BAB 2

TINJAUAN PUSTAKA

2.1 Kebiasaan Hidup Berwawasan Lingkungan

Pendidikan lingkungan merupakan suatu proses yang bertujuan membentuk perilaku, nilai dan kebiasaan untuk menghargai lingkungan hidup. Dengan definisi diatas dapat menyimpulkan bahwa pendidikan lingkungan hidup harus diberikan sejak dini kepada anak-anak[2]. Tindakan ini untuk mempersiapkan anak-anak yang siap mencegah dan menghadapi bencana lingkungan. Penyadaran masalah-masalah lingkungan dapat dilakukan dengan beragam pendekatan dan beberapa tahap. Tahap-tahap tersebut sekurang-kurangnya ada 2, yaitu :

1. Menyadari bahwa masalah lingkungan sebagai suatu kenyataan yang harus di atasi.
2. Melakukan analisis terhadap masalah tersebut dengan jalan mengidentifikasi penyebab utama.

2.2 Tumbuhan

Tumbuhan terdiri dari akar, batang, daun, bunga, dan buah. Setiap bagian memiliki fungsi masing-masing[5].

1. Bagian- bagian akar

Bagian akar terdiri dari rambut (bulu) akar dan tudung akar. Rambut akar berfungsi menyerap air dan zat hara dari dalam tanah. Fungsi akar ada 3 yaitu, menopang tumbuhan, menyerap air dan zat hara, dan menyimpan cadangan makanan. Akar dibedakan menjadi dua jenis. Jenis akar yang pertama yaitu akar serabut yang berupa akar berukuran kecil dan tumbuh dipangkal batang dimana semua ukuran akar hampir sama besar dan sama panjang. Akar serabut dimiliki oleh tumbuhan monokotil, seperti padi, jagung, tebu dan kelapa. Jenis akar kedua adalah akar tunggang, berupa akar utama yang tumbuh lurus ke bawah dan merupakan kelanjutan dari batang, akar-akar yang lain merupakan cabang dari akar tunggang. Akar tunggang ini dimiliki oleh tumbuhan dikotil, seperti mangga dan jambu.

2. Bagian-bagian Batang

Pada batang berkayu, jika batang berkayu kita potong maka akan tampak kayu, pembuluh, dan kulit. Kayu sangat keras dan berguna untuk menopang pohon. Pembuluh adalah saluran untuk dilewati makana, garam mineral (zat-zat hara), dan air. Pembuluh ini terletak di bawah kulit. Pembuluh ada dua macam yaitu *xilem* dan *floem*. *Xilem* berfungsi mengangkut air dan garam mineral dari akar menuju daun. *Floem* berfungsi mengangkut makanan yang dibuat di daun ke seluruh bagian tumbuhan. Pada tumbuhan berkayu antara *xilem* dan *floem* terdapat bagian yang disebut kambium. Kambium mengalami pertumbuhan ke arah luar membentuk kulit dan ke arah dalam membentuk kayu. Lingkaran tahun dapat dilihat pada pohon batang berkayu. Fungsi batang adalah menyalurkan air, zat hara, dan makanan. Batang juga berfungsi sebagai alat pernapasan karena memiliki lentisel. Jenis batang ada dua, yaitu batang berkayu dan batang tidak berkayu. Batang berkayu terdiri dari batang basah dan batang rumput.

3. Bagian-bagian Daun

Daun merupakan bagian tumbuhan yang tumbuh di batang dan umumnya berwarna hijau. Daun memiliki tiga bagian utama yaitu pelapah daun, tangkai daun, dan helaian daun. Daun yang mempunyai semua bagian itu disebut daun lengkap, contohnya daun talas. Dan daun yang tidak mempunyai semua bagian itu disebut daun tidak lengkap, contohnya daun singkong, daun sirih, dan daun mangga. Berdasarkan jumlahnya daun dibedakan menjadi daun majemuk dan daun tunggal. Daun majemuk adalah daun yang memiliki beberapa helai daun pada setiap tangkai daun. Daun tunggal adalah daun yang memiliki satu helai daun pada setiap tangkai daun. Daun berfungsi sebagai tempat pembuatan makan, tempat penguapan dan alat pernapasan. Bagian tumbuhan tempat membuat makan adalah daun karena mengandung *klorofil* (zat hijau daun). Proses pembuatan makanan ini disebut *fotosintesis* dan dibantu oleh cahaya matahari. Daun berfungsi sebagai alat penguapan dan pernapasan karena memiliki *stomata* dipermukaannya. Bentuk daun ada 4 yaitu menyirip, melengkung, sejajar, dan menjari.

4. Bagian-bagian Bunga

Bunga terdiri dari tangkai bunga, kelopak bunga, mahkota bunga, benang sari, dan putik. Tangkai bunga, berfungsi menghubungkan bunga dengan batang. kelopak bunga, berfungsi membungkus mahkota bunga ketika masih kuncup. Mahkota bunga berfungsi sebagai perhiasan bunga. Benang sari, berfungsi sebagai alat kelamin jantan. Putik, berfungsi sebagai alat kelamin betina. Bunga yang memiliki benang sari dan putik disebut bunga sempurna. Jika hanya memiliki putik saja disebut bunga betina dan jika hanya memiliki benang sari saja disebut bunga jantan. Fungsi bunga adalah sebagai alat perkembangbiakan karena memiliki benang sari dan putik.

5. Bagian-bagian Buah

Buah merupakan hasil dari perkembangan bunga. Buah terdiri dari kulit buah, daging buah, dan biji. Kulit buah adalah lapisan luar buah. Daging buah adalah bagian yang biasa dimakan. Biji buah adalah bagian yang umumnya berada di tengah buah. Buah befungsi sebagai cadangan makanan dan tempat adanya biji.

2.3 *Metamorfosis*

Beberapa jenis hewan seperti kupu-kupu, katak, nyamuk, kecoak, dan capung mengalami *metamorfosis* pada daur hidupnya. Hewan-hewan tersebut baru menyerupai induknya setelah mengalami beberapa perubahan bentuk. *Metamorfosis* sempurna pada kupu-kupu dimulai dari telur ulat, kepompong sampai menjadi kupu-kupu.

2.4 Pengertian *Game*

Dalam kamus bahasa Indonesia, "*Game"* adalah permainan, yaitu kegiatan yang kompleks yang di dalamnya terdapat peraturan, *play*, dan budaya. *Game* merupakan permainan komputer yang dibuat dengan teknik dan metode animasi. Jika ingin mendalami pengunaan animasi harus memahami pembuatan *game* atau jika ingin membuat *game*, maka harus memahami teknik dan metode animasi, sebab keduanya saling berkaitan[6].

2.4.1 *Game* Edukasi

Game edukasi adalah salah satu bentuk *genre game* yang digunakan untuk memberikan pembelajaran kepada penggunanya melalui media permainan yang

menarik. *Game* edukasi biasanya ditujukan untuk anak-anak sekolah dasar sehingga permainan warna sangat diperlukan [7].

2.4.2 *Game Adventure*

Game adventure adalah jenis *game* bersifat petualangan dimana pemain mengendalikan sebuah tokoh yang mengikuti suatu alur cerita dengan menghadapi berbagai tantangan dan teka-teki [8].

2.5 Pemograman Berbasis Objek

Pemograman berorientasi objek (*Object-Oriented Programing*) adalah pembangunan perangkat lunak yang mengoragisasikan preangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan. Pemograman berorientasi objek merupakan suatu cara bagaimana system perangkat lunak dibangun melalui pendakatan objek [9].

2.5.1 Konsep Dasar Berorientasi Objek

Pendekatan berorientasi objek merupakan suatu tehnik atau cara pendekatan dalam melihat permasalahan dan sistem. Pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata. Berikut adalah beberapa konsep dasar yang harus dipahami tentang pemograman berorientasi objek :

1. Kelas (*class*)

Kelas adalah kumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statis dan himpunan objek yang sama yang mungkin lahir atau diciptakan dari kelas tersebut.

2. Objek (*object*)

Objek adalah abstraksi dan sesuatu yang mewakili dunia nyata yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi yang dapat diterapkan atau berpengaruh pada status objeknya. Secara teknis, sebuah kelas saat program dieksekusi maka akan dibuat sebuah objek.

3. Metode (*method*)

Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek. Metode atau Operasi dapat berasal dari event, aktifitas atau aksi, keadaan, fungsi, dan kelakuan dunia nyata.

4. Atribut (*attribute*)

Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek.

5. Abstraksi (*abstraction*)

Prinsip untuk mempresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

6. Enkapulasi (*encapsulation*)

Pembungkusan atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi objek sehingga objek lain tidak mengetahui cara kerjannya.

7. Pewarisan (*inheritance*)

Mekanisme yang memungkinkan satu objek mewarisi sebagian atau seluruh definisi dan objek lain sebagai bagian dari dirinya.

8. Antarmuka (*interface*)

Antarmuka atau *interface* sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang di deklarasikan tanpa isi.

9. Reusability

Pemanfaatan kembali objek yang sudah didefinisikan untuk suatu permasalahan pada permasalahan lainnya yang melibatkan objek tersebut.

10. Generalisasi dan Spesialisasi

Menunjukan hubungan antara kelas dan objek yang umum dengan kelas dan objek yang khusus.

11. Komunikasi Antar Objek

Komunikasi antar objek dilakukan lewat pesan (*message*) yang dikirim dari suatu objek ke objek lainnya.

2.5.2 Analisis Berorientasi Objek

Analisis berorientasi objek atau *Object Oriented Analysis* (OOA) adalah tahapan unntuk menganalisis spesifikasi atau kebutuhan akan sistem yang akan dibangun dengan konsep berorientasi objek.

2.5.3 Desain Berorientasi Objek

Desain berorientasi objek atau *Object Oriented Design* (OOD) adalah tahapan perantara untuk memetakan spesifikasi atau kebutuhan sistem yang akan dibangun dengan konsep berorientasi objek ke desain pemodelan agar lebih mudah diimplementasikan dengan pemograman berorientasi objek.

Pemodelan berorientasi objek biasanya dituangkan dalam dokumentasi perangkat lunak dengan menggunakan perangkat pemodelan beorientasi objek, diantaranya adalah UML (*Unified Modeling Language*).

2.5.4 Bahasa Pemograman yang Mendukung PBO

1. Bahasa Pemograman *Smalltalk*

   Smaltalk merupakan salah satu bahasa pemograman yang dikembangkan untuk mendukung pemograman berorientasi objek mulai tahun 1978.

2. Bahasa Pemograman *Eiffel*

   Effiel merupakan bahasa pemograman yang dikembangkan untuk mendukung pemograman berorientasi objek mulai than 1985 oleh Bertrand Mater dan *compiler* Eiffel selesai dibuat pada tahun 1987.

3. Bahasa Pemograman C++

   Bahasa pemograman C++ merupakan pengembangan lebih lanjut dari bahasa pemograman C untuk mendukung pemograman berorientasi objek.

4. Bahasa Pemograman PHP

   PHP dibuat pertama kali oleh seorang perekayasa perangkat lunak (*softwaew engineering*) yang bernama Rasmus Lerdoff. Dengan beberapa versi yang telah diperbaharui. PHP 5 sudah mendukung pemograman berorientasi objek.

5. Bahasa Pemograman *Java*

   Java dikembangkan oleh perusahaan *Sun Microsystem*. *Java* menurut *Sun Microsystem* adalah nama untuk sekumpulan teknologi untuk membuat dan

menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan. *Java* merupakan bahasa pemograman objek murni karena semua kode program dibungkus dalam kelas.

6. Bahasa Pemograman *Actionscript* 3.0

Bahasa berorientasi objek untuk membuat aplikasi dan konten multimedia ber-*script* untuk pemutaran di *runtime flash client* (seperti *Flash Player* dan *Adobe AIR*). Dengan sintaks yang mirip dengan *Java* dan *C#*.

2.6 Pemodelan Sistem

Pemodelan sistem yang berorientasi objek dapat dimodelkan dengan menggunakan UML (*Unified Modeling Language*). UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Berikut merupakan bagian-bagian dari UML[10].

1. *Use Case*

Use Case merupakan pemodelan dari kebutuhan sistem terhadap interaksi pengguna kepada sistem. *Use Case* dapat menjadi gambaran kejadian dimana pengguna melakukan interaksi terhadap sistem.

2. *Activity Diagram*

Activity Diagram merupakan sebuah notasi diagram yang menjelaskan tentang sebuah sekenario yang terjadi pada sebuah *use case* dimana untuk memudahkan pembacaaan sekenario.

3. *Class Diagram*

Class Diagram merupakan pemodelan dari sebuah *class-class* yang akan dibangun pada sebuah sistem. *Class diagram* akan menggambarkan semua relasi yang terjadi terhadap *class-class*.

4. *Sequence Diagram*

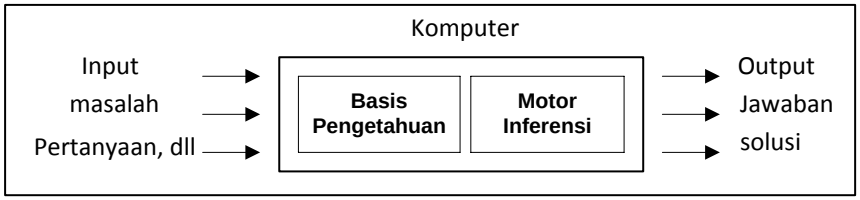
Sequence diagram merupakan pemodelan untuk menggambarkan interaksi antar pengguna dan sistem, dimana pada suatu *use case* akan dijelaskan secara sekuen bagaimana porses interaksi anatara pengguna dan sistem dijelaskan. *Sequence diagram* akan menjelaskan keterkaitan *class-class* yang akan dibangun.

2.7 *Artificial Intelligence*

Kecerdasan buatan atau *Artificial Intelligence* merupakan salah satu bagian ilmu komputer yang membuat agar mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia[11].

Untuk melakukan aplikasi kecerdasan buatan ada 2 bagian utama yang sangat dibutuhkan, yaitu [11] :

1. Basis Pengetahuan (*Knowledge Base*), berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan lainnya.
2. Motor Inferensi (*Inference Engine*), yaitu kemampuan menarik kesimpulan berdasarkan pengalaman.



Gambar 2. 1 Sistem Yang Menggunakan Kecerdasan Buatan [11]

Input yang diberikan pada system yang menggunakan kecerdasan buatan berupa masalah. Pada sistem harus dilengkapi dengan sekumpulan pengetahuan yang ada pada basis pengetahuan. Sistem harus memiliki *interface engine* agar mampu mengambil kesimpulan berdasarkan fakta atau pengetahuan. *Output* yang diberikan berupa solusi masalah sebagai hasil dari inferensi.

Secara umum untuk membangun suatu *system* yang mampu menyelesaikan masalah, perlu pertimbangkan 4 hal[11] :

1. Mendefinisikan masalah dengan tepat. Pendefinisian ini mencakup spesifikasi yang tepat mengenai keadaan awal dan solusi yang diharapkan.
2. Menganalisis masalah tersebut serta mencari beberapa teknik penyelesaian masalah yang sesuai.
3. Merepresentasikan pengetahuan yang perlu untuk menyelesaikan masalah tersebut.
4. Memilih teknik penyelesaian masalah yang terbaik.

2.7.1 Searching Sebagai Teknik Pemecahan Masalah

Ada banyak teknik-teknik penyelesaian masalah yang dapat dipakai diantaranya yaitu[11]:

1. *Searching* (pencarian), pemecahan masalah dengan menggunakan searching akan lebih mudah bila obyeknya direpresentasikan sebagai *graph*.
2. *Reasoning* (penalaran)
3. *Planning* yaitu memecah masalah kedalam *sub-sub* masalah yang lebih kecil, menyelesaikan *sub-sub* masalah satu demi satu kemudian menggabungkan solusi dari sub-sub masalah tersebut menjadi sebuah solusi yang lengkap.
4. *Learning* yaitu program komputer yang secara otomatis sanggup belajar dan meningkatkan *performance*-nya melalui pengalaman.

Pada dasarnya teknik *searching* (pencarian) dibagi menjadi 2 kelompok besar yaitu pencarian buta (*blind search*) dan pencarian terbimbing (*heuristic search*). Pencarian buta (*blind search*) adalah pencarian solusi tanpa adanya informasi yang dapat mengarahkan pencarian untuk mencapai *goal state* (tujuan) dari *current state* (keadaan sekarang). Selain itu Pencarian buta (*blind search*) biasanya tidak efisien karena waktu akses dan memori yang dibutuhkan cukup besar. Untuk mengatasi hal ini maka perlu ditambahkan suatu informasi pada domain yang bersangkutan sehingga proses pencarian yang baru akan dihasilkan. Pencarian seperti ini disebut sebagai *informed search* atau pencarian terbimbing (*heuristic search*). Teknik pencarian terbimbing (*heuristic search*) merupakan suatu strategi untuk melakukan proses pencarian secara selektif dan dapat memandu proses pencarian yang memiliki kemungkinan sukses paling besar.

2.7.2 Algoritma A*

Bentuk yang paling dikenal luas dari perbaikan *metode best-first search* disebut A* *search* (diucapkan "A-starsearch"). A* mengevaluasi node dengan menggabungkan g (n) biaya untuk mencapai node, dan h (n) untuk mendapatkan biaya dari node asal ke node tujuan. A* akan meminimumkan total biaya lintasan. Pada kondisi yang tepat, A* akan memberikan solusi yang terbaik dalam waktu yang optimal [12].

Algoritma A* menyelesaikan masalah yang menggunakan graf untuk perluasan ruang statusnya. Dengan kata lain digunakan untuk menyelesaikan permasalah yang bisa direpresentasikan dengan graf. Algoritma A* menerapkan teknik *heuristic* dalam membantu penyelesaian persoalan. *Heuristic* adalah penilai yang memberi harga pada tiap simpul yang memandu A* mendapatkan solusi yang diinginkan. Dengan *Heuristic* yang benar, maka A* pasti akan mendapatkan solusi (jika memang ada solusinya) yang dicari. *Heuristic* yang digunakan adalah *heuristic manhattan. Heuristic manhattan* yaitu *h(n) = abs(n.x-tujuan.x) + abs(n.y-tujuan.y).*

Beberapa terminologi dasar yang terdapat pada algoritma yaitu[11] :

1. Simpul (*node*) adalah petak-petak kecil sebagai representasi dari *area* pencarian (*pathfinding*). Bentuknya dapat berupa persegi, lingkaran, maupun segitiga.
2. Simpul asal/mulai (*source node*) adalah sebuah terminologi untuk posisi awal sebuah benda.
3. Simpul akhir/tujuan (*destination node*) adalah tempat tujuan yang ingin dicapai pada pencarian.
4. Simpul sekarang (*current node*) adalah simpul terbaik sebelumnya yang dipilih dan menjadi titik acuan untuk membangkitkan simpul tetangganya (*adjacent*).
5. Open list adalah tempat menyimpan data simpul yang diakses dari simpul asal/mulai (*source node*) atau dari tetangga simpul sekarang (*current node*) yang belum pernah berada di *open list* maupun *closed list*.
6. Closed list adalah tempat menyimpan simpul yang pernah menjadi simpul sekarang (*current node*).
7. Halangan/penghalang adalah simpul yang tidak dapat dilalui.

Fungsi f sebagai estimasi fungsi evaluasi terhadap *node* n dapat dituliskan :

$$F(n) = g(n) + h(n)$$

Gambar 2. 2 Fungsi Evaluasi [11]

dengan :

f(n) = fungsi evaluasi.

g(n) = biaya yang sudah dikeluarkan dari keadaan awal sampai keadaan n.

h(n) = estimasi biaya untuk sampai pada suatu tujuan mulai dari n.

Simpan *source node* ke *open list* dan *closed list*, karena simpul ini terpilih sebagai *Bestnode*. Bangkitkan semua suksesor dari simpul tersebut. Jika semua suksesornya tidak ada di *open list* maupun di *closed list*, maka masukan suksesornya ke *open list*. Semua suksesor cek nilai f nya, nilai f terkecil. Setelah itu *current node* simpan suksesornya dan *current node* tersebut menjadi *parent* yang baru. Begitu seterusnya hingga mencapai simpul tujuan (*destination node*). Jika suksesor yang dibangkitkan sudah ada di *open list* maupun di *closed list* maka abaikan.

2.8 *Tools* yang Digunakan

Dalam bagian ini akan membahas *tools* yang digunakan yaitu *Adobe Flash CS5, Actionscript 3, dan Adobe Photoshop CS3*.

2.8.1 *Adobe Flash CS5*

Adobe flash CS5 adalah salah satu program unggulan dari *adobe system* yang khusus digunakan untuk membuat animasi gambar *vektor*. Animasi yang dihasilkan menggunakan file *extension .swf* dan dapat dijalankan di *web browser* yang telah ter-*install Adobe Flash Player*. *Flash CS5* telah menjadi standar program pembuatan animasi seperti membangun situs *game*, *banner*, menu interaktif, *e-card* sampai pengembangan aplikasi untuk *mobile*, *smartphone* dan komputer tablet[13].

2.8.2 *Actionscript* 3.0

Actionscript 3.0 adalah adalah bahasa berorientasi objek untuk membuat aplikasi dan konten multimedia ber-*script* untuk pemutaran di *runtime flash client* (seperti *Flash Player* dan *Adobe AIR*). Dengan sintaks yang mirip dengan *Java* dan *C#*, bahasa *Actionscript* seharusnya akrab bagi *programmer* berpengalaman [14]. Beberapa fitur kunci dari bahasa *Actionscript* 3.0 adalah :

1. *First-class support* untuk *constructs* berorientasi objek umum, seperti kelas, objek, dan *interface*

2. *Single-threaded execution model*
3. *Runtime type-checking*
4. *Compile-time type-checking* opsional
5. Fitur dinamis seperti *runtime* pembuatan fungsi konstruktor baru dan *variable*
6. *Runtime exceptions*
7. *Direct support* untuk CML sebagai *tipe* data *built-in*
8. *Package* untuk pengorganisasian *code libraries*
9. *Namespaces* untuk kualifikasi *identifier*
10. *Regular expressions*

2.8.3 *Adobe Photoshop CS3*

Adobe Photoshop atau biasa disebut *Photoshop* adalah perangkat lunak editor citra buatan *Adobe Systems* yang dikhususkan untuk pengeditan foto atau gambar dan pembuatan efek. Perangkat lunak ini banyak digunakan oleh fotografer digital dan perusahaan iklan sehingga dianggap sebagai pimimpin pasar (*Merket Leader*) untuk perangkat lunak pengolahan gambar atau foto. Versi kedelapan aplikasi ini disebut dengan *Photoshop CS* (*Creative Suite*), versi sembilan disebut *Adobe Photoshop CS2*, versi sepuluh disebut *Adobe Photoshop CS3*. *Photoshop* tersedia untuk *Microsoft Windows, Mac OS X, Mac OS versi 9* ke atas. Dapat juga digunakan oleh sistem operasi lain seperti *Linux* dengan bantuan perangkat lunak tertentu seperti *CrossOver*.

Tool dalam *Adobe Photoshop* adalah alat yang dapat membantu pengguna dalam mengedit. *Adobe Photoshop CS3* memiliki 59 *tool* yang dapat dipakai oleh pengguna. *Tool* tersebut terdiri dari berbagai macam *tool* dengan kegunaan yang spesifik [15].

Beberapa *tool - tool* yang ada di *photoshop* antara lain :

1. *History Brush Tool*
2. *Eraser Tool*
3. *Path Selection Tool*
4. *Direct Selection tool*
5. *Pen Tool*

6. *Shape Tool*
7. *Brush Tool*
8. *Audio Annotation Tool*
9. *Eyedropper Tool*
10. *Measure Tool*
11. *Text Tool*
12. *Hand Tool*