

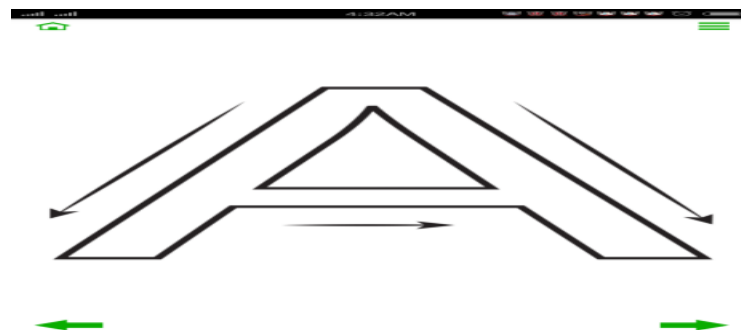
a. Tampilan Aplikasi *Cartoon Alphabet Menu Learn*

Di bawah ini adalah tampilan dari isi menu *Learn* berisikan contoh huruf besar dan huruf kecil serta gambar dan nama gambar yang berawalan sesuai huruf depan, di menu ini diharapkan pengguna dapat mengetahui contoh huruf *alphabet* dari A sampai Z dan disertai dengan bacaan yang berawalan dari huruf yang dipilhkan untuk memudahkan pengguna disertai contoh gambar supaya memudahkan pengguna membaca sebagai contoh dapat dilihat pada Gambar 2.2 Menunjukkan contoh huruf A besar dan huruf A kecil, disertai contoh terusan kata yang berawalan huruf A di Gambar 2.2 yaitu huruf diteruskan menjadi “Aladdin” untuk mempermudah dan untuk menarik pengguna ditambah lagi gambar yang menunjukkan karakter yang bersangkutan dengan huruf awal sebagai contoh ditampilkan ini disertakan gambar “Aladdin”.



Gambar 2.2 Tampilan Menu *Learn*

b. Tampilan Aplikasi *Cartoon Alphabet Menu Practice*



Gambar 2.3 Tampilan Menu *Practice*

Tampilan Menu Practice diatas adalah berisi cara pembuatan huruf alphabet dari A sampai dengan Z gambar 2.3 sebagai contoh huruf A terdapat anak panah menunjukan cara pembuatan huruf A besar, ini bertujuan memberikan gambaran kepada pengguna untuk memulai membuat huruf A, untuk lebih jelasnya dapat dilihat pada Gambar 2.3

2.1.2 Aplikasi Edukasi Anak Mengenal Huruf

Aplikasi edukasi anak mengenal huruf adalah aplikasi pembelajaran yang dibuat dan di-*upload* di *playstore* yang ditujukan kepada pengguna di kalangan anak-anak usia di bawah lima tahun untuk belajar mengenal huruf, mengingat dan berlatih untuk meningkatkan kinerja otak kiri di usia dini. Aplikasi ini terdiri dari tiga menu utama yaitu menu belajar, mengingat dan berlatih, Gambar 2.4 menunjukkan tampilan utama dari aplikasi.



Gambar 2.4 Tampilan Utama Aplikasi Edukasi Anak Mengenal Huruf

a. Tampilan Menu Belajar

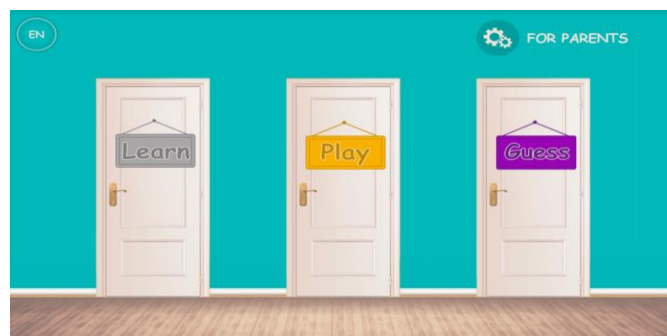
Aplikasi menu belajar ini berisikan huruf alphabet dari A sampai dengan Z dengan kombinasi huruf besar dan kecil sehingga diharapkan pengguna dapat mudah menghafal dan mengetahui huruf alphabet, untuk memudahkan pengguna dalam menghafal dalam menu ini huruf dikombinasikan warna-warna yang berbeda agar dapat membantu mempermudah dalam penghafalan. Tampilan menu belajar dapat dilihat di Gambar 2.5



Gambar 2.5 Tampilan Menu Belajar

2.1.3 Aplikasi *Learn Color*

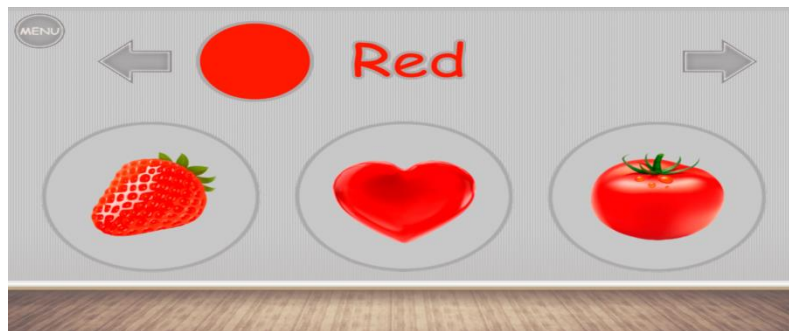
Aplikasi *learn color* ini adalah aplikasi jenis pembelajaran yang sudah dibuat dan di upload di *playstore* atau *android market* oleh aplikasi bersifat *free* yang dipergunakan untuk pembelajaran anak usia balita atau di bawah lima tahun aplikasi ini terdiri tiga menu utama yaitu *learn*, *play* dan *guess*, dalam menu *learn* berisikan tentang pengenalan warna dan diberikan contoh objek sesuai warna misalkan warna merah dan juga gambar objeknya juga gambar-gambar yang berwarna merah. Menu *play* adalah pencocokan warna misalkan ada warna merah pengguna dihadapkan dengan pilihan warna jadi pengguna harus memilih warna yang sama, sedangkan menu *guess* berisikan mirip dengan menu *play* karena masih menggunakan pencocokan warna. Untuk menunjukkan Aplikasi *learn color* ini pada Gambar 2.6



Gambar 2.6 Tampilan utama Aplikasi *learn color*

a. Tampilan Menu Learn

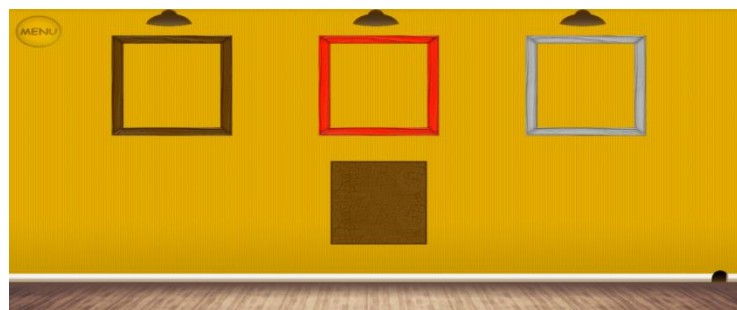
Tampilan menu *learn* ini berisikan tiga pembelajaran dalam pencocokan warna yaitu *read, orange, yellow, green, blue, violet, brown, gray, dan pink*, dalam menu *learn* ini pembuat aplikasi menambahkan suara berdasarkan warna yang ada dalam aplikasi ini. Aplikasi ini dipergunakan untuk mempermudah pengguna mencocokkan warna sebagai contoh pada Gambar tampilan red yang berarti warna merah ditambahkan contoh tiga objek gambar yang berdasarkan warna tersebut seperti contoh Gambar 2.7. Untuk lebih jelasnya dapat dilihat pada Gambar 2.7 gambar *red*.



Gambar 2.7 Contoh Menu Learn

b. Tampilan Menu Play

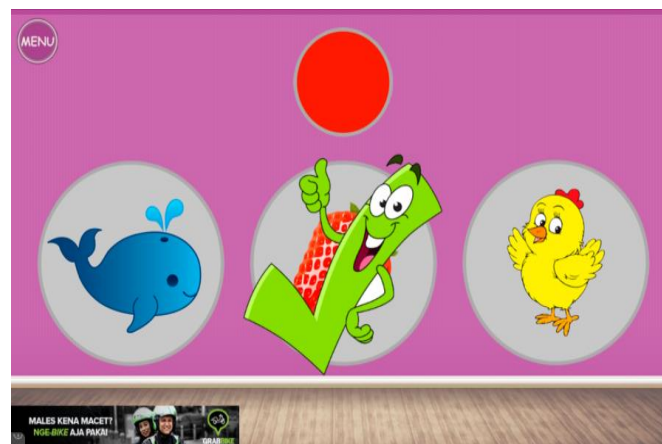
Didalam menu *play* ini pengunjung diharapkan dapat mencocokkan warna sesuai gambar yang tampil seperti contoh pada Gambar di bawah ini. Dalam aplikasi ini disayangkan tidak bisa berjalan sebagai mana mestinya untuk melihat tampilan menu *play* pada Gambar 2.8 di bawah ini.



Gambar 2.8 Tampilan Menu Play

c. Tampilan Menu *Gues*

Menu *gues* ini sebenarnya juga sama dengan menu *play* dalam menu ini pengguna dituntun untuk memilih warna sesuai soal warna seperti Gambar 2.9 di bawah ini misal warna soal muncul merah maka ada pilihan gambar yang berwarna sama dengan soal jika menjawab benar akan keluar gambar centang seperti gambar di bawah ini.



Gambar 2.9 Tampilan Menu *Gues*

2.1.4 Penelitian yang Sudah dilakukan

Penelitian ini sudah dilakukan dengan mengambil tiga contoh aplikasi pembelajaran dengan men-*download* dan menggunakan aplikasi yang diperoleh dari *playstore* atau android market, dari ketiga aplikasi yang menjadi bahan contoh dalam penelitian ini ditemukan kelemahan dan kelebihan dari ketiga aplikasi tersebut :

1. Aplikasi *Cartoon Alphabet*

Aplikasi *Cartoon Alphabet* yang sudah dibuat oleh programmer (di link *playstore/ sumber*) yang sudah di upload di *playstore market* android aplikasi ini tentang pembelajaran tentang mengenal huruf besar dan kecil disertai gambar dan nama objek gambar sekaligus mengajarkan kepada pengguna untuk bisa membaca. Aplikasi hanya tersedia dua menu *Learn* dan *Practice*. Menu pertama yaitu *Learn* berisikan contoh huruf besar dan huruf kecil serta gambar dan nama gambar yang berawalan sesuai huruf depan. Menu ke dua yaitu *Practice* menu berisikan cara menulis huruf *alphabet* dari huruf A sampai dengan Z, Kelamahan Aplikasi *Cartoon Alphabet* adalah dalam menu *learn* hanya menunjukkan cara

membuat huruf besar tidak menampilkan cara membuat huruf kecil, jadi belum lengkap dalam pengenalan huruf dalam aplikasi ini.

2. Aplikasi Edukasi Anak Mengenal Huruf

Aplikasi edukasi anak mengenal huruf adalah aplikasi pembelajaran yang dibuat dan di *upload* di *playstore* oleh yang ditujukan kepada pengguna di kalangan anak-anak usia di bawah lima tahun untuk belajar mengenal huruf, mengingat dan berlatih untuk meningkatkan kinerja otak kiri di usia dini. Aplikasi ini terdiri dari tiga menu utama yaitu menu belajar, mengingat dan berlatih. Dalam aplikasi ini dinilai masih banyak kekurangannya seperti hanya ada single menu di tampilan utama, dan tidak ada *background* untuk pengguna dapat membuat bosan memakai aplikasi dikarenakan penggunaannya ditujukan kepada anak-anak usia dini.

3. Aplikasi *Learn Color*

Aplikasi *learn color* ini adalah aplikasi jenis pembelajaran dipergunakan untuk pembelajaran anak usia balita atau di bawah lima tahun. Aplikasi ini terdiri tiga menu utama yaitu *learn*, *play* dan *guess*. Dalam menu *learn* berisikan tentang pengenalan warna dan diberikan contoh objek sesuai warna misalkan warna merah dan juga gambar objeknya juga gambar-gambar yang berwarna merah. Menu *play* adalah pencocokan warna misalkan ada warna merah pengguna dihadapkan dengan pilihan warna jadi pengguna harus memilih warna yang sama, sedangkan menu *guess* berisikan mirip dengan menu *play* karena masih menggunakan pencocokan warna. Kelemahan aplikasi terdapat setiap menu karena dari ketiga menu yang ada pada dasarnya sama hanya merubah objek gambar secara prinsip sama.

Dari ketiga contoh aplikasi di atas yang bersumber aplikasi di *playstore* menunjukkan beberapa kelemahan, dengan kasus di atas maka penelitian ini membuat aplikasi yang serupa untuk diterapkan ke usia dini untuk TK IDHATA. Aplikasi diberi nama aplikasi pembelajaran untuk TK IDHATA aplikasi ini secara prinsip sama dengan contoh aplikasi di atas, dengan beberapa pengembangan agar lebih menarik lagi seperti berikut:

Dalam menu utama atau tampilan utama aplikasi pembelajaran ini menyediakan tiga menu utama yaitu menu bhs.inggris, mewarnai, suara hewan, dan bangunan serta difasilitasi menu keluar aplikasi, aplikasi juga didukung *backsound* supaya membuat pengguna tidak merasakan jenuh dan membuat imajinasi pengguna lebih bervariasi, dan juga ditambahkan gambar-gambar tokoh kartun yang lucu khas anak-anak usia dini.

Dari perbandingan antara ketiga contoh aplikasi yang sudah dianalisis dengan aplikasi yang dibuat aplikasi yang telah dibuat lebih banyak variasi dari sisi manfaat dan kegunaan aplikasi pembelajaran untuk TK idhata ini ditujukan untuk anak usia dini untuk belajar mengenal warna, jenis hewan, jenis bangunan, suara hewan, aplikasi ini sangat membantu belajar, tidak menutup kemungkinan aplikasi ini masih kurang dari segi grafis.

2.2 Pengertian Aplikasi

Menurut Abdul Kadir (2003), aplikasi adalah suatu program yang dibuat oleh pemakai yang ditujukan untuk melakukan suatu tugas khusus. Berdasarkan definisi diatas dapat disimpulkan bahwa aplikasi adalah program yang dibuat untuk melakukan tugas khusus dalam perusahaan. Menurut Supriyanto (2005), aplikasi adalah program yang memiliki aktivitas pemrosesan perintah yang diperlukan untuk melaksanakan permintaan pengguna dengan tujuan tertentu.

2.3 Pembelajaran

Pembelajaran pada dasarnya adalah suatu proses komunikasi yang melibatkan guru sebagai sumber informasi, pesan ktif dan efisien. Media pembelajaran ini diperuntukan agar siswa dapat menerima dapat menangkap pembelajaran atau yang di kenal sebagai materi pelajaran. Media pembelajaran adalah segala sesuatu yang dapat digunakan agar proses pembelajarannya berjalan secara efesecara benar dan utuh segala informasi yang disampaikan sebagai pesan pembelajaran. Ketika teknologi khususnya teknolgi informasi belum berkembang seperti ini ketika ilmu pengetahuan belum sepesat ini, proses pembelajaran biasanya berlangsung pada tempat dan waktu tertentu. Proses pembelajaran adalah proses komunikasi antara guru dan siswa bahasa verbal sebagai media utama

penyampaian materi pelajaran. Proses pembelajaran sangat tergantung pada guru sebagai sumber belajar Wina Sanjaya (2012).

2.3.1 Faktor-Faktor yang Berpengaruh Terhadap Sistem Pembelajaran

Menurut Wina Sanjaya (2012), terdapat beberapa faktor yang dapat mempengaruhi kegiatan proses sistem pembelajaran di antaranya faktor guru, faktor siswa, sarana, alat dan media yang tersedia, serta faktor lingkungan.

a. Faktor Guru

Guru adalah komponen yang berberpengaruh dalam suatu proses pembelajaran. Media dalam proses pembelajaran, akan dipengaruhi oleh guru itu sendiri tentang hakikat pembelajaran. Guru yang menganggap hanya sebatas menyampaikan materi pelajaran, akan berbeda dengan guru yang menganggap mengajar adalah suatu pemberian bantuan kepada peserta didik.

b. Faktor Siswa

Siswa adalah organisme yang unik yang berkembang sesuai dengan tahap perkembangannya. Perkembangan anak adalah perkembangan seluruh aspek kehidupan, akan tetapi tempo dan irama perkembangan masing-masing anak pada setiap aspek tidak selalu sama. Proses pembelajaran dapat dipengaruhi oleh perkembangan anak yang tidak selalu sama. Oleh sebab itu sistem pembelajaran yang dikembangkan guru dan media yang dapat dimanfaatkan oleh guru, akan sangat tergantung pada aspek perkembangan siswa itu sendiri.

c. Faktor Sarana dan Prasarana

Sarana adalah segala sesuatu yang mendukung secara langsung terhadap kelancaran proses pembelajaran, misalnya media pembelajaran alat-alat pelajaran, perlengkapan sekolah dan lain sebagainya, sedangkan prasarana adalah segala sesuatu yang secara tidak langsung dapat mendukung keberhasilan proses pembelajarannya misalnya, jalan menuju sekolah, penerangan sekolah, kamar kecil dan lain sebagainya. Kelengkapan sarana dan prasarana akan membantu guru dalam proses pembelajarannya.

d. Faktor Lingkungan

Dilihat dari dimensi lingkungan ada dua faktor yang dapat mempengaruhi proses pembelajaran, yaitu faktor organisasi kelas dan faktor iklim sosial-psikologis.

Faktor organisasi kelas yang di dalamnya meliputi jumlah siswa dalam satu kelas merupakan aspek penting yang dapat mempengaruhi proses pembelajaran. Organisasi kelas yang terlalu besar akan kurang efektif untuk mencapai tujuan pembelajaran. Kelompok belajar yang besar dalam satu kelas berkecenderungan terhadap siswa misalnya kepuasan belajar setiap siswa akan cenderung menurun hal ini disebabkan terbatasnya pelayanan setiap guru terhadap siswa karena jumlah kelompok belajar yang terlalu dalam jumlah banyak.

Iklim sosial-psikologis secara internal adalah hubungan antara orang yang terlibat dalam lingkungan sekolah misalnya, iklim sosial antara siswa dengan siswa. Antara guru dengan guru, bahkan antara guru dengan pimpinan sekolah.

2.4 Multimedia

Menurut Wina Sanjaya (2012), multimedia merupakan salah satu jenis media audiovisual yang memiliki karakteristik tersendiri, yakni penggabungan dari beberapa media yang dapat disatukan dalam komputer. Multimedia merupakan ragam pengemasan materi pembelajaran dengan memadukan berbagai ragam media untuk dipelajari siswa sesuai dengan tujuan yang akan dicapai. Terdapat berbagai macam media yang dapat dipadukan di antaranya teks, suara, gambar atau foto, film (video), animasi. Multimedia digunakan sebagai alat bantu mengajar, yang penggunaannya sangat ditentukan dan tergantung pada guru dalam proses pembelajaran dalam kelas, pada saat mengajar guru menggunakan berbagai jenis media seperti audio, video slide, gambar, foto dan sebagainya.

2.4.1 Bentuk Multimedia

a. Multimedia Linear

Menurut Wina Sanjaya (2012), multimedia linear adalah multimedia yang bersifat sekuensial atau berurutan, setiap siswa atau pemakai multimedia ini menggunakannya sesuai dengan urutan setahap demi setahap sesuai dengan

pengemasan materi yang ditentukan. Siswa belajar berdasarkan bagian-bagian yang didesain sedemikian rupa secara berurutan dengan waktu yang telah ditentukan. Bentuk multimedia yang bersifat linear memiliki kelebihan di antaranya sebagai berikut:

1. Lebih mudah dalam pengembangannya. Hal ini disebabkan multimedia yang bersifat linier bentuknya lebih sederhana yang tidak banyak menggunakan fungsi kontrol.
2. Multimedia ini mudah digunakan siswa karena siswa tidak dihadapkan pada berbagai frame dan menu pilihan.
3. Multimedia linier terdiri atas bagian-bagian atau unit-unit terkecil bahan pelajaran, dengan demikian lebih mudah dalam kontrol penguasaan materi oleh siswa.
4. Bentuk umpan balik dapat dilakukan sehingga segera pula siswa dapat memperbaiki apabila diperlukan.

b. Multimedia Interaktif

Multimedia interaktif adalah multimedia yang tidak bersifat linier, namun siswa memiliki pilihan sesuai dengan menu yang ditawarkan. (Wina Sanjaya, 2012). Dalam mempelajari satu topik bahasan siswa dapat memilih mana yang akan dipelajari lebih dahulu. Dengan demikian ciri khas dari multimedia interaktif adalah semacam pengontrol yang biasa disebut *graphical user interface (GUI)*, yang bisa, *scroll* berupa *icon*, *button* atau yang lainnya. Setiap *GUI* tersebut dapat dioperasikan oleh siswa (pemakai) untuk mencari informasi yang diinginkan.

Beberapa manfaat penggunaan multimedia interaktif diantaranya:

1. Multimedia interaktif sifatnya lebih dinamis sehingga tidak membosankan.
2. Multimedia interaktif memberikan pilihan menu yang lebih beragam sehingga siswa lebih menyukai sebagai pemakai media ini.
3. Kajian materi pelajaran yang lebih lengkap memungkinkan multimedia interaktif lebih memiliki keanekaragaman materi yang dapat dipahami siswa

2.5 AdobeFlash

Menurut Heni A. Puspitosari (2010), *Adobe Flash CS5* merupakan program animasi vector yang telah banyak digunakan oleh animator untuk membuat animasi, salah satunya adalah untuk membuat presentasi multimedia. Di dalam pembuatan slide slide presentasi tersebut, bisa disisipkan gambar, sound, text, maupun video. Terdapat program aplikasi yang dapat digunakan untuk membuat presentasi multimedia, dan salah satunya adalah *AdobeFlash CS5* dalam pembuatan presentasi multimedia, antara lain:

- a) Memiliki kemampuan untuk membuat presentasi yang atraktif baik presentasi otomatis atau presentasi interaktif.
- b) Kemudahan dalam melakukan penyisipan unsur multimedia seperti sound, gambar ataupun video
- c) Program mudah dijalankan.

2.6 CorelDRAW

Menurut Agnessia Atitita (2011), Program *CorelDraw* adalah program aplikasi desain grafis yang berguna untuk membuat desain vektor, logo, dan layout halaman, Hampir setiap tahun Program *CorelDraw* memperbarui versi dan fasilitas di dalamnya sehingga mempermudah dalam membuat desain desainya. Program *CorelDraw versi X5* memiliki banyak kelebihan dibanding *corelDraw* sebelumnya. *Coreldraw X5* ini membawa beberapa fitur antara lain ;

1. Video Tutorial

CorelDRAW X5 menyediakan video tutorial yang berguna untuk mempelajari program *CorelDRAW X5* dengan cepat.

2. Tooltips

Ketika posisi pointer berada di atas ikon atau tombol, maka tool tip akan muncul menampilkan informasi tentang nama, fungsi dan cara penggunaannya.

3. *Corel PowerTRACE X5*

Dengan perintah *PowerTrace*, user dapat mengkonversi objek bitmap menjadi objek vektor. Objek vektor yang dihasilkan dapat diedit lebih halus dan akurat dibanding versi program *CorelDRAW* sebelumnya.

4. *Drawing Tools*

Ada beberapa *Drawing tool* atau alat gambar baru di dalam *CorelDRAW X5* seperti: *B-Spline tool*, skala mata panah, *Connector tool* dan *Dimension tool*.

5. *Mesh Fill tool*

Mesh Fill tool digunakan untuk mewarnai objek dengan berbagai campuran warna. Dengan tombol *Smooth mesh color* yang baru pada *property bar*, user dapat menghasilkan transisi warna yang lebih halus. Selain itu, jumlah node per mesh telah berkurang sehingga lebih mudah untuk memanipulasi objek.

6. *Bounding Box*

Ketika menggambar garis atau kurva dengan *curve tool*, user dapat menampilkan atau menyembunyikan *bounding box*, sehingga user lebih mudah mengatur bentuk dan arah garis tanpa terganggu oleh *bounding box*.

7. *Round corners*

Saat membuat kotak dengan *Rectangle tool*, user dapat mengatur sudut kotak dengan bentuk *chamfer*, *scallop*, atau *round corners* dari *property bar*.

8. *Lock Toolbar*

Tampilan *Toolbar* dapat dikunci diposisi tertentu sehingga tidak mungkin tergeser saat user mengklik tool. Jika user ingin memindah *toolbar*, user dapat membuka kuncinya dan mengatur posisi *toolbar* pada layar.

9. Kotak dialog *Export for Web*

Kotak dialog *Export for Web* dapat digunakan untuk mengekspor desain dari program *CorelDRAW* ke tampilan web dengan perintah *File Export for Web*.

2.7 Adobe Photoshop

Adobe Photoshop merupakan perangkat lunak pengolahan gambar. Photoshop menciptakan gambar yang mudah diakses ke data file, yakni photoshop CS4 memperlancar design dan mengolah foto berkualitas. Fitur yang paling menonjol dari tampilan *interface photoshop CS4* adalah penempatan dokumen dalam bentuk tab (*tabbed document*). Berbeda dengan photoshop sebelumnya, dokumen atau file image ditampilkan dalam jendela-jendela kanvas tersendiri sehingga tampilan jendela sering kali tampak berantakan ketika membuka file dalam jumlah yang cukup banyak. *Photoshop CS4* memiliki keunggulan yang memperlancar alur pekerjaan dan

Dan menghindari kekacauan tampilan jendela. Setiap dokumen atau file image yang dibuka akan ditempatkan secara berdampingan, dengan begitu peralihan dari dokumen yang satu ke dokumen yang lain menjadi lebih mudah dan cepat (Jubilee Enterprise, 2009).

2.8 Analisis dan Perancangan Sistem

2.8.1 Analisis Sistem

Analisis Sistem adalah penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mendefinisikan dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga diusulkan perbaikan-perbaikan. Analisis sistem bertujuan untuk mengidentifikasi permasalahan-permasalahan yang ada pada sistem di mana aplikasi dibangun yang meliputi perangkat keras (*hardware*), perangkat lunak (*software*) dan pengguna (*User*). Analisis ini diperlukan sebagai dasar bagi tahapan perancangan sistem. Analisis sistem meliputi spesifikasi aplikasi, spesifikasi pengguna, dan lingkungan operasi (Cristian Putri : 2003).

2.8.2 Perancangan Sistem

Setelah analisis sistem langkah selanjutnya adalah perancangan sistem. Perancangan sistem secara umum didefinisikan sebagai pengidentifikasian komponen-komponen sistem informasi dengan tujuan untuk dikomunikasikan dengan pemakai (Hans-Erik Eriksson : 2004) .

Object Oriented Program(OOP) merupakan paradigma baru dalam rekayasa *software* yang didasarkan pada obyek dan kelas (Hans-Erik Eriksson: 2004). Diakui para ahli bahwa *object - oriented* merupakan metodologi terbaik yang ada saat ini dalam rekayasa *software*. *Object-oriented* memandang *software* bagian perbagian dan menggambarkan satu bagian tersebut dalam satu obyek.

Teknologi obyek menganalogikan sistem aplikasi seperti kehidupan nyata yang didominasi oleh obyek, dengan demikian keunggulan teknologi obyek adalah bahwa model yang dibuat akan sangat mendekati dunia nyata yang masalahnya akan dipecahkan oleh sistem yang dibangun. Model obyek, *atribut* dan perlakuannya bisa langsung diambil dari obyek yang ada di dunia nyata.

Unified Modeling Language (UML) merupakan standar yang relatif terbuka yang dikendalikan oleh OMG (*Object Menegement Group*), sebuah konsorium terbuka yang terdiri dari banyak perusahaan. OMG dibentuk untuk membuat standar-standar yang mendukung interoperabilitas sistem berorientasi objek. Menurut (Russ Milles & Kim Hamilton : 2006), UML terdiri 7 buah diagram, adapun diagram yang sering digunakan dalam pemodelan, yaitu *Usecase Diagram, Class Diagram, Sequence Diagram, Activity Diagram, State Diagram, Componen Diagram, Deploymen Diagram*, dalam pembuatan UML diatas digunakan Program Aplikasi *Rational Rose*, berikut pembahasannya




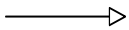



2.8.2.1 Use Case Diagram

Use case diagram menggambarkan sejumlah *external actors* dan hubungannya ke *use case* yang diberikan oleh sistem. Use case adalah deskripsi fungsi yang disediakan oleh sistem dalam bentuk teks sebagai dokumentasi dari *use case symbol* namun dapat juga dilakukan dalam *activity diagrams*. Use case

digambarkan hanya di lihat dari luar oleh *actor* (keadaan lingkungan sistem yang di lihat user) dan bukan bagaimana fungsi yang ada dalam sistem (Dana S. Kusumo 2004:3).

Simbol-simbol yang digunakan pada *use case diagram* disajikan pada Tabel 2.1.

Tabel 2.1. Tabel simbol *use case diagram*

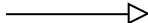



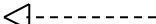
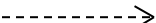
NO	GAMBAR	NAMA	KETERANGAN
1.		<i>Aktor</i>	Idealization orang eksternal, proses, atau hal yang berinteraksi dengan sistem, subsistem atau kelas.
2.		<i>Use case</i>	Sebuah <i>use case</i> menggambarkan interaksi dengan <i>actor</i> sebagai urutan pesan antarsistem dan aktor satu atau lebih.
3.		<i>System Boundary</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
4.		<i>Generalization</i>	Hubungan antara <i>use case</i> umum dan <i>use case</i> yang lebih spesifik yang mewarisi dan menambahkan fitur.
5.		<i>Comunication Association</i>	Jalur komunikasi antara <i>actor</i> dan <i>use case</i> yang berpartisipasi didalam.
6.		<i>Extend</i>	Penyisipan perilaku tambahan ke dalam basis <i>use case</i> yang tidak tahu tentang hal itu.
7.		<i>include</i>	Penyisipan perilaku tambahan ke dalam basis <i>use case</i> yang secara eksplisit menggambarkan penyisipan.

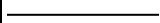
2.8.2.2 Class Diagram

Class diagram menggambarkan struktur statis *class* di dalam sistem. *Class* merepresentasikan sesuatu yang di tangani oleh sistem. *Class* dapat berhubungan dengan yang lain melalui berbagai cara: *associated* (terhubung satu sama lain), *dependent* (satu *class* tergantung/menggunakan *class* yang lain), *specialized* (satu *class* merupakan spesialisasi dari *class* lainnya), atau *package* (grup bersama sebagai satu unit). Sebuah sistem biasanya mempunyai beberapa *class diagram* (Dana S. Kusumo 2004:3).

simbol-simbol yang digunakan pada *class diagram* disajikan pada Tabel 2.2.

Tabel 2.2. Tabel simbol *class diagram*


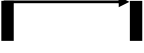

NO	GAMBAR	NAMA	KETERANGAN
1.		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagai perilaku dan struktur data dari objek yang ada di atas objek induk (<i>ancestor</i>).
2.		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari dua objek.
3.		<i>Class</i>	Himpunan dari objek-objek yang terbagi atribut serta operasi yang sama.
4.		<i>Collaboration</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
5.		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6.		<i>Dependency</i>	Hubungan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung

			pada elemen yan tidak mandiri.
7.		<i>Association</i>	Untuk menghubungkan objek satu dengan objek yang lainnya.

2.8.2.3 Sequence Diagram

Sequence diagram menggambarkan kolaborasi dinamis antara sejumlah objek. Kegunaan untuk menunjukkan rangkaian pesan yang di kirim Antara objek juga interaksi antara objek, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem (Dana S. Kusumo 2004:4). Simbol-simbol yang digunakan pada *sequence diagram* disajikan pada Tabel 2.3.

Tabel 2.3. Tabel simbol *sequence diagram*

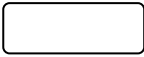


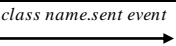
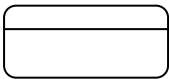
NO	GAMBAR	NAMA	KETERANGAN
1.		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2.		<i>Message</i>	<i>Message</i> ditampilkan sebagai anak panah dari lifeline dari satu objek ke objek yang lain.
3.		<i>Return</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.

2.8.2.4 Statechart Diagram

Statechart diagram menggambarkan semua *state* (kondisi) yang di miliki oleh suatu *object* dari suatu *class* dan keadaan yang menyebabkan *state* berubah. Kejadian dapat berupa *object* lain yang mengirim pesan. *State class* tidak digambarkan untuk semua *class*, hanya yang mempunyai sejumlah *state* yang terdefini dengan baik dan kondisi *class* berubah oleh *state* yang berbeda (Dana S.

Kusumo 2004:4). Simbol-simbol yang digunakan pada *statechart diagram* disajikan pada Tabel 2.4.

Tabel 2.4. Tabel simbol *statechart diagram*

NO	GAMBAR	NAMA	KETERANGAN
1.		<i>State</i>	Meliputi seluruh pesan dari <i>object</i> yang dapat mengirim dan menerima.
2.		<i>Start state</i>	Masing-masing diagram harus mempunyai satu dan hanya satu <i>start state</i>
3.		<i>Stop State</i>	Sebuah <i>object</i> boleh mempunyai banyak <i>stop state</i>
5.		<i>State</i> <i>Transition</i>	Sebuah kejadian yang memicu sebuah <i>state object</i> dengan cara memperbarui satu atau lebih nilai atributnya.
6.		<i>State Detail</i>	<i>Action-action</i> yang mengiringi seluruh <i>state transition</i> ke sebuah <i>state</i> .

2.8.2.5 Activity Diagram

Activity diagram menggambarkan rangkaian aliran aktifitas, digunakan untuk mendeskripsikan aktifitas yang di bentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi (Dana S. Kusumo 2004:4). Simbol-simbol yang digunakan pada *use activity diagram* disajikan pada Tabel 2.5.

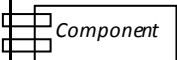
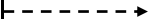
Tabel 2.5. Tabel simbol *activity diagram*

NO	GAMBAR	NAMA	KETERANGAN
1.		<i>Activity State</i>	Aktivitas yang mewakili pelaksanaan dalam pernyataan dalam prosedur atau pelaksanaan kegiatan dalam alur kerja.
2.		<i>Branch/Merge</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
3.		<i>Initial State</i>	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
4.		<i>Final State</i>	Status akhir yang dilakukan sistem.
5.		<i>Fork/Join</i>	Sebuah control yang ditampilkan dengan cara yang sama seperti statechart, oleh beberapa panah masuk atau meninggalkan baris heavy synchronization

2.8.2.6 Component Diagram

Component diagram menggambarkan struktur fisik kode dari komponen. Komponen dapat berupa *sourcecode*, komponent biner, atau *executable component*. Sebuah komponen berisi informasi tentang *logic class* atau *class* yang diimplementasikan sehingga membuat pemetaan dari *logical view* ke *component view* (Dana S. Kusumo 2004:4). Simbol-simbol yang digunakan pada *component diagram* disajikan pada Tabel 2.6.

Tabel 2.6. Tabel simbol *component diagram*

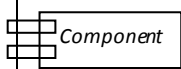
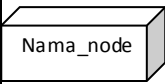

NO	GAMBAR	NAMA	KETERANGAN
1.		<i>Component</i>	Sebuah komponen melambangkan sebuah entitas <i>software</i> dalam sebuah sistem. Sebuah komponen dinotasikan sebagai sebuah kotak segiempat dengan dua kotak kecil tambahan yang menempel di sebelah kirinya.
2.		<i>Dependency</i>	Sebuah <i>Dependency</i> digunakan untuk menotasikan relasi antara dua komponen.

3 *Deployment Diagram*

Deployment Diagram menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak sistem, menunjukkan hubungan komputer dengan perangkat (*nodes*) satu sama lain dan jenis hubungannya. Di dalam *nodes*, *executeable component* dan *object* yang dialokasikan untuk memperlihatkan unit perangkat lunak yang di eksekusi oleh *node* tertentu dan ketergantungan komponen (Dana S. Kusumo 2004:4).

Simbol-simbol yang digunakan pada *deployment diagram* disajikan pada Tabel 2.7.

Tabel 2.7. Tabel simbol *deployment diagram*

NO	GAMBAR	NAMA	KETERANGAN
1.		<i>Component</i>	Komponen-komponen yang ada diletakkan di dalam <i>node</i> .
2.		<i>Node</i>	<i>Node</i> menggambarkan bagian-bagian <i>hardware</i> dalam sebuah sistem.
3.		<i>Association</i>	Sebuah <i>association</i> digambarkan sebagai sebuah garis yang menghubungkan dua <i>node</i> yang mengindikasikan jalur komunikasi antara element-elemen <i>hardware</i> .

2.9 Pengujian *Five view*

Disadur dari *SoftwareTesting and QA Theory and Practice (Chapter 17: Software Quality) University of Waterloo*, pengujian *fiveview* adalah pengujian yang sifatnya deskriptif dimana *software* yang diuji dinilai melalui lima sudut pandang atau kategori yang berbeda melalui penilaian dari *expertise* atau orang yang berpengalaman dari masing-masing sudut pandang. Lima sudut pandang tersebut adalah sebagai berikut:

2.9.1 *Transcendental view*

Kualitas menurut pandangan ini ialah sesuatu yang dapat dikenali melalui pengalaman tapi tidak dapat selalu digambarkan. Objek atau *software* yang bagus itu menonjol dan dapat dengan mudah dikenali.

2.9.2 *User view*

Kualitas menyangkut sejauh mana produk memenuhi kebutuhan dan harapan pengguna dan apakah suatu produk cocok untuk digunakan. Pendapat ini bersifat sangat *personal*. Sebuah produk berkualitas baik jika memuaskan

sebagian besar pengguna. Hal ini berguna untuk mengidentifikasi fitur dari produk yang pengguna anggap penting. Pandangan ini dapat mencakup banyak unsur subjek, seperti kegunaan, keandalan, dan keefisienan.

2.9.3 Manufacturing view

Pandangan ini berkaitan dengan faktor dalam industri manufaktur, apakah produk memenuhi persyaratan atau tidak. Setiap penyimpangan dari persyaratan yang dinilai mengurangi kualitas produk. Konsep proses memainkan peran kunci. Produk yang dibuat harus orisinal sehingga biaya berkurang, misal biaya pembangunan dan biaya pemeliharaan.

Kesesuaian dengan persyaratan dan spesifikasi menyebabkan keseragaman dalam produk tapi beberapa berpendapat bahwa keseragaman tersebut tidak menjamin kualitas. Kualitas produk dapat secara bertahap ditingkatkan dengan memperbaiki proses.

2.9.4 Product view

Jika sebuah produk diproduksi dengan sifat internal (misalnya bahan dan tindakan) yang baik, maka produk akan memiliki sifat eksternal atau *output* yang baik dan dapat dieksplorasi hubungan antara sifat internal dan kualitas eksternal.

2.9.5 Value-based view

Ini merupakan penggabungan dari dua konsep yaitu keunggulan dan kelayakan. Kualitas adalah ukuran dari keunggulan dan nilai adalah ukuran layak.

Berapa banyak pengguna bersedia membayar untuk tingkat kualitas tertentu. Kualitas tidak berarti jika produk memenuhi nilai ekonomi. Pandangan berbasis nilai antara biaya dan kualitas.