BAB II
TINJAUAN PUSTAKA

2.1. Tinjauan Sekolah

Pada sub bab ini membahas peninjauan terhadap tempat penelitian yaitu SMA Negeri 1 Ciledug. SMA Negeri 1 Ciledug adalah salah satu Sekolah Menengah Atas (SMA) yang berada di kabupaten Cirebon. Sekolah ini dibangun diatas tanah seluas 5.660 m^2. Sekolah ini mendapat akreditasi A dari dinas Pendidikan.

2.1.1. Sejarah Sekolah

Pendidikan merupakan suatu bagian yang tak terpisahkan dari kebutuhan hidup suatu bangsa, karena pendidikan adalah salah satu modal dasar untuk Mengembangkan kepribadian suatu bangsa. Sesuai dengan amanat pembukaan Undang-Undang Dasar 1945 yaitu mencerdaskan kehidupan bangsa. Sedangkan Pendidikan kita telah diatur oleh UU No.2 tahun 1989 tentang sistem pendidikan nasional yang berdasarkan Pancasila dan Undang-Undang Dasar 1945.

Sejalan dengan upaya pemerintah dalam meningkatkan pelayanan dan kualitas pendidikan yang dapat dicapai apabila didukung oleh sarana dan prasarana yang memadai. Untuk itu maka SMA Negeri 1 Ciledug terus berupaya berbenah diri melakukan peningkatan baik secara fisik maupun mutu outputnya.

SMA Negeri 1 Ciledug pada awalnya merupakan sekolah swasta dibawah naungan Yayasan Pemda Kabupaten Cirebon. Kemudian dialihkan statusnya menjadi Negeri pada tahun 1999. Usianya yang masih muda dan dengan latar belakang pengalihan status tersebut menjadikan SMA Negeri 1 Ciledug sulit untuk sejajar dengan SMA Negeri lainnya karena dari infrastruktur yang ada masih banyak yang rusak dan kurang memenuhi syarat.

Dengan predikat sekolah yang terakreditasi A, RSSN dan MBS maka kami siap membenahi diri untuk mempersiapkan dan melaksanakan program untuk

mewujudkan upaya peningkatan dan pengembangan SMA Negeri I Ciledug sesuai dengan mottonya "PRIMA" (prestasi, Religius, Intelektual, dan Mandiri) dibutuhkan dukungan dan bantuan dari semua pihak baik masyarakat, orang tua, pemerintah daerah kabupaten, provinsi sampai pusat untuk melengkapi sarana dan prasarana sebagai factor penunjang dalam kelancaran proses pembelajaran. Misalnya untuk menerapkan program pengembangan ICT perlu menggunakan media atau alat pembelajaran yang memadai yaitu menggunakan komputer yang cukup memadai baik dari kuantitas maupun kualitasnya.

2.1.2. Logo Sekolah

Sebagaimana kita ketahui bahwa setiap daerah, instansi maupun organisasi mempunyai logo dengan ciri khas masing-masing. Logo merupakan gambaran identitas suatu daerah, suatu instansi maupun organisasi. Berikut ini adalah logo dari SMAN 1 Ciledug.



Gambar 2. 1 Logo SMAN 1 Ciledug

2.1.3. Visi dan Misi

Visi dan misi merupakan syarat wajib bagi sebuah instansi atau organisasi. Setiap instansi atau organisasi memiliki visi dan misi yang berbeda, semua tergantung tujuan yang akan dicapai oleh masing-masing instansi atau organisasi. Berikut ini merupakan visi dan misi yang dimiliki SMAN 1 Ciledug :

A. Visi

Terwujudnya insan pendidikan yang memiliki kualifikasi : PRIMA

(Prestasi, Religius, Intelektual, Mandiri).

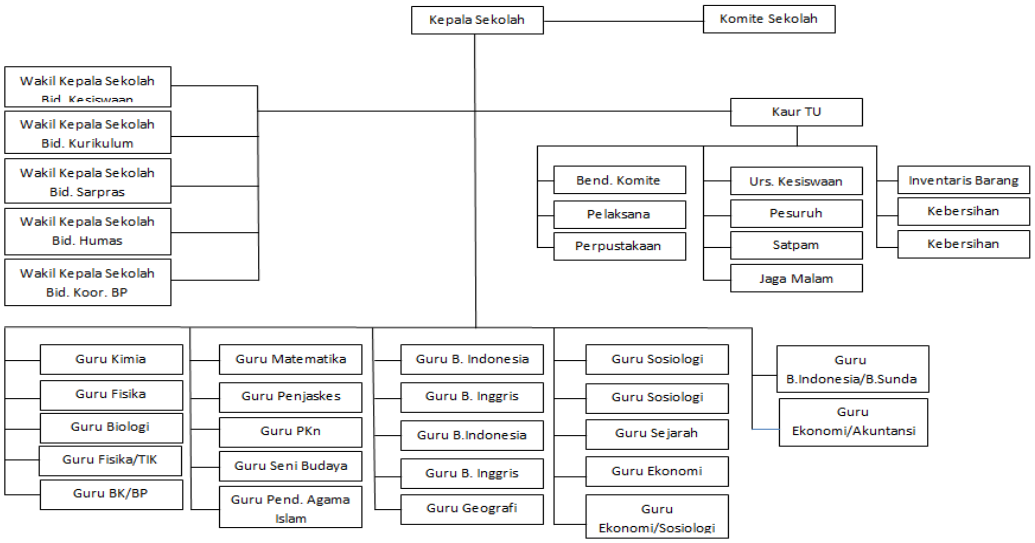
B. Misi

1. Mewujudkan anak didik yang berprestasi.berakhlak mulia dan berketerampilan.
2. Mewujudkan lingkungan pendidikan yang agamis, asri dan kondusif.
3. Meningkatkan prestasi kerja yang efektif ,efesien dan tertib.
4. Mengembangkan hubungan yang harmonis baik kedalam maupun keluar.
5. Mewujudkan keseimbangan antara hak dan kewajiban.

2.1.4. Struktur Organisasi

Struktur organisasi merupakan suatu kerangka dalam manajemen sekolah agar suatu sekolah dapat berjalan sebagaimana mestinya sesuai dengan tujuan pertama kali berdirinya perusahaan/organisasi tersebut. Struktur organisasi memuat tatanan kerja yang ada di sekolah tersebut yang menyangkut tugas, tanggung jawab dan wewenang. Semuanya ada agar tercipta hubungan yang baik yang lebih dinamis dan masing-masing mempunyai tugas dan tanggung jawab sendiri juga mempunyai rasa hormat terhadap yang lain baik itu berkedudukan diatas, dibawah atau setingkat. Tidak berjalannya struktur organisasi dengan baik akan mengarahkan perusahaan tersebut pada kehancuran maka dari itu struktur organisasi merupakan faktor penting dalam suatu sekolah.

Struktur organisasi merupakan suatu bentuk susunan keanggotaan yang membedakan jabatan masing-masing personil didalam suatu perusahaan, manfaat lainnya dengan dibuatnya struktur organisasi dalam proses kerja. Pegawai perlu mengetahui struktur organisasi perusahaan, agar dapat melakukan kegiatan yang sesuai dengan fungsi dan tidak menyimpang dari yang telah ditentukan, sehingga pegawai mengetahui hak dan kewajibannya didalam sekolah tersebut. Berikut ini adalah gambar struktur organisasi SMAN 1 Ciledug.



Gambar 2. 2 Struktur Organisasi SMAN 1 Ciledug

2.2. Landasan Teori

Landasan teori dimanfaatkan sebagai pemandu agar fokus penelitian sesuai dengan fakta di lapangan. Selain itu landasan teori juga bermanfaat untuk memberikan gambaran umum tentang latar penelitian dan sebagai bahan pembahasan hasil penelitian. Peneliti mengutip beberapa teori yang berhubungan dengan variabel-variabel penelitian. Dan teori-teori ini merupakan landasan dalam penelitian.

2.2.1. Fisika

Fisika adalah sains atau ilmu tentang alam dalam makna yang terluas. Fisika mempelajari gejala alam yang tidak hidup atau materi dalam lingkup ruang dan waktu. Para fisikawan atau ahli fisika mempelajari perilaku dan sifat materi dalam bidang yang sangat beragam, mulai dari partikel submikroskopis yang membentuk segala materi (fisika partikel) hingga perilaku materi alam semesta sebagai satu kesatuankosmos.

Beberapa sifat yang dipelajari dalam fisika merupakan sifat yang ada dalam semua sistem materi yang ada, seperti hukumkekekalan energi. Sifat semacam ini sering disebut sebagai hukum fisika. Fisika sering disebut sebagai "ilmu paling mendasar", karena setiap ilmu alam lainnya (biologi, kimia, geologi, dan lain-lain) mempelajari jenis sistem materi tertentu yang mematuhi hukum

fisika. Misalnya, kimia adalah ilmu tentang molekul dan zat kimia yang dibentuknya. Sifat suatu zat kimia ditentukan oleh sifat molekul yang membentuknya, yang dapat dijelaskan oleh ilmu fisika seperti mekanika kuantum,termodinamika, dan elektromagnetika.

Fisika juga berkaitan erat dengan matematika. Teori fisika banyak dinyatakan dalam notasi matematis, dan matematika yang digunakan biasanya lebih rumit daripada matematika yang digunakan dalam bidang sains lainnya. Perbedaan antara fisika dan matematika adalah: fisika berkaitan dengan pemerian dunia material, sedangkan matematika berkaitan dengan pola-pola abstrak yang tak selalu berhubungan dengan dunia material. Namun, perbedaan ini tidak selalu tampak jelas. Ada wilayah luas penelitan yang beririsan antara fisika dan matematika, yakni fisika matematis, yang mengembangkan struktur matematis bagi teori-teori fisika.

2.2.1.1. Pembelajaran Fisika

Pendidikan Fisika menjadi suatu bidang ilmu yang memiliki agar setiap siswa terutama yang duduk di bangku sekolah menengah atas (SMA) Jurusan IPA, memiliki kemampuan untuk bernalar dan berpikir analisis induktif dan deduktif dengan menggunakan prinsip dan konsep fisika untuk menjelaskan berbagai peristiwa alam dan menyelesaikan masalah baik secara kualitatif maupun kuantitatif.

SMAN 1 Ciledug sampai saat masih menggunakan kurikulum 2010. Adapun materi-materi yang diberikan, seperti vektor, gerak melingkar, gerak parabola, gaya gesekan, gaya gravitasi, elastisitas bahan, gerak harmonik sederhana, usaha, energi, daya, energi potensial, impuls, momentum dan hukum kekekalan momentum.

2.2.2. Multimedia

Multimedia adalah penggunaan komputer untuk menyajikan dan menggabungkan teks, suara, gambar, animasi, audio dan video dengan alat bantu (*tool*) dan koneksi (*link*) sehingga pengguna dapat melakukan navigasi,

berinteraksi, berkarya dan berkomunikasi. Multimedia sering digunakan dalam dunia informatika. Selain dari dunia informatika, multimedia juga diadopsi oleh dunia *game*, dan juga untuk membuat *website*.

Multimedia dimanfaatkan juga dalam dunia pendidikan dan bisnis. Di dunia pendidikan, multimedia digunakan sebagai media pengajaran, baik dalam kelas maupun secara sendiri-sendiri atau otodidak. Di dunia bisnis, multimedia digunakan sebagai media profil perusahaan, profil produk, bahkan sebagai media kios informasi dan pelatihan dalam sistem *e-learning*.

Pada awalnya multimedia hanya mencakup media yang menjadi konsumsi indra penglihatan (gambar diam, teks, gambar gerak video, dan gambar gerak rekaan/animasi), dan konsumsi indra pendengaran (suara) dan juga berupa (berwujud). Dalam perkembangannya multimedia mencakup juga kinetik (gerak) dan bau yang merupakan konsumsi indra penciuman. Multimedia mulai memasukkan unsur kinetik sejak diaplikasikan pada pertunjukan film 3 dimensi yang digabungkan dengan gerakan pada kursi tempat duduk penonton. Kinetik, dan film 3 dimensi membangkitkan *sense* realistis.

Bau mulai menjadi bagian dari multimedia sejak ditemukan teknologi reproduksi bau melalui telekomunikasi. Dengan perangkat *input* pendeteksi bau, seorang operator dapat mengirimkan hasil *digitizing* bau tersebut melalui internet. Komputer penerima harus menyediakan perangkat *output* berupa mesin reproduksi bau. Mesin reproduksi bau ini mencampurkan berbagai jenis bahan bau yang setelah dicampur menghasilkan *output* berupa bau yang mirip dengan *data* yang dikirim dari internet. Dengan menganalogikan dengan *printer*, alat ini menjadikan feromon-feromon bau sebagai pengganti tinta. *Output* bukan berupa cetakan melainkan aroma.

2.2.2.1. Elemen Multimedia

Teknologi multimedia mulai diperkenalkan pada pertengahan tahun 80-an dan telah membuka dimensi baru dalam Teknologi Informasi. Teknologi multimedia telah membawa satu perubahan drastis pada industri media karena

kemampuannya menyatukan kelebihan televisi sebagai alat *audio visual* dan kekuatan teknologi mesin cetak dan kemampuan teknologi interaktif komputer.

Terdapat 5 elemen multimedia penting yang mendasar pada yaitu:

1. Teks

   Merupakan elemen multimedia yang menjadi dasar utama dalam menyampaikan informasi, karena teks adalah jenis data yang paling sederhana dan membutuhkan tempat penyimpanan yang paling kecil. Biasanya dihasilkan oleh program pengolah kata dan merupakan informasi yang utama pada sebagian besar multimedia. Teks memegang peranan dasar dalam menyusun dokumen, karena hampir seluruh aplikasi multimedia menggunakan teks sebagai alat presentasi informasi yang paling sesuai untuk mendeskripsikan suatu nama, definisi atau aturan Memberi penekanan untuk sesuatu materi yang ingin disampaikan. Jenis-jenis teks yang terdapat pada sebuah *software* ada berbagai macam, seperti : *Times New Roman, Arial, Calibri, Comic sans* dan lain lain.

2. Grafis

   Merupakan elemen paling penting, memberikan penekanan secara visual terhadap sesuatu presentasi maklumat. Membantu menyampaikan informasi dengan lebih berkesan.Menjadikan presentasi atau penyampaian informasi dengan lebih menarik. Penyampaian informasi akan menjadi semakin efektif dan bermanfaat, terutama informasi yang tidak dapat dijelaskan dengan kata-kata. Berikut adalah bidang aplikasi spesifik dari grafika komputer:

   a. Antarmuka pengguna
   b. Peta (*Cartography*)
   c. Kesehatan
   d. Perancangan objek
   e. Sistem multimedia
   f. Presentasi grafik
   g. Presentasi saintifik
   h. Pemrosesan citra

i. Simulasi

3. Audio

Membantu menyampaikan informasi dengan lebih efektif (misalnya: penggunaan suara latar atau kesan audio khusus). Membantu meningkatkan daya tarik terhadap sesuatu tayangan.Membantu meningkatkan daya tarik terhadap isi yang di presentasikan. Multimedia tidak akan lengkap jika tanpa suara. Suara terbagi menjadi tiga katagori :

a. Ucapan (*speech*) : suara orang berbicara.
b. Musik (*music*) : suara yang dihasilkan oleh alat musik.
c. Efek suara (*sound effect*) : suara yang dibuat untuk menciptakan kesan atau kejadian, seperti suara tembakan, halilintar, gelas pecah, dll.

4. Video

Menyediakan metode penyaluran informasi yang amat menarik dan hidup. Merupakan elemen atau media yang sangat dinamis dan efektif dalam menyampaikan informasi. Merupakan gabungan elemen multimedia yang lengkap karena menggabungkan semua elemen multimedia yanga ada untuk menyajikan informasi video menggunakan system animasi yang diambil melalui suatu kamera video dan disimpan dalam bentuk file dan format tertentu.

5. Animasi

Berperan sebagai penarik pada sebuah presentasi dan sangat membantu dalam menjelaskan suatu konsep yang kompleks dengan mudah. Animasi mengacu pada gambar-gambar yang bergerak. Animasi dapat dihasilkan dengan menayangkan frame-frame (bingkai-bingkai) gambar secara cepat untuk menghasilkan efek pergerakan. Animasi dapat dibagi ke dalam dua katagori :

a. Animasi buatan. Animasi yang dihasilkan oleh manusia melalui bantuan program komputer.
b. Animasi video. Animasi yang dihasilkan melalui rekaman darikamera video dan kejadian yang sesungguhnya.

2.2.3. Augmented Reality

Augmented Reality adalah teknologi yang menggabungkan benda maya dua dimensi dan ataupun tiga dimensi ke dalam sebuah lingkungan nyata tiga dimensi lalu memproyeksikan benda-benda maya tersebut dalam waktu nyata. Tidak seperti realitas maya yang sepenuhnya menggantikan kenyataan, realitas tertambah sekadar menambahkan atau melengkapi kenyataan.

Benda-benda maya menampilkan informasi yang tidak dapat diterima oleh pengguna dengan inderanya sendiri. Hal ini membuat realitas tertambah sesuai sebagai alat untuk membantu persepsi dan interaksi penggunanya dengan dunia nyata. Informasi yang ditampilkan oleh benda maya membantu pengguna melaksanakan kegiatan-kegiatan dalam dunia nyata.

Realitas tertambah dapat diaplikasikan untuk semua indera, termasuk pendengaran, sentuhan, dan penciuman. Selain digunakan dalam bidang-bidang seperti kesehatan, militer, industri manufaktur, realitas tertambah juga telah diaplikasikan dalam perangkat-perangkat yang digunakan orang banyak, seperti pada telepon genggam.

2.2.3.1. Sejarah Augmented Reality

Konsep pertama *augmented reality* dikenalkan oleh Morton Heilig, seorang *cinematographer* pada tahun 1950an. Ketika itu *Augmented Reality* membutuhkan sebuah alat yang besar sebagai alat *output*. Alat *output* dapat berupa yang dipasang ditubuh kita (dikenal dengan nama HMD, *Head Mounted Device*), ada juga yang berupa monitor, seperti monitor TV, LCD, monitor ponsel, dll. Alat HMD pertama kali ditemukan pada tahun 1968 oleh Ivan Sutherland dari Harvard University. *Augmented reality* dengan *input* berupa sensor GPS diperkenalkan pada tahun 2003 dari hasil penelitian Loomis, dkk pada karya ilmiahnya *Personal guidance system for the visually impaired using GPS, GIS, and VR technologies*, pada tahun 1994.

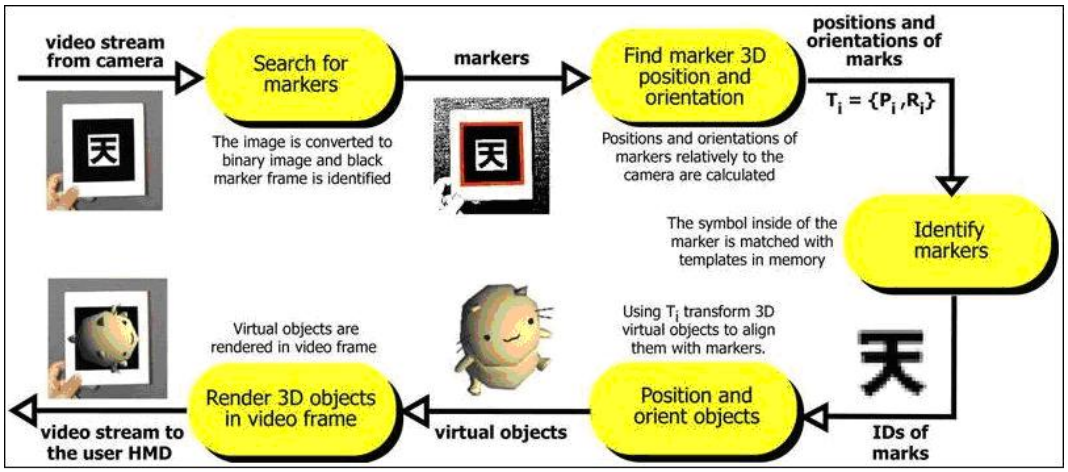
Pada tahun 1996, Rekimoto dalam karya ilmiahnya *Augmented Reality Using the 2D Matrix Code. In Proceedings of the Workshop on Interactive Systems and Software* memperkenalkan marker 2D untuk pertama kalinya. Dua

tahun kemudian ARtoolkit, *augmented reality library* pertama kali diluncurkan oleh Kato.

Pada tahun 2009 Lab MIT(Mistry, dkk) meneliti *sixth sense project dan Wear Ur World – A Wearable Gestural Interface* dimana *augmented reality* di implementasikan pada kehidupan sehari-hari.

2.2.3.2. Prinsip Kerja *Augmented Reality*

Sistem *Augmented Reality* bekerja berdasarkan deteksi citra dan citra yang digunakan adalah *marker*. Prinsip kerjanya sebenarnya cukup sederhana. *Camera* yang telah dikalibrasi akan mendeteksi marker yang diberikan, kemudian setelah mengenali dan menandai pola *marker*, *webcam* akan melakukan perhitungan apakah *marker* sesuai dengan *database* yang dimiliki. Bila tidak, maka informasi *marker* tidak akan diolah, tetapi bila sesuai maka informasi *marker* akan digunakan untuk me-*render* dan menampilkan objek 3D atau animasi yang telah dibuat sebelumnya.



Gambar 2. 3 Prinsip Kerja *Augmented Reality*

2.2.3.3. Metode Augmented Reality

Metode yang dikembangkan pada *augmented reality* saat ini terbagi menjadi dua metode, yaitu *Marker Based Tracking* dan *Markless Augmented Reality*.

a. Marker Based Tracking

Marker biasanya merupakan ilustrasi hitam dan putih persegi dengan batas hitam tebal dan latar belakang putih. Komputer akan mengenali posisi dan orientasi marker dan menciptakan dunia virtual 3D yaitu titik (0,0,0) dan tiga sumbu yaitu X, Y, dan Z. *Marker Based Tracking* ini sudah lama dikembangkan sejak 1980-an dan pada awal 1990-an mulai dikembangkan untuk penggunaan *Augmented Reality*.

b. Markless Augmented Reality

Salah satu metode *Augmented Reality* yang saat ini sedang berkembang adalah metode "*Markerless Augmented Reality*", dengan metode ini pengguna tidak perlu lagi menggunakan sebuah *marker* untuk menampilkan elemen-elemen digital, dengan *tool* yang disediakan Qualcomm untuk pengembangan *Augmented Reality* berbasis *mobile device*, mempermudah pengembang untuk membuat aplikasi yang *markerless*.

Seperti yang saat ini dikembangkan oleh perusahaan *Augmented Reality* terbesar di dunia Total Immersion dan Qualcomm, mereka telah membuat berbagai macam teknik *Markerless Tracking* sebagai teknologi andalan mereka, seperti *Face Tracking*, 3D *Object Tracking*, dan *Motion Tracking*.

1. Face Tracking

Algoritma pada komputer terus dikembangkan, hal ini membuat komputer dapat mengenali wajah manusia secara umum dengan cara mengenali posisi mata, hidung, dan mulut manusia, kemudian akan mengabaikan objek-objek lain di sekitarnya seperti pohon, rumah, dan lain - lain.

2. 3D *Object Tracking*

Berbeda dengan *Face Tracking* yang hanya mengenali wajah manusia secara umum, teknik 3D *Object Tracking* dapat mengenali semua bentuk benda yang ada disekitar, seperti mobil, meja, televisi, dan lain-lain.

3. Motion Tracking

Komputer dapat menangkap gerakan, *Motion Tracking* telah mulai digunakan secara ekstensif untuk memproduksi film-film yang mencoba mensimulasikan gerakan.

4. GPS *Based Tracking*

Teknik GPS *Based Tracking* saat ini mulai populer dan banyak dikembangkan pada aplikasi *smartphone* (iPhone dan Android), dengan memanfaatkan fitur GPS dan kompas yang ada didalam *smartphone*, aplikasi akan mengambil data dari GPS dan kompas kemudian menampilkannya dalam bentuk arah yang kita inginkan secara *realtime*, bahkan ada beberapa aplikasi menampikannya dalam bentuk 3D.

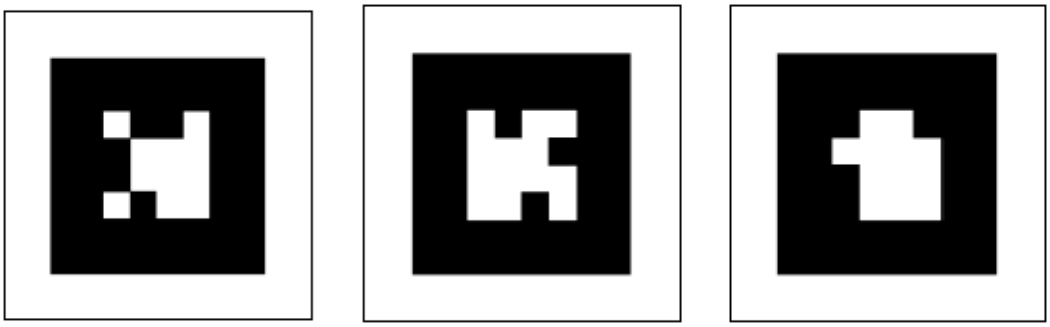
2.2.4. Marker

Marker adalah pola yang dibuat dalam bentuk gambar yang telah dicetak dengan printer yang akan dikenali oleh kamera. *marker* merupakan gambar yang terdiri atas *border outline* dan *pattern image*. *marker* biasanya dengan warna hitam dan putih. Cara pembuatannya pun sederhana tetapi harus diperhatikan ketebalan *marker* yang akan dibuat, ketebalan *marker* jangan kurang dari 25 % dari panjang garis tepi agar pada saat proses deteksi *marker* dapat lebih akurat.

Ciri-ciri yang umum digunakan untuk mengenali satu atau beberapa obyek di dalam citra adalah ukuran, posisi atau lokasi, dan orientasi atau sudut kemiringan obyek terhadap garis acuan yang digunakan. *marker* terdapat dua intensitas warna yaitu warna hitam dan putih atau sering disebut sebagai citra biner. Citra biner memisahkan daerah (*region*) dan latar belakang dengan tegas, walaupun potensi munculnya kekeliruan selalu ada. Kekeliruan di sini adalah kesalahan mengelompokan *pixel* ke dalam golongannya, apakah *pixel* milik suatu daerah dikelompokan sebagai latar belakang atau sebaliknya. Kesalahan seperti ini sering disebut dengan *noise*. Warna putih pada *marker* menunjukan warna sebuah objek, sedangkan warna hitam menunjukan latar belakang. Intensitas warna pada suatu objek memiliki warna yang lebih rendah (gelap), sedangkan latar belakang mempunyai intensitas yang lebih tinggi (terang). Namun pada kenyataannya dapat saja berlaku kebalikannya, yaitu objek mempunyai intensitas

tinggi dan latar belakang mempunyai intensitas rendah. Kombinasi ini biasanya tergantung pada sifat latar belakang pada saat citra tidak tampil terang sekali (putih) atau gelap sekali (hitam), melainkan di antaranya dengan demikian suatu objek yang sama dapat tampil lebih terang atau lebih gelap daripada latar belakangnya dalam citra, tergantung pada gelap atau terangnya warna yang melatarbelakanginya.

Ukuran *marker* yang digunakan dapat mempengaruhi penangkapan pola *marker* oleh kamera. Semakin besar ukuran *marker* semakin jauh jarak yang bisa ditangkap oleh kamera dalam mendeteksi *marker*. Namun disinilah masalahnya, ketika *marker* bergerak menjauhi kamera, jumlah *pixel* pada layar kamera menjadi lebih sedikit dan ini bisa mengakibatkan pendeteksian tidak akurat.



Gambar 2. 4 Contoh *Marker*

2.2.5. Bahasa Pemrograman C#

C# (dibaca: C sharp) merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh *Microsoft* sebagai bagian dari inisiatif kerangka *.NET Framework*. Bahasa pemrograman ini dibuat berbasiskan bahasa C++ yang tellah dipengaruhi oleh aspek-aspek ataupun fitur bahasa yang terdapat pada bahasa-bahasa pemrograman lainnya seperti *Java, Delphi, Visula Basic,* dan lain-lain dengan beberapa penyederhanaan. Menurut standar *ECMA-334 C# Language Specification*, nama C# terdiri atas sebuah huruf Latin **C** (U+0043) yang diikuti oleh tanda pagar yang menandakan angka **#** (U+0023). Tanda pagar # yang digunakan memang bukan tanda kres dalam seni musik (U+266F), dan tanda pagar # (U+0023) tersebut digunakan karena karakter kres dalam seni musik tidak terdapat di dalam *keyboard* standar.

Standar European Computer Manufacturer Association (ECMA) mendaftarkan beberapa tujuan desain dari bahasa pemrograman C#, sebagai berikut:

a. Bahasa pemrograman C# dibuat sebagai bahasa pemrograman yang bersifat *bahasa pemrograman general-purpose* (untuk tujuan jamak), berorientasi objek, modern, dan sederhana.
b. Bahasa pemrograman C# ditujukan untuk digunakan dalam mengembangkan komponen perangkat lunak yang mampu mengambil keuntungan dari lingkungan terdistribusi.
c. Portabilitas *programmer* sangatlah penting, khususnya bagi *programmer* yang telah lama menggunakan bahasa pemrograman C dan C++.
d. Dukungan untuk internasionalisasi (*multi-language*) juga sangat penting.
e. C# ditujukan agar cocok digunakan untuk menulis program aplikasi baik dalam sistem klien-server (*hosted system*) maupun sistem *embedded* (*embedded system*), mulai dari perangkat lunak yang sangat besar yang menggunakan sistem operasi yang canggih hingga kepada perangkat lunak yang sangat kecil yang memiliki fungsi-fungsi terdedikasi.
f. Meskipun aplikasi C# ditujukan agar bersifat 'ekonomis' dalam hal kebutuhan pemrosesan dan memori komputer, bahasa C# tidak ditujukan untuk bersaing secara langsung dengan kinerja dan ukuran perangkat lunak yang dibuat dengan menggunakan bahasa pemrograman C dan bahasa rakitan.
g. Bahasa C# harus mencakup pengecekan jenis (*type checking*) yang kuat, pengecekan larik (*array*), pendeteksian terhadap percobaan terhadap penggunaan Variabel-variabel yang belum diinisialisasikan, portabilitas kode sumber, dan pengumpulan sampah (*garbage collection*) secara otomatis.

2.2.6. Bahasa Pemrograman *Java*

Java adalah bahasa pemrograman tingkat tinggi yang berorientasi objek dan program *Java* tersusun dari bagian yang disebut kelas. Kelas terdiri atas metode-metode yang melakukan pekerjaan dan mengembalikan informasi setelah melakukan tugasnya. Para pemrogram *Java* banyak mengambil keuntungan dari

kumpulan kelas di pustaka kelas *Java*, yang disebut dengan *Java Application Programming Interface* (API). Kelas-kelas ini diorganisasikan menjadi sekelompok yang disebut paket *(package). Java* API telah menyediakan fungsionalitas yang memadai untuk menciptakan *applet* dan aplikasi canggih. Jadi ada dua hal yang harus dipelajari dalam *Java*, yaitu mempelajari bahasa *Java* dan bagaimana mempergunakan kelas pada *Java* API. Kelas merupakan satu-satunya cara menyatakan bagian eksekusi program, tidak ada cara lain. Pada *Java* program *Javac* untuk mengkompilasi file kode sumber *Java* menjadi kelas-kelas *bytecode*. File kode sumber mempunyai ekstensi *.*Java*. Kompilator *Javac* menghasilkan file *bytecode* kelas dengan ekstensi *.*class*. *Interpreter* merupakan modul utama sistem *Java* yang digunakan aplikasi *Java* dan menjalankan program *bytecode Java*.

Beberapa keunggulan *Java* yaitu *Java* merupakan bahasa yang sederhana. *Java* dirancang agar mudah dipelajari dan digunakan secara efektif. *Java* tidak menyediakan fitur-fitur rumit bahasa pemrograman tingkat tinggi, serta banyak pekerjaan pemrograman yang mulanya harus dilakukan manual, sekarang digantikan dikerjakan *Java* secara otomatis seperti dealokasi memori. Bagi pemrogram yang sudah mengenal bahasa C++ akan cepat belajar susunan bahasa *Java* namun harus waspada karena mungkin *Java* mengambil arah (semantiks) yang berbeda dibanding C++.

Java merupakan bahasa berorientasi objek *(OOP)* yaitu cara ampuh dalam pengorganisasian dan pengembangan perangkat lunak. Pada OOP, program komputer sebagai kelompok objek yang saling berinteraksi. Deskripsi ringkas OOP adalah mengorganisasikan program sebagai kumpulan komponen, disebut objek. Objek-objek ini ada secara independen, mempunyai aturan-aturan berkomunikasi dengan objek lain dan untuk memerintahkan objek lain guna meminta informasi tertentu atau meminta objek lain mengerjakan sesuatu. Kelas bertindak sebagai modul sekaligus tipe. Sebagai tipe maka pada saat jalan, program menciptakan objek-objek yang merupakan instan-instan kelas. Kelas dapat mewarisi kelas lain. *Java* tidak mengijinkan pewarisan jamak namun

menyelesaikan kebutuhan pewarisan jamak dengan fasilitas antarmuka yang lebih elegan.

Seluruh objek diprogram harus dideklarasikan lebih dulu sebelum digunakan. Ini merupakan keunggulan *Java* yaitu *Statically Typed*. Pemaksaan ini memungkinkan kompilator *Java* menentukan dan melaporkan terjadinya pertentangan (ketidakkompatibelan) tipe yang merupakan barikade awal untuk mencegah kesalahan yang tidak perlu (seperti mengurangkan variabel bertipe integer dengan variabel bertipe string). Pencegahan sedini mungkin diharapkan menghasilkan program yang bersih. Kebaikan lain fitur ini adalah kode program lebih dapat dioptimasi untuk menghasilkan program berkinerja tinggi.

Java menggunakan model pengamanan tiga lapis *(three-layer security model)* untuk melindungi sistem dari *untrusted Java code.* Pertama, *bytecode verifier* membaca *bytecode* sebelum dijalankan dan menjamin *bytecode* memenuhi aturan-aturan dasar bahasa *Java.* Kedua, *class loader* menangani pemuatan kelas *Java* ke *runtime interpreter*. Ketiga, manajer keamanan menangani keamanan tingkat aplikasi dengan mengendalikan apakah program berhak mengakses sumber daya seperti sistem *file*, *port* jaringan, proses eksternal dan sistem *window*.

Platform independence adalah kemampuan program bekerja di sistem operasi yang berbeda. Bahasa *Java* merupakan bahasa yang secara sempurna tidak bergantung *platform*. Tipe variabel *Java* mempunyai ukuran sama di semua *platform* sehingga variabel bertipe integer berukuran sama tidak peduli dimana program *Java* dikompilasi. Begitu telah tercipta *file .class* dengan menggunakan kompilator *Java* di *platform* manapun, maka *file .class* tersebut dapat dijalankan di *platform* manapun. Jadi “dimanapun dibuat, dimanapun dapat dijalankan”. Slogan ini biasa diringkas sebagai *Write Once, Run Anywhere (WORA).*

Java termasuk bahasa *Multithreading*. *Thread* adalah untuk menyatakan program komputer melakukan lebih dari satu tugas di satu waktu yang sama. *Java* menyediakan kelas untuk menulis program *multithreaded*, program mempunyai lebih dari satu *thread* eksekusi pada saat yang sama sehingga memungkinkan program menangani beberapa tugas secara konkuren.

Program *Java* melakukan *garbage collection* yang berarti program tidak perlu menghapus sendiri objek-objek yang tidak digunakan lagi. Fasilitas ini mengurangi beban pengelolaan memori oleh pemrogram dan mengurangi atau mengeliminasi sumber kesalahan terbesar yang terdapat di bahasa yang memungkinkan alokasi dinamis.

Java mempunyai mekanisme *exception-handling* yang ampuh. *Exception-handling* menyediakan cara untuk memisahkan antara bagian penanganan kesalahan dengan bagian kode normal sehingga menuntun ke struktur kode program yang lebih bersih dan menjadikan aplikasi lebih tegar. Ketika kesalahan yang serius ditemukan, program *Java* menciptakan *exception*. *Exception* dapat ditangkap dan dikelola program tanpa resiko membuat sistem menjadi turun.

Program *Java* mendukung *native method* yaitu fungsi ditulis di bahasa lain, biasanya C/C++. Dukungan *native method* memungkinkan pemrogram menulis fungsi yang dapat dieksekusi lebih cepat dibanding fungsi ekivalen di *Java*. *Native method* secara dinamis akan di-*link* ke program *Java*, yaitu diasosiasikan dengan program saat berjalan.

Selain itu keuntungan menggunakan bahasa pemrograman *Java* antara lain memori pada *Java* secara otomatis dilengkapi *garbage collector* yang berfungsi mendealokasi memori yang tidak diperlukan. Tidak ada lagi upaya yang dilakukan pemrogram untuk melakukan *dispose*(). Kita tidak lagi dibebani urusan korupsi memori. *Java* menerapkan array sebenarnya, menghilangkan keperluan aritmatika pointer yang berbahaya dan mudah menjadi salah. Menghilangkan pewarisan jamak *(multiple inheritance)* diganti fasilitas antarmuka. Dan mudah dijalankan diberbagai *platform*.

Grafical User Interface (GUI) adalah salah satu kemampuan *Java* dalam mendukung dan manajemen antarmuka berbasis grafis. Tampilan grafis yang akan ditampilkan terhubung dengan program serta tempat penyimpanan data. Elemen dasar di *Java* untuk penciptan tampilan berbasis grafis adalah dua paket yaitu AWT dan *Swing*. *Abstract Windowing Toolkit* (AWT), atau disebut juga *"Another Windowing Toolkit"*, adalah pustaka *windowing* bertujuan umum dan *multiplatform* serta menyediakan sejumlah kelas untuk membuat GUI di *Java*.

Dengan AWT, dapat membuat window, menggambar, manipulasi gambar, dan komponen seperti *Button, Scrollbar, Checkbox, TextField,* dan menu *pull-down*.

Penggunaan komponen AWT ditandai dengan adanya instruksi : *import Java.awt.*; Swing* merupakan perbaikan kelemahan di AWT. Banyak kelas *swing* menyediakan komponen alternatif terhadap AWT. Contohnya kelas *JButton swing* menyediakan fungsionalitas lebih banyak dibanding kelas *Button*. Selain itu komponen *swing* umumnya diawali dengan huruf "J", misalnya *JButton, JTextField, JFrame, JLabel, JTextArea, JPanel,* dan sebagainya. Teknologi *swing* menggunakan dan memperluas gagasan-gagasan AWT. Sementara, penggunaan komponen *Swing* ditandai dengan adanya instruksi : *import Javax.swing.*;*

Beberapa perbedaan AWT dan *Swing*, AWT merupakan komponen *heavyweight* (kelas berat) sedangkan *Swing lightweight* (kelas ringan). *Swing* memiliki lebih banyak komponen. Fasilitas *Swing Look and Feel* : *Metal, Windows,* Motif. Komponen *Swing* berdasar *model-view*, yaitu suatu cara pengembangan komponen dengan pemisahan penyimpanan dan penanganan data dari representasi visual data.

Bahasa pemrograman *Java* merupakan salah satu bahasa pemrograman yang umum digunakan untuk mengembangkan aplikasi basis data yang dibuat menggunakan MySQL.

2.2.7. Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung[2]. UML adalah metodologi untuk mengembangkan sistem OOP dan sekelompok perangkat *tool* untuk mendukung pengembangan sistem tersebut. UML mulai diperkenalkan oleh *Object Management Group,* sebuah organisasi yang telah mengembangkan model, teknologi dan standar OOP sejak tahun 1980-an. Sekarang UML sudah mulai banyak digunakan oleh para praktisi OOP. UML merupakan dasar bagi perangkat (*tool*) desain berorientasi objek dari IBM.

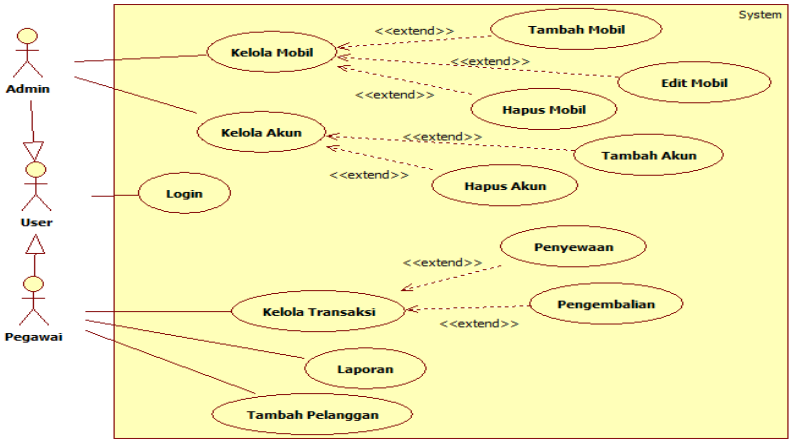
UML adalah suatu bahasa yang digunakan untuk menentukan, memvisualisasikan, membangun, dan mendokumentasikan suatu sistem

informasi. UML dikembangkan sebagai suatu alat untuk analisis dan desain berorientasi objek oleh Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. Namun demikian UML dapat digunakan untuk memahami dan mendokumentasikan setiap sistem informasi. Penggunaan UML dalam industri terus meningkat. Ini merupakan standar terbuka yang menjadikannya sebagai bahasa pemodelan yang umum dalam industri peranti lunak dan pengembangan sistem.

2.2.7.1. Diagram Unified Modeling Language

UML terdiri atas pengelompokan diagram-diagram sistem menurut aspek dan sudut pandang tertentu. Diagram merupakan gambaran permasalahan maupun solusi dari permasalahan suatu model. UML mempunyai 9 diagram, yaitu :

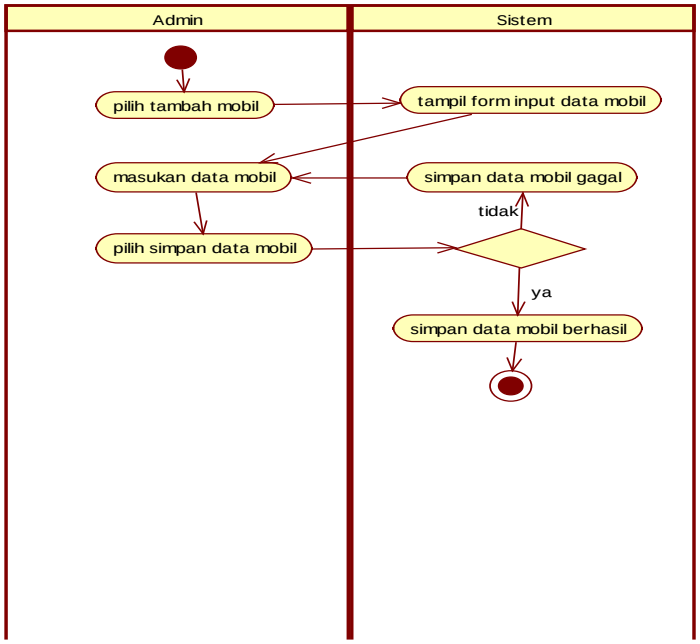
a. ***Use case* diagram** digunakan untuk memodelkan bisnis proses berdasarkan perspektif pengguna sistem. Use case diagram terdiri atas diagram untuk use case dan *actor*. *Actor* merepresentasikan orang yang akan mengoperasikan atau orang yang berinteraksi dengan sistem aplikasi. *Use case* merepresentasikan operasi-operasi yang dilakukan oleh *actor*. *Use case* digambarkan berbentuk elips dengan nama operasi dituliskan di dalamnya. *Actor* yang melakukan operasi dihubungkan dengan garis lurus ke *use case*.



Gambar 2. 5 Contoh Use Case Diagram

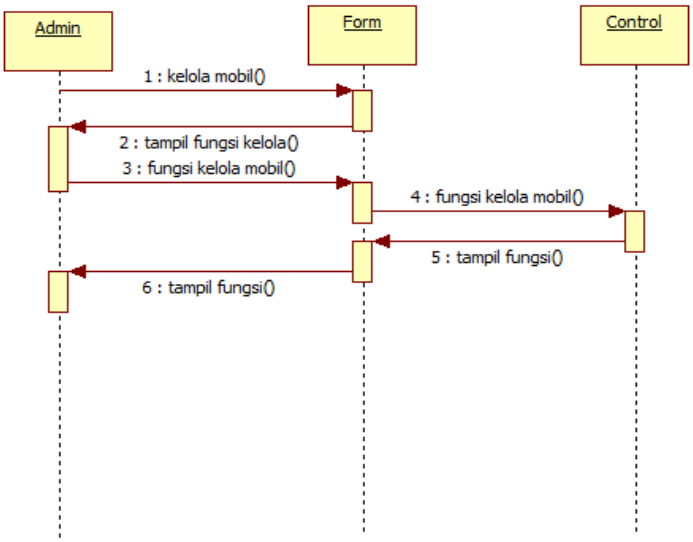
b. ***Activity* diagram** menggambarkan aliran fungsionalitas sistem. Pada tahap pemodelan bisnis, diagrama aktivitas dapat digunakan untuk menujukkan

aliran kerja bisnis *(business work flow)*. Dapat juga digunakan untuk menggambarkan aliran kejadian *(flow of event)* dalam *use case*.



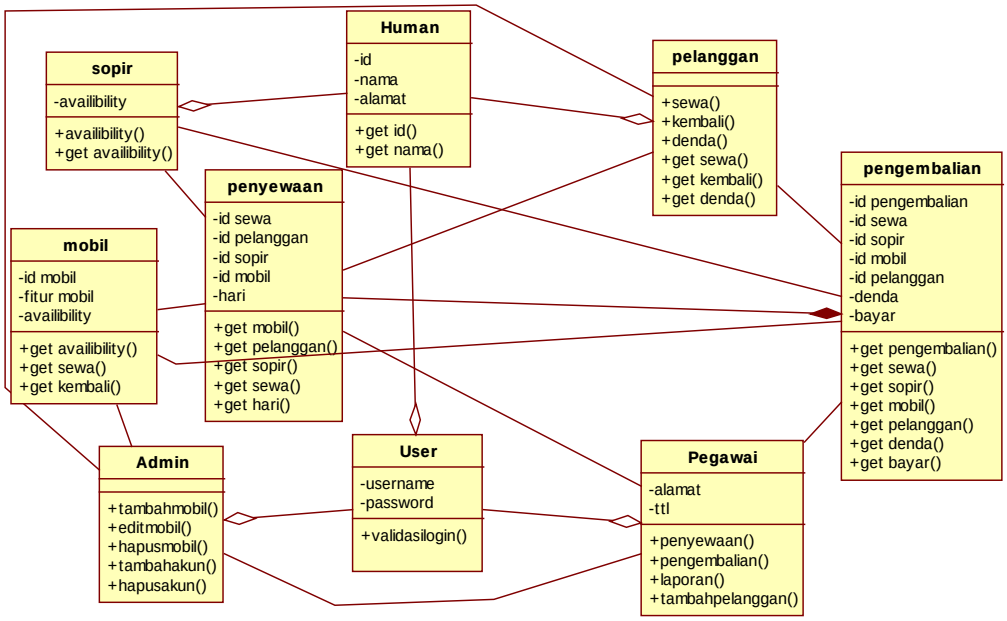
Gambar 2. 6 Contoh Activity Diagram

c. ***Sequence* diagram** digunakan untuk menunjukkan aliran fungsionalitas dalam *use case*. Diagram sekuensial adalah diagram yang disusun berdasarkan urutan waktu. Kita membaca diagram sekuensial dari atas ke bawah. Setiap diagram sekuensial mempresentasikan suatu aliran dari beberapa aliran di dalam *use case*. Kita dapat membaca diagram ini dengan memperhatikan obyek-obyek dan pesan-pesan yang ada di diagram. Obyek yang terlibat dalam aliran ditunjukkan dengan bujur sangkar yang ada di atas diagram.



Gambar 2. 7 Contoh Sequence Diagram

d. ***Class* diagram** menggambarkan struktur statis *class* di dalam sistem. *class* merepresentasikan sesuatu yang ditangani oleh sistem. *class* dapat berhubungan dengan yang lain melalui berbagai cara: *associated* (terhubung satu sama lain), *dependent* (satu *class* tergantung/menggunakan *class* yang lain), *specialed* (satu *class* merupakan spesialisasi dari *class* lainnya), atau *package* (*group* bersama sebagai satu unit). sebuah sistem biasanya mempunyai beberapa *class* diagram.



Gambar 2. 8 Contoh Class Diagram

e. ***Collaboration* diagram** dipakai untuk memodelkan interaksi antar objek di dalam sistem. Berbeda dengan *sequence* diagram yang lebih menonjolkan kronologis dari operasi-operasi yang dilakukan, *collaboration* diagram lebih fokus pada pemahaman atas keseluruhan operasi yang dilakukan oleh objek.

f. ***State* Diagram**, yang memperliatkan semua keadaan (*state*) yang dapat dimiliki oleh kelas dan *event* yang dapat merubah keadaan tersebut.

g. ***Component* Diagram**, yang memperlihatkan struktur fisik dari *source code* dalam *terminology code components*. Komponen berisi informasi tentang logical *class* dapat berupa komponen *source code*, komponen biner atau komponen yang dapat dieksekusi.

h. ***Deployment* Diagram**, yang memperlihatkan arsitektur fisik dari *hardware* dan *software* pada sistem.

i. ***Object* Diagram**, yang merupakan varian dari kelas diagram yang memperlihatkan lebih detail banyaknya obyek yang mengintantiasi (*instances*) kelas.

2.2.8. Pemrograman Berorientasi Objek

Pemrograman berorientasi objek merupakan paradigma pemrograman yang berorientasikan kepada objek. Semua data dan fungsi di dalam paradigma ini dibungkus dalam *kelas-kelas* atau *objek-objek*. Bandingkan dengan logika pemrograman terstruktur. Setiap objek dapat menerima pesan, memproses data, dan mengirim pesan ke objek lainnya,

Model data berorientasi objek dikatakan dapat memberi fleksibilitas yang lebih, kemudahan mengubah program, dan digunakan luas dalam teknik piranti lunak skala besar. Lebih jauh lagi, pendukung OOP mengklaim bahwa OOP lebih mudah dipelajari bagi pemula dibanding dengan pendekatan sebelumnya, dan pendekatan OOP lebih mudah dikembangkan dan dirawat.

Berikut konsep dasar Pemrograman Berorientasi Objek, yaitu :

a. Kelas adalah kumpulan atas definisi data dan fungsi-fungsi dalam suatu unit untuk suatu tujuan tertentu. Sebagai contoh *'class of dog'* adalah suatu unit yang terdiri atas definisi-definisi data dan fungsi-fungsi yang menunjuk pada berbagai macam perilaku/turunan dari anjing. Sebuah class adalah dasar dari modularitas dan struktur dalam pemrograman berorientasi objek. Sebuah class secara tipikal sebaiknya dapat dikenali oleh seorang non-programmer sekalipun terkait dengan domain permasalahan yang ada, dan kode yang terdapat dalam sebuah class sebaiknya (relatif) bersifat mandiri dan independen.
b. Objek adalah membungkus data dan fungsi bersama menjadi suatu unit dalam sebuah program komputer; objek merupakan dasar dari modularitas dan struktur dalam sebuah program komputer berorientasi objek.
c. Abstraksi adalah kemampuan sebuah program untuk melewati aspek informasi yang diproses olehnya, yaitu kemampuan untuk memfokus pada inti. Setiap objek dalam sistem melayani sebagai model dari "pelaku" abstrak yang dapat melakukan kerja, laporan dan perubahan keadaannya, dan berkomunikasi dengan objek lainnya dalam sistem, tanpa mengungkapkan bagaimana kelebihan ini diterapkan. Proses, fungsi atau

metode dapat juga dibuat abstrak, dan beberapa teknik digunakan untuk mengembangkan sebuah pengabstrakan.

d. Enkapsulasi memastikan pengguna sebuah objek tidak dapat mengganti keadaan dalam dari sebuah objek dengan cara yang tidak layak; hanya metode dalam objek tersebut yang diberi izin untuk mengakses keadaannya. Setiap objek mengakses interface yang menyebutkan bagaimana objek lainnya dapat berinteraksi dengannya. Objek lainnya tidak akan mengetahui dan tergantung kepada representasi dalam objek tersebut.
e. Polimorfisme tidak bergantung kepada pemanggilan subrutin, bahasa orientasi objek dapat mengirim pesan; metode tertentu yang berhubungan dengan sebuah pengiriman pesan tergantung kepada objek tertentu di mana pesa tersebut dikirim. Contohnya, bila sebuah burung menerima pesan "gerak cepat", dia akan menggerakan sayapnya dan terbang. Bila seekor singa menerima pesan yang sama, dia akan menggerakkan kakinya dan berlari. Keduanya menjawab sebuah pesan yang sama, namun yang sesuai dengan kemampuan hewan tersebut. Ini disebut polimorfisme karena sebuah variabel tungal dalam program dapat memegang berbagai jenis objek yang berbeda selagi program berjalan, dan teks program yang sama dapat memanggil beberapa metode yang berbeda di saat yang berbeda dalam pemanggilan yang sama.

2.2.9. Kuesioner

Salah satu instrumen pengumpul data dalam penelitian adalah kuesioner, atau disebut juga daftar pertanyaan (terstruktur). Kuesioner ini biasanya berkaitan erat dengan masalah penelitian, atau juga hipotesis penelitian yang dirumuskan.

Sebelum mebuat kuesioner, ada baiknya peneliti mengantisipasi kemungkinan adanya kesalahan yang sering terjadi berkaitan dengan pelaksanaan pengumpulan data dari responden. Beberapa permasalahan yang mungkin dan bahkan sering terjadi dan bagaimana cara memperbaikinya sebagai berikut :

a. Responden sering menganggap wawancara tidak masuk akal dan bahkan sering menganggapnya sebagai dalih untuk tujuan-tujuan tertentu misalnya komersial. Alternatif pemecahannya antara lain adalah menyampaikannya dalam pengantar bahwa penelitian yang akan dilakukan benar-benar untuk tujuan nonkomersial. Tentu saja dengan kata-kata yang baik dan sopan.
b. Responden merasa terganggu dengan adanya informasi yang dirasa menyerang dirinya atau kepentingannya, misalnya takut dirilis di media massa. Pemecahannya adalah menghindari pertanyaan yang sensitif, serta diyakinkan bahwa tidak akan ada nama responden di dalamnya.
c. Responden menolak bekerja sama atas dasar pengalaman masa lalu. Upayakan untuk meyakinkan responden bahwa ini beda, beri pengertian bahwa responden dalam hal ini turut berjasa dalam membantu penelitian ini.
d. Responden yang tergolong dirinya kelompok minoritas sehingga merasa lelah karena sering dijadikan kelinci percobaan. Namun jika hal seperti ini terjadi, peneliti bisa menggunakan instrumen lain, atau bahkan mencari sumber data yang lain.
e. Responden orang ‘penting’ dan sering merasa tahu akan apa yang akan ditelitinya. Cara pemecahannya adalah dengan metode menyanjung orang penting tadi, misalnya dengan mengatakan bahwa hanya dialah orang satu-satunya yang bisa memberikan informasi tentang masalah ini.
f. Responden menjawab dengan pertimbangan normatif, berpikir baik atau jelek. Katakan kepadanya bahwa penelitian ini semata-mata untuk pengembangan ilmu, dan bukan untuk kepentingan lain. Selain itu nama responden juta tidak perlu dicantumkan.
g. Responden merasa takut akan ‘kebodohannya’ dalam menjawab pertanyaan ini. Katakan kepadanya bahwa jawaban apapun dari responden itu penting, dan tidak ada yang salah dalam menjawab.
h. Responden mengatakan tidak ada waktu untuk menjawabnya, atau merasa itu bukan bidang minatnya. Pemecahannya adalah mengatakan bahwa dialah satu-satunya orang yang bisa memberikan informasi yang diperlukan dalam penelitian ini.

Kegagalan dalam membuat kuesioner :

a. Luncuran pertanyaan ganda: Jangan menanyakan satu masalah dalam satu pertanyaan. Contoh, apakah anda sering menyobek buku di perpustakaan selagi tidak ada pengawas yang melihatnya; dan apakah anda juga sering mencoreti buku milik perpustakaan untuk kepentingan penjelasan secara khusus?.
b. Pertanyaan yang mengaahkan: Hindari bentuk pertanyaan seperti ini. Contoh, menurut presiden, kita harus mengencangkan ikat pinggang dalam menghadapi krisis ekonomi yang berkepanjangan ini. Anda setuju, bukan?. Pertanyaan seperti ini biasanya dijawab secara langsung dengan kata ‘setuju’. Bisa dibayangkan bahwa jika semua pertanyaan dijawab dengan setuju.
c. Pertanyaan sensitif: Hati-hati dengan pertanyaan sensitif seperti contoh berikut: Anda pernah melakukan onani?; Anda pernah melakukan hubungan seks sebelum nikah?. Pertanyaan jenis ini termasuk kategori sensitif, bahkan kurang ajar.
d. Pertanyaan yang menakut-nakuti: Contoh. Di daerah ini sering terjadi perampokan dan penodongan di malam hari. Bisa Anda sebutkan orangnya?; atau, Anda tentu mengetahui peristiwa pembunuhan yang terjadi beberapa waktu lalu di daerah ini, karena andalah yang paling dekat dengan tempat kejadian perkara (TKP). Kami datang untuk menyelidikinya, oleh karena itu tolong jawab dengan sejujurnya pertanyaan-pertanyaan kami.

2.2.9.1. Kuesioner Tertutup dan Terbuka

Ada dua jenis pertanyaan dalam kuesioner, yakni pertanyaan terbuka, tertutup dan gabungan tertutup dan terbuka. Pertanyaan dengan jawaban terbuka adalah pertanyaan yang memberikan kebebasan penuh kepada responden untuk menjawabnya. Di sini peneliti tidak memberikan satupun alternatif jawaban. Sedangkan pertanyaan dengan jawaban tertutup adalah sebaliknya, yaitu semua alternatif jawaban responden sudah disediakan oleh peneliti. Responden tinggal memilih alternatif jawaban yang dianggapnya sesuai.

Kuesioner dengan jawaban tertutup, salah satu keuntungannya untuk kuesioner ini adalah sebagai berikut :

a. Jawaban-jawaban bersifat standar dan bisa dibandingkan dengan jawaban orang lain.
b. Jawaban-jawabannya jauh lebih mudah dikoding dan dianalisis, bahkan sering secara langsung dapat dikoding dari pertanyaan yang ada, sehingga hal ini dapat menghemat tenaga dan waktu.
c. Responden lebih merasa yakin akan jawaban-jawabannya, terutama bagi mereka yang sebelumnya tidak yakin.
d. Jawaban-jawaban relatif lebih lengkap karena sudah dipersiapkan sebelumnya oleh peneliti.
e. Analisis dan formulasinya lebih mudah jika dibandingkan dengan model kuesioner dengan jawaban terbuka.

Meskipun demikian, ada juga kelemahannya, yakni :

a. Sangat mudah bagi responden untuk menebak setiap jawaban, meskipun sebetulnya mereka tidak memahami masalahnya.
b. Responden merasa frustrasi dengan sediaan jawaban yang tidak satu pun yang sesuai dengan keinginannya.
c. Sering terjadi jawaban-jawaban yang terlalu banyak sehingga membingungkan responden untuk memilihnya.
d. Tidak bisa mendeteksi adanya perbedaan pendapat antara responden dengan peneliti karena responden hanya disuruh memilih alternatif jawaban yang tersedia.

Kuesioner dengan jawaban terbuka, keuntungannya antara lain adalah :

a. Dapat digunakan manakala semua alternatif jawaban tidak diketahui oleh peneliti, atau manakala peneliti ingin melihat bagaimana dan mengapa jawaban responden serta alasan-alasannya. Hal ini sangat baik untuk menambah pengetahuan peneliti akan masalah yang diutarakannya.
b. Membolehkan responden untuk menjawab sedetil atau serinci mungkin atas apa yang ditanyakan peneliti. Dalam hal ini pendapat responden dapat diketahui dengan baik oleh peneliti.

Kuesioner dengan jawaban tertutup dan terbuka (gabungan), untuk menjembatani kekurangan-kekurangan seperti tadi, maka sering digunakan

pertanyaan model gabungan antara keduanya. Dengan model tertutup dan tebuka, semua kekurangan seperti tadi bisa diatasi. Misalnya dalam satu pertanyaan, disamping disediakan alternatif jawaban oleh peneliti, juga perlu disediakan alternatif terbuka (c. ……) untuk diisi sendiri oleh responden sesuai dengan pendapatnya secara bebas. Dalam mengolah data untuk model terakhir ini, bisa dilakukan pengelompokan ulang atas semua jawaban responden pada alternatif terbuka tadi. Atau bisa juga peneliti melihat ulang apakah jawaban responden yang terakhir itu sebenarnya sudah termasuk ke dalam salah satu alternatif jawaban yang tersedia. Dan jika ternyata jawabannya sama dengan salah satu alternatif jawaban yang tersedia namun dalam bahasa yang berbeda, peneliti bisa menganggapnya sebagai jawaban seperti pada alternatif yang tersedia tadi.

Contoh sebuah pertanyaan sederhana dengan alternatif jawabannya, tujuan Anda berkunjung ke perpustakaan adalah :

1. Mengerjakan tugas-tugas akademik.
2. Mencari informasi akademik untuk kepentingan tugas dari dosen.
3. Menambah wawasan.
4. …… menambah pengetahuan. (Responden menjawab dengan tulisan sendiri pada alternatif yang terbuka ini).

Kita bisa melihat bahwa sebenarnya jawaban responden tersebut sama atau hampir sama dengan alternatif nomor (3) menambah wawasan.

2.2.9.2. Susunan Pertanyaan

Ada aturan umum dalam menyusun urutan pertanyaan yang dibuat, meskipun tidak mutlak, yakni sebagai berikut :

a. Pertanyaan sensitif dan pertanyaan model jawaban terbuka sebaiknya ditempatkan di bagian akhir kuesioner.
b. Pertanyaan-pertanyaan yang mudah sebaiknya ditempatkan pada bagian awal kuesioner.
c. Susunlah pertanyaan dengan pola susunan yang saling berkaitan satu sama lain secara logis.
d. Susunlah pertanyaan sesuai dengan susunan yang logis, runtut, dan tidak meloncat-loncat dari tema satu ke tema yang lain.

e. Jangan gunakan pasangan pertanyaan yang mengecek reliabilitas. Misalnya, setujukah Anda terhadap aborsi? Sementara itu di tempat lain, ada pertanyaan, tidak setujukan Anda terhadap aborsi?.
f. Gunakan pertanyaan secara singkat dan jelas, tidak bertele-tele.

2.2.10. Skala Likert

Skala Likert adalah suatu skala psikometrik yang umum digunakan dalam angket dan merupakan skala yang paling banyak digunakan dalam riset berupa survei. Skala Likert digunakan untuk mengukur sikap, pendapat dan persepsi seseorang atau sekelompok orang tentang fenomena sosial[7].Nama skala ini diambil dari nama Rensis Likert, yang menerbitkan suatu laporan yang menjelaskan penggunaannya. Sewaktu menanggapi pertanyaan dalam skala Likert, responden menentukan tingkat persetujuan mereka terhadap suatu pernyataan dengan memilih salah satu dari pilihan yang tersedia. Biasanya disediakan lima pilihan skala dengan format seperti ini:

1. Sangat tidak setuju
2. Tidak setuju
3. kurang setuju
4. Setuju
5. Sangat setuju

Selain pilihan dengan lima skala seperti contoh di atas, kadang digunakan juga skala dengan tujuh atau sembilan tingkat. Suatu studi empiris menemukan bahwa beberapa karakteristik statistik hasil kuesioner dengan berbagai jumlah pilihan tersebut ternyata sangat mirip.

Skala Likert merupakan metode skala bipolar yang mengukur baik tanggapan positif ataupun negatif terhadap suatu pernyataan. Empat skala pilihan juga kadang digunakan untuk kuesioner skala Likert yang memaksa orang memilih salah satu kutub karena pilihan "netral" tak tersedia.

2.3. Software Pendukung

Berikut ini akan dijelaskan mengenai *software-software* pendukung yang digunakan untuk membangun aplikasi yang akan dibuat. *Software* yang digunakan

antara lain seperti Unity 3D, NyARToolkit, Blender, Adobe Photoshop, Android SDK, dan Powtoon.

2.3.1. Unity 3D

Unity adalah sebuah *game engine* yang memungkinkan anda, baik perseorangan mau pun tim, untuk membuat sebuah *game* 3d dengan mudah dan cepat[3]. *Unity* itu bagus dan penuh perpaduan dengan aplikasi yang profesional. Editor pada *Unity* dibuat dengan *user interface* yang sederhana. *Editor* ini dibuat setelah ribuan jam yang mana telah dihabiskan untuk membuatnya menjadi nomor satu dalam urutan rankking teratas untuk *editor game*. Grafis pada *unity* dibuat dengan grafis tingkat tinggi untuk OpenGL dan directX. *Unity* mendukung semua format file, terutamanya format umum seperti semua format dari *art applications*. *Unity* cocok dengan versi 64-bit dan dapat beroperasi pada *Mac OS x* dan *windows* dan dapat menghasilkan *game* untuk *Mac, Windows, Wii, iPhone, iPad* dan *Android*.

Unity secara rinci dapat digunakan untuk membuat *video game 3D, real time* animasi 3D dan visualisasi arsitektur dan isi serupa yang interaktif lainnya. *Editor Unity* dapat menggunakan *plugin* untuk *web player* dan menghasilkan *game browser* yang didukung oleh *Windows* dan *Mac*. *Plugin web player* dapat juga dipakai untuk *widgets Mac*. *Unity* juga akan mendukung *console* terbaru seperti *PlayStation 3* dan *Xbox 360*. Pada tahun 2010, telah memperoleh *Technology Innovation Award* yang diberikan oleh *Wall Street Journal* dan tahun 2009, *Unity Technology* menjadi 5 perusahaan *game* terbesar. Tahun 2006, menjadi juara dua pada *Apple Design Awards*.

Server aset dari *Unity* dapat digunakan semua scripts dan aset game sebagai solusi dari versi kontrol dan dapat mendukung proyek yang terdiri atas banyak *gigabytes* dan ribuan dari file *multi-megabyte. Editor Unity* dapat menyimpan metadata dan versi mereka, itu dapat berjalan , pembaharuan dan didalam perbandingan versi grafis. *Editor Unity* dapat diperbaharui dengan sesegera mungkin seperti file yang telah dimodifikasi. *Server* aset *Unity* juga

cocok pada *Mac, Windows* dan *Linux* dan juga berjalan pada PostgreSQL, database server *opensource*.

Perizinan atau *license* dari *Unity* ada dua bentuk. Ada *Unity* dan *Unity Pro*. Versi *Unity* tersedia dalam bentuk gratis, sedang versi *Unity Pro* hanya dapat dibeli. Versi *Unity Pro* ada dengan fitur bawaan seperti efek *post processing* dan *render efek texture*. Versi *Unity* merupakan yang gratis memperlihatkan aliran untuk *game web* dan layar *splash* untuk *game* yang berdiri sendiri. *Unity* dan *Unity Pro* menyediakan tutorial, isi, contoh projek, wiki, dukungan melalui forum dan perbaruan kedepannya. *Unity* digunakan pada iPhone, iPod dan iPad operating system yang mana iOS ada sebagai *add-ons* pada *Unity editor* yang telah ada lisensinya, dengan cara yang sama juga pada Android.

2.3.2. NyARToolkit

NyarToolkit adalah sebuah *library* yang dibuat untuk mempermudah pembuatan aplikasi *augmented reality*. *Toolkit* ini berupa *library* yang didalamnya telah terdapat parameter-parameter standar seperti *camera calibration*, pembuatan nyar dan *papervision*, pembuatan bitmap data, dan sebagainya. kelebihan NyAR4psg dibandingkan dengan *library* yang lain adalah:

1. *Library* NyAR4psg menyediakan lisensi gratis.
2. Pengembang sangat aktif
3. Aplikasi yang dibuat dengan menggunakan NyAR4psg dengan *processing* ini bisa di jalankan di OS (Windows,Mac, dan Linux) mana saja, dan berbahasa *Java* yang dapat juga berekstensi *.*Java*, sehingga dapat ditampilkan di halaman *web*. Pengguna dapat menggunakan aplikasi yang dibangun tanpa harus menginstal aplikasi ini terlebih dahulu.
4. *Library* NyAR4psg mendukung format .OBJ yang dapat dibuat di Software 3D yang sering digunakan untuk AR yaitu 3DSMax , Blender, dll.
5. *Library* NyAR4psg menggunakan aplikasi Processing yang berbahasa *Java* ini lebih ringan daripada menggunakan aplikasi *Java* lainnya seperti eclipse.

Adapun beberapa kekurangan yang dimiliki oleh *library* NyARToolkit adalah sebagai berikut:

1. Pengguna *library* NyAR4psg menggunakan *Java* oleh sebab itu untuk dapat menggunakan *library* ini pengguna dituntut untuk dapat memahami bahasa pemograman *Java*.
2. Desain gambar yang akan dijadikan sebagai marker dikonversi terlebih dahulu ke ekstensi *.pat.

Aplikasi NyarToolkit dapat memungkinkan pencitraan *virtual* secara *live* ke dalam dunia nyata. Rahasianya ada pada kotak hitam yang digunakan sebagai penanda pelacakan (marker). Pelacakan NyarrToolkit bekerja sebagai berikut:

1. Kamera menangkap data dari *marker* dalam dunia nyata dan mengirimkan informasinya ke komputer.
2. *Software* pada komputer akan melacak bentuk kotak dari *marker* dan mendeteksi berapa *video frame*-nya.
3. Bila kotak telah ditemukan, maka *software* menggunakan perhitungan matematis untuk menghitung posisi dari kamera *relative* terhadap kotak hitam pada *marker*.
4. Setelah dikalkulasi maka model grafis akan dimunculkan pada posisi yang sama dan berada di dalam lingkup kotak hitam, lalu ditampilkan ke layar untuk melihat grafis dalam dunia nyata.

2.3.3. Blender

Blender merupakan software pembuat objek 3 dimensi yang mampu untuk membuat model dan animasi[5]. Perangkat lunak ini digunakan untuk membuat film animasi, efek visual, model cetak 3D, aplikasi 3D interaktif dan permainan video. Blender memiliki beberapa fitur termasuk pemodelan 3D, pentekstur an, penyunting gambar bitmap, penulangan, simulasi cairan dan asap, simulasi partikel, animasi, penyunting video, pemahat digital, dan *rendering*.

2.3.4. Adobe Photoshop

Adobe Photoshop adalah perangkat lunak editor citra buatan Adobe Systems yang dikhususkan untuk pengeditan foto/gambar dan pembuatan efek. Perangkat lunak ini banyak digunakan oleh fotografer digital dan perusahaan iklan

sehingga dianggap sebagai pemimpin pasar (*market leader*) untuk perangkat lunak pengolah gambar/foto.

2.3.5. Android SDK

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 *pluginAndroid 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 (misalnya, *reboot*, menginstal paket perangkat lunak dengan jarak jauh).

Android SDK telah dirilis pada tanggal 12 November 2007. Dan pada tanggal 15 Juli 2008 tim *Android Developer Challenge* sengaja mengirimkan *email* ke semua pendatang di *Android Developer Challenge* untuk mengumumkan bahwa rilis SDK terbaru telah tersedia pada halaman *download* pribadi. Email tersebut juga ditujukan kepada pemenang *Android Developer Challenge* putaran pertama. Sebuah penyataan bahwa Google telah menyediakan rilis SDK terbaru untuk beberapa pengembang dan bukan untuk orang lain.

Pada tanggal 18 Agustus 2008, Android SDK 0.9 *beta* dirilis. Rilis ini menyediakan API yang diperbarui dan diperluas, perbaikan pada alat-alat pengembangan dan desain terbaru untuk layar awal. Petunjuk untuk meng-*upgrade* SDK sudah tersedia pada rilis sebelumnya. Pada tanggal 23 September 2008, Android 1.0 SDK telah dirilis. Pada tanggal 9 Maret 2009, Google merilis versi 1.1 untuk telepon seluler Android. Rilis terbaru tersebut termasuk dukungan

untuk pencarian dengan suara, harga aplikasi, perbaikan jam *alarm*, perbaikan pengiriman gmail, perbaikan surat pemberitahuan dan peta.

Pada pertengahan Mei 2009, Google merilis versi 1.5 (Cupcake) pada sistem operasi Android dan SDK. Pembaruan ini termasuk banyak fitur baru seperti perekaman video, dukungan untuk *bluetooth*, sistem *keyboard* pada layar dan pengenalan suara. Rilis ini juga membuka *AppWidget framework* kepada para pengembang yang memungkinkan orang untuk membuat *widget* sendiri pada halaman *home*. Pada September 2009 versi 1.6 (Donut) dirilis yang menampilkan hasil pencarian yang lebih baik dan penggunaan indikator baterai.

Aplikasi Android dipaketkan ke dalam format .apk dan disimpan pada *folder*/data/app. Pengguna dapat menjalankan perintah adb root untuk mengakses *folder*tersebut karena *root* memiliki izin untuk mengakses folder tersebut.

2.3.6. Powtoon

Powtoon Ltd adalah software animasi berbasis Web yang memungkinkan pengguna untuk membuat presentasi animasi dengan memanipulasi objek sebelum dibuat, impor gambar, tersedia musik dan pengguna membuat voice-overs. Powtoon Ltd menggunakan mesin Adobe Flex untuk menghasilkan file XML yang dapat bermain di Powtoon Ltd tampilan online, diekspor ke YouTube atau download sebagai file MP4. Powtoon Ltd juga tersedia di Google Chrome Store dan memiliki aplikasi pada Edmodo.com. Powtoon Ltd adalah perusahaan yang menjual software berbasis cloud untuk membuat presentasi animasi dan video. Powtoon Ltd didirikan pada Januari 2012. Perusahaan ini merilis versi beta pada bulan Agustus 2012 dan sudah mendapat banyak pelanggan. Pada Desember 2012 Powtoon Ltd berinvestasi dari perusahaan Venture Capital LA berbasis Startup Minds. Pada Februari 2013 Powtoon Ltd memperkenalkan opsi akun gratis yang memungkinkan pengguna untuk membuat animasi video yang dapat diekspor ke YouTube.