BAB 2
LANDASAN TEORI

2.1 Tinjauan Perusahaan

Studio Gembira *Photo* adalah salah satu studio foto terbesar di kota Bandung. Dengan pelayanan jasa fotografi yang beragam dan kualitas hasil cetak foto yang baik telah menjadikan Gembira *Photo* sebagai tempat tujuan pelanggan yang ingin mendapatkan hasil cetak foto yang memuaskan. Gembira *Photo* juga berperan sebagai tempat rujukan bagi studio-studio lain dalam skala yang lebih kecil dari Gembira *Photo*. Letaknya yang strategis di jalan Inggit Garnasih No 21 Bandung menjadikan Gembira *Photo* mudah didatangi para pelanggan.

Gembira *Photo* didirikan oleh Jaya Koswara yang sekaligus jadi pemilik, sampai sekarang Gembira *Photo* sudah berusia 32 tahun. Umur yang sangat matang, serta dengan berbagai pengalaman telah menjadikan Gembira *Photo* tetap eksis di kota Bandung menghadapi persaingan antara studio foto yang semakin ketat.

2.1.1 Sejarah Singkat

Studio Gembira *Photo* berdiri pada tanggal 18 Mei 1984 dan telah berbadan hukum pada tanggal 18 juni 1985. Pendiri Gembira *Photo* adalah Jaya Koswara dengan jumlah karyawan saat itu sebanyak 5 orang. Dunia usaha bidang fotografi bagi Jaya Koswara bukanlah dunia baru yang digelutinya. Kurang lebih 32 tahun lamanya beliau berkencipung dalam dunia pembuatan pas foto dan percetakan. Mengawali karir sebagai seorang manajer laboratorium pada tahun 1982 sampai dengan 1983. Kemudian tahun 1983 sampai 1984 menjadi kepala cabang laboratorium cuci cetak dan menangani pemasaran divisi cuci cetak di salah satu cetak di salah satu perusahaan percetakan foto terkemuka di kota Bandung.

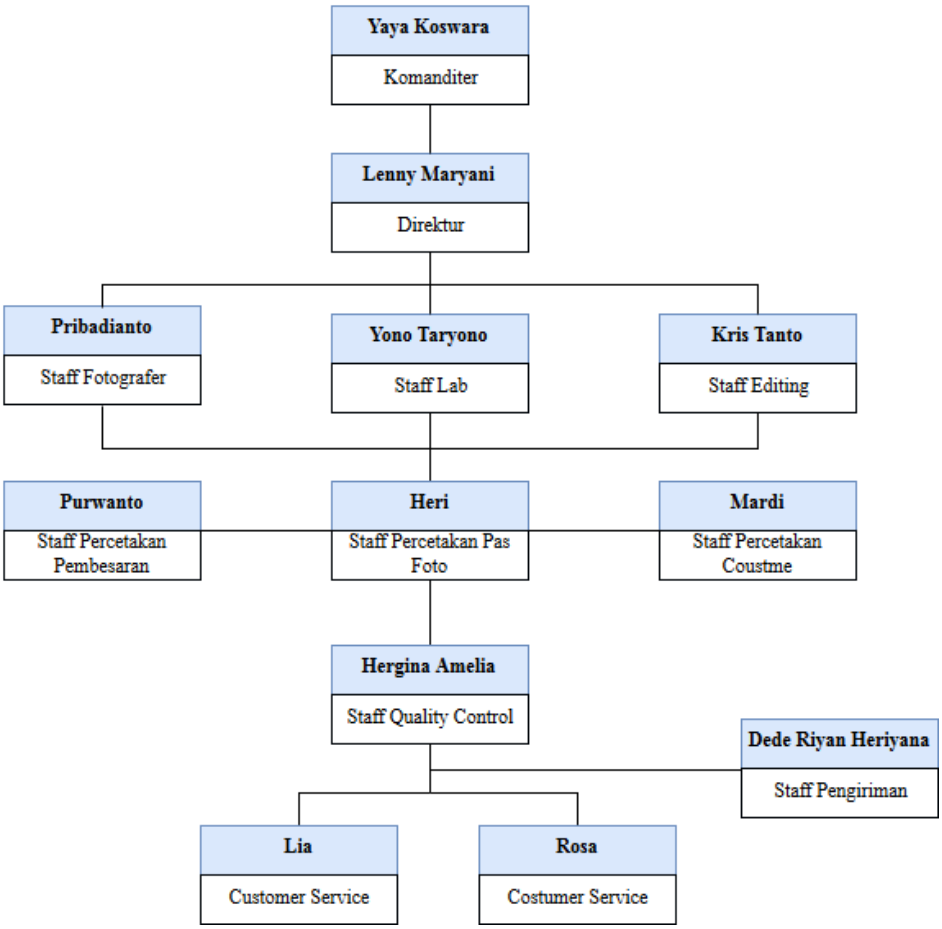
Berbekal pengalaman, keuletan, dan semangat kerja yang profesional, Bapak mendirikan sebuah usahan yang bergerak di dunia fotografi dan pembuatan pas foto, yang diberi nama Stuido *Gembira Photo*. Perkembangan fotografi di *Gembira Photo* dari tahun ke tahun makin berkembang dengan tetap

mempertahankan ciri khasnya sebagai pelopor spesialisasi baru dalam bidang fotografi di Bandung khususnya sampai saat ini.

2.1.2 Struktur Organisasi

Struktur organisasi adalah suatu susunan dan hubungan antara tiap bagian serta posisi yang ada pada suatu organisasi atau perusahaan dalan menjalankan kegiatan operasional untuk mencapai tujuan. Struktur organisasi menggambarkan dengan jelas pemisahan kegiatan pekerjaan antara yang satu dengan yang lain dan bagaimana hubungan aktivitas dan fungsi dibatasi [4].

Adapun Struktur Organisasi dari Studio Gembira *Photo* adalah sebagai berikut :



Gambar 2.1 Struktur Organisasi Studio Gembira *Photo*

2.1.3 Deskripsi Tugas

Deskripsi tugas digunakan untuk mengetahui tugas, wewenang tanggung jawab dari masing-masing bagian. Deskripsi tugas yang ada di Studio Gembira *Photo* di jelaskan pada Tabel 2.1.

Tabel 2.1 Deskripsi Tugas

No	Jabatan	Deskripsi Tugas
1	Komanditer	Bertugas sebagai pengawas dari berjalanya suatu Studio Gembira *Photo*
2	Direktur	Bertugas sebagai perwalian perusahaan dalam hubunganya dengan dunia luar perusahaan
3	Staff Fotografer	Bertugas sebagai pengambilan foto dalam perusahaan
4	Staff Lab	Bertugas untuk mengatur lab yang akan dipakai dalam perusahaan
5	Staff Editing	Bertugas untuk mengedit bagian foto yang mau di cetak
6	Staff Percetakan	Bertugas untuk mencetak semua foto yang masuk dari editing
7	Staff Quality Control	Bertugas untuk mengcek semua foto yang di cetak
8	Staff Pengeriman	Bertugas untuk mengirim foto ke pelanggan
9	Coustomer Service	Bertugas untuk melayanin semua pelanggan

2.1.4 Logo Gembira Photo

Logo Gembira *Photo* dapat dilihat pada Gambar 2.2.



Gambar 2.2 Logo Gembira *Photo*

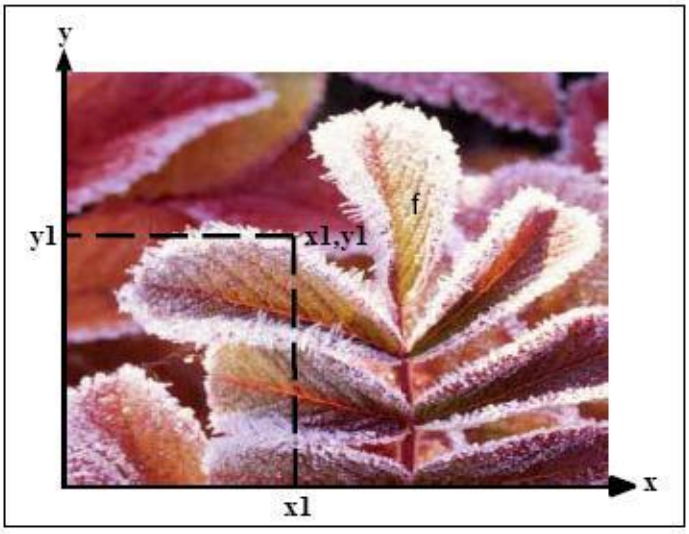
2.2 Citra

Jika ditinjau dari sudut pandang matematis, citra merupakan fungsi mnerus (*continue*) dari intentitas cahaya pada bidang dwimatra. Sumber cahaya menerangi objek, objek memantulkan kembali sebgai dari berkas cahaya tersebut. Pantulan cahaya ini ditangkap oleh alat-alat optik, misalnya mata pada manusia, kamera, pemindai (*scanner*), dan sebagainya, sehingga bayangan objek yang disebut citra tersebut direkam [5].

Citra adalah representasi dari suatu obyek nyata baik dalam bentuk dua dimensi ataupun tiga dimensi menjadi bentuk gambar digital yang dimengerti oleh komputer. Definisi citra lainya adalah sebuah fungsi intensitas warna dua dimensi f(x,y) dimana x dan y mewakili kordinat lokasi suatu titik dan nilai dari fungsi yang merupakan tingkat intensitas warna atau tingkat keabu-abuan dari titik tersebut.

2.2.1 Citra Digital

Citra digital dapat didefinisikan sebagai fungsi dua variabel, f(x,y), dimana x dan y adalah kordinat spasial dan nilai f(x,y) yang merupakan intensitas citra pada kordinat tersebut. Teknologi dasar untuk menciptakan dan menampilkan warna pada citra digital berdasarkan pada sebuah warna merupak kombinasi dari tiga warna dasar, yaitu merah, hijau, dan biru RGB (*Red, Green, Blue*) [6].



Gambar 2.3 Contoh Citra Digital [5]

2.2.2 Citra Foto

Citra foto adalah gambar yang dihasil kan dengan menggunakan sensor kamera. Contoh sitra foto (dalam hal ini foto udara) dapat dilihat pada Gambar 2.4 [7].



Gambar 2.4 Contoh Citra Foto [7]

Untuk mendapatkan citra foto atau gambar sesuai dengan yang diinginkan tidak mungkin di lakukan pada gambar atau foto tercetak. Namun dengan citra digital dimungkinkan melakukan pengolahan atau editing citra. Beberapa program dekstop publishing selain mempunyai fasilitas pengolahan citra secara itomatis, tersedia pula fasilitas pengolahan citra customize, khususnya untuk operasi filter-risasi. Operasi ini dapat dilakukan dengan teknik sederhana melalui algoritma dan teknik- teknik trasnformasi citra sederhanan. Algoritma dalam transformasi citra khususnya pada filterisasi digital memodifikasi nilai digital pixel tampilan warna citra berdasarkan pertimbangan nilai digital warna dari pixel yang bersangkutan dan tetangga (*neighbourhood*).

2.2.3 Pengolahan Citra

Pengolahan citra (image processing) adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjad citra yang kualitasnya menjadi lebih baik [8]. Pengolahan citra ini sangat diperlukan walaupun citra sangat kaya dengan informasi, namun seringkali citra mengalami penurunan mutu (degradasi), misalnya mengandung cacat atau derau (noise), warnanya terlalu kontars, kurang tajam, kabur (blurring) dan sebagainya. Tentu saja citra semacam ini menjadi sulit

diinterpretasi karena informasi yang disampaikan oleh citra tersebut menjadi berkurang. Operasi-operasi pada pengolahan citra diterapkan pada citra bila :

1. Perbaikan atau modifikasi citra dilakukan untuk meningkatkan kualitas penampakan citra menonjolkan beberapa aspek informasi yang terkandung dalam citra. Contoh : perbaikan kontras gelap atau terang, perbaikan tepian objek, penajaman, pemberian warna semu dan lain sebagainya.
2. Adanya cacat pada citra sehingga perlu dihilangkan atau diminimumkan (image restoration). Contoh : penghilangan kesamaran (deblurring) citra tampak kabur karena pengaturan fokus lensa tidak tepat, penghilangan noise dan lan sebagainya.
3. Elemen dalam citra perlu dikelompokan, dicocokan atau diukur (image segmentation). Operasi ini berkaitan erar dengan pengenalan pola.
4. Diperlukan ekstraksi ciri-ciri tertentu yang dimiliki citra untuk membantu dalam pengidentifikasian objek (image analysis). Proses segmentasi kadangkala diperlukan untuk melokalisasi objek yang diinginkan dari sekelilingnya.
5. Sebagian citra perlu digabung dengan bagian citra yang lain (image recontruction). Contoh : beberapa foto rontgen digunakan untuk membentuk ulang gambar organ tubuh.
6. Citra perlu dimampatkan (image compression) contoh : suatu file citra berukuran 256 KB dimampatkan menjadi 49 KB.
7. Menyembunyikan data rahasia (berupa citra atau teks) pada citra sehingga keberadaan data rahasia tersebut tidak diketahui orang (steganografi dan watermaking).

Beberapa jenis operasi pengolahan citra sebagai berikut :

1. Modifikasi kecemerlangan (Brightness Modification)

   Mengubah nilai keabuan atau warna dari gelap menuju terang atau sebaliknya mengubah citra yang terlalu cemerlang menjadi gelap.
2. Peningkatan kontras (Contrast Enchanment)

Dengan peningkatan kontras mak titik yang cendrung gelap menjadi lebih cemerlang.

3. Negasi

   Negasi yaitu operasi untuk mendapatkan citra negatif (negative image).

4. Pencerminan (Flipping)

   Pencerminan merupakan proses menggambar citra ke bentuk kebalikannya seperti ketika bercermin.

5. Rotasi (Rotation)

   Rotasi yaitu proses memutar koordinat citra sesuai derajat yang telah ditemtukan.

6. Pemotongan (Cropping)

   Pemotongan yaitu proses mengambil bagian dari suatu citra sesuai dengan kebutuhan.

7. Penskalaan (Scalling)

   Scalling yaitu proses pengubahan ukuran citra menjadi lebih besar atau menjadi lebih kecil.

8. Model Warna Pada Citra

   Warna adalah persepsi yang dirasakan oleh sistem visual manusia terhadap panjang gelombang cahaya yang dipantulkan oleh objek. Setiap warna mempunyai panjang gelombang yang berbeda [8]. Warna merah mempunyai panjang gelombang paling tinggi, sedangkan wrna ungu mempunyai panjang gelombang paling rendah. Warna-warna yang diterima oleh mata merupakan hasil kombinasi cahaya dengan panjang gelombang berbeda. Penelitian memperlihatkan bahwa kombinasi warna yang memberikan rentang warna yang paling lebar adalah red (R), green (G), blue (B).

2.3 Tahapan Proses Pengambilan Gambar Pas Foto

Pengambilan gambar pas foto pada pembahasan ini merupakan tahap-tahap proses pengambilan pas foto menggunakan metode yang dipakai. Tahap proses pengambilan foto adalah sebagi berikut :

a. Proses ambil foto
b. Proses *background subtraction*
c. Proses ganti warna *background*

2.3.1 Proses Ambil Foto

Proses ambil foto merupakan pengambil foto yang akan di proses dengan *background subtraction*. Dalam proses pengambilan foto ini dilakukan dua tahap pengambilan yaitu pengambilan foto background dan pengambilan foto background beserta objek dalam pengambilan sendiri di tempat yang sama.

Contoh ambil foto background yang akan dideteksi sebagai berikut :



Gambar 2.5 Contoh Foto *Background*

Contoh ambil foto background dan objek yang akan dideteksi sebagai berikut :



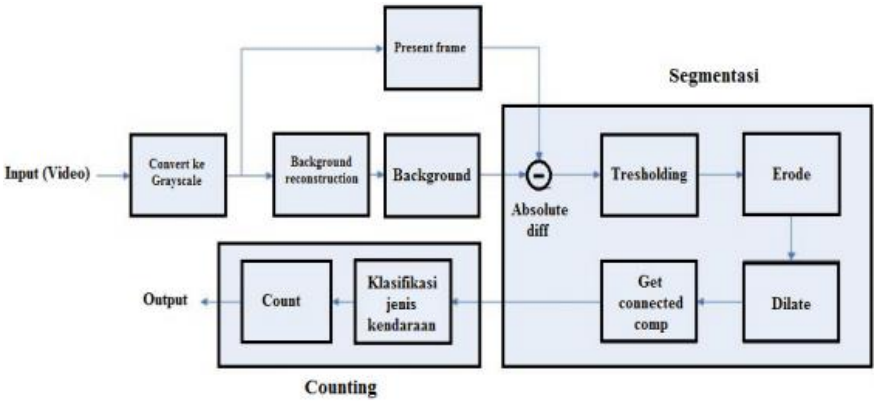
Gambar 2.6 Contoh Foto Background dan Objek

2.3.2 Proses *Background Subtraction*

Background pada pembahasan ini merupakan sejumlah piksel-piksel gambar yang diam dan tidak bergerak didepan kamera. Model *background* yang paling sederhana mengasumsikan bahwa seluruh kecerahan piksel *background* berubah-ubah secara bebas, tergantung pada distribusi normalnya. Karakteristik

background dapat dihitung dengan mengakumulasi beberapa jumlah foto sehingga akan menemukan jumlah nilai-nilai piksel [9].

Background subtraction memiliki beberapa tahap sebelum dapat mendeteksi objek, tahap-tahap dari metode ini adalah sebagai berikut :



Gambar 2.7 Contoh Proses *Background Subtraction* [9]

1. *Convert Grayscale*

Citra *grayscale* adalah citra yang hanya memiliki satu nilai kanal pada setiap pixelnya, dengan kata lain nilai bagian *Red* = *Green* = *Blue*. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas. Warna yang dimiliki adalah warna dari hitam, keabuan dan putih. Tingkat keabuan di sini merupakan warna abu dengan berbagai tingkatan dari hitam hingga mendekati putih. [10]

Pengubahan dari citra berwarna ke bentuk *grayscale* mengikuti aturan sebagai berikut :

$$Y = (0{,}229 * R) + (0{,}587 * G) + (0{,}114 * B) \ldots\ldots (1)$$

dimana :

Y = derajat keabuan

R = nilai pixel channel *Red*

G = nilai pixel channel *Green*

B = nilai pixel channel *Blue*

Contoh gambar hasil *grayscale* adalah sebagai berikut :



Gambar 2.8 Contoh Gambar Hasil Konversi Grayscale [10]

2. *Absolute*

Absolute adalah tahap pengurangan setiap nilai piksel citra current frame dengan setiap nilai piksel citra *background*, untuk mendapatkan *foreground* sebagai objek yang dideteksi. Proses pengurangan citra dilakukansecara absolut dimana perbedaan setiap *pixel* dari kedua citra akan diperolehdan selalu bernilai positif. Karena teknik ini akan melihat perbedaan untuksetiap pixel di dalam citra, sehingga kedua citra harus memiliki tipe data danukuran yang sama [10]. *Absolute* dapa dilakukan dengan menggunakan persamaan berikut:

$$Z_{(x,y)} = \left|B_{(x,y)} - C_{(x,y)}\right| \dots\dots\dots (2)$$

dimana :

Z = nilai piksel *absolute*

B = nilai piksel foto background dan objek

C = nilai piksel foto background

(x,y) = koordinat nilai piksel pada matriks

3. *Threshold*

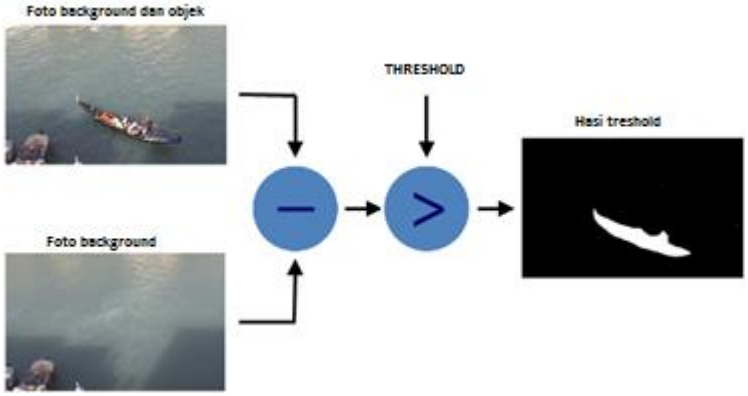
Threshold merupakan teknik yang sederhana dan efektif untuk segmentasi citra. Proses *thresholding* sering disebut dengan proses binerisasi. Pada beberapa aplikasi pengolahan citra, terlebih dahulu dilakukan threshold terhadap citra gray level untuk dapat menjadi citra biner (citra yang memiliki nilai level keabuan 0 atau 255) [10]. Untuk menentukan nilai *threshold* dari tiap nilai matriks citra dapat menggunakan persamaan berikut :

$$g(x,y) = \{bernilai\ 1\ jika\ f(x,y) \geq T\}$$
$$g(x,y) = \{bernilai\ 0\ jika\ f(x,y) < T\} \dots\dots\dots\dots\dots\dots (3)$$

dimana :

g(x,y) = merupakan nilai matrix citra yang akan di-threshold.

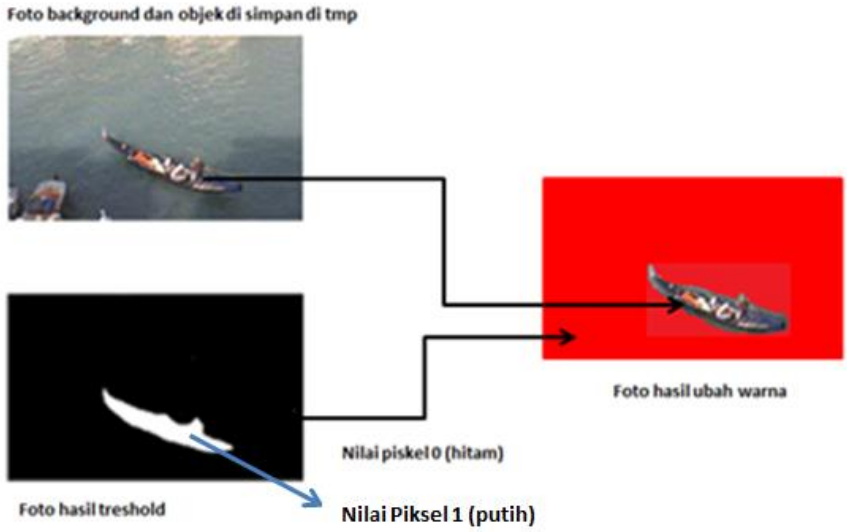
T = merupakan nilai threshold (0 – 255)



Gambar 2.9 Ilustrasi *Background Subtraction* [10]

2.3.2 Proses Rubah Warna Background

Proses rubah warna background dilakukan setelah melakukan proses thershold. Dalam proses perubahan warna dilakukan pergantian warna sesuai dengan warna yang diinginkan. Adapun cara pergantian warna dengan melakukan proses travesal(bergantian satu persatu) pada setiap piksel pada gambar hasil threshold. Untuk setiap piksel, jika nilai piksel tersebut adalah 0 (hitam atau #000000) maka ganti piksel tersebut dengan warna penganti (merah atau biru), jika nilai piksel tersebut adalah 1 (putih atau #FFFFFF) maka ganti piksel tersebut dengan piksel pada koordinat yang bersesuaiaan dengan nilai piksel pada tmp.



Gambar 2.10 Contoh Proses Rubah Warna Background

2.4 Foto

Foto, pada dasarnya merupakan sarana berkomunikasi bagi seorang fotografer kepada orang lain tentang suatu hal. Foto dapat diartikan sebagai ungkapan melalui bahasa gambar atau visual. Lewat selembar foto, seseorang dapat bercerita mengenai auatu peristiwa, kegiatan, epspresi, kenangan atau bahkan berbagi ide/gagasan kepada orang lain. pada perkembangan, foto juga berperan sebagai penyampai berita. Foto menjadi fakta atas sebuah peristiwa [11].

Selain definisi diatas dan makna kiasan yang dapat dikiaskan dengan kata foto. Secara kategorisasi foto juga harus dibedakan menjadi beraga. Kategorisasi ini bertujuan untuk memudahkan pembuatan dan pemanfaatannya sesuai dengan standar kualitas bagi masing-masing keperluan. Ada banyak sekali kategori foto, antara lain: foto keluarga, foto dokumentasi, foto resmi, foto salon, foto kriminal, foto kedokteran, foto infa merah, foto bawah laut, foto satelit, foto udara,foto mikro, foto jurnalistik, dan lain-lain.

Selain itu, ada pula kategori foto berdasarkan ukuran. Misalnya pas foto, foto seluruh badan, foto KTP, paspor, foto postcard, dan lain-lain. Ada pula pembedaan sesuai dengan jenis kameranya. Misalnya foto analog (dengan film) dan foto digital.

2.5 Pas Foto

Pas foto merupakan foto kecil yang diambil dari kepala sampai dada dengan keadaan tubuh tegak yang sering diperlukan sebagai tanda pengenal atau informasi identitas seseorang. Pas foto digunakan untuk aktivitas dan dokumentasi formal untuk pembuatan pasport, surat izin mengemudi, kartu tanda penduduk, pendaftaran smbtn, ijazah sekolah, foto wisuda dan untuk melamar pekerjaan [12]. Proses pembuatan pas foto tidak lepas dari apa saja yang perlu diperhatikan dalam membuat pas foto. Dalam hal ini syarat untuk membuat pas foto untuk kualitas gambar harus cukup tajam dan fokus, posisi badan dan kepala tegak sejajar menghadap kamera, untuk wajah antara 25% - 50% dari foto dan kepala terletak di tengah secara horizontal (jarak kepala kebats kiri kurang lebih sama dengan jarak kepala ke batas kanan). Untuk standar warna *background* dalam membuat pas foto berwaran merah atau biru, dalam ukuran pas foto terdapat banyak ukuran yaitu 2x3cm, 3x4cm, 4x6cm. Untuk jenis kertas yang digunakan kertas *glossy photo paper*.



Gambar 2.11 Contoh Pas Foto

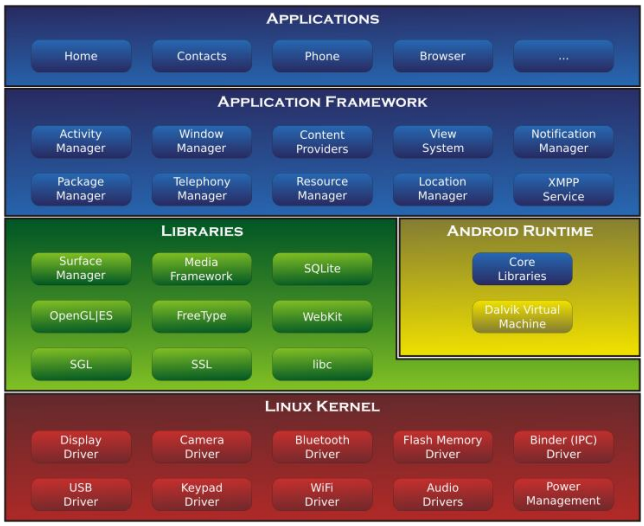
2.6 Android

Android adalah sistem operasi untuk telpon seluler yang berbasis linux, pada penelitian ini, *Android* digunakan sebagai media pembangunan suatu apliaksi. *Android* menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak.

Awalnya, Google Inc. Membeli Android Inc., pendatang baru yang membuat perangkat lunak untuk ponsel. Kemudian untuk mengembangkan *Android*, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan perangkat keras, perangkat lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia [13].

2.6.1 Arsitektur Android

Google menggembangkan *Android* seperti sebuah software. Setiap lapisan dari tumpukan ini terdiri dari beberapa program yang mendukung fungsi-fungsi spesifikasi dari sistem operasi [14]. Dalam paket system operasi *android* terdiri dari beberapa unsure seperti tampak pada Gambar 2.12. Secara sederhana arsitektur *android* merupakan sebuah kernel Linux dan sekumpulan pustaka C/C++ dalam suatu Framework yang menyediakan dan mengatur alur proses aplikasi.



Gambar 2.12 Arsitektur *Android* [14]

Berikut adalah penjelasan mengenai masing-masing lapisan pada arsitektur *android*:

1. Linux Kernel

Android dibangun diatas kernel Linux. Namun secara *android* bukanlah linux, karena dalam *android* tidak terdapat paket standar yang dimiliki oleh linux lainya. Linux merupakan sistem operasi terbuka yang handal dalam manajemen memori dan proses. Oleh karenanya pada *android* hanya terdapat

beberapa servis yang diperlukan seperti keamanan, manajemen memori, manajemen proses, jaringan dan *driver*. Kernel linux menyediakan *driver* layar, kamera, *keypad*, *Wifi*, *Flash Memory*, Audio, dan IPC (*Interprocess Communication*) untuk mengatur aplikasi dan lubang keamanan [14].

2. Liberies

Android menggunakan bebrapa paket pustaka yang terdapat pada C/C++ dengan standar *Berkeley Software Distribution* (BSD) hanya setengah dari yang aslinya untuk tertanam pada kernel linux. Bebreapa pustaka diantaranya:

1. Media *Library* untuk memutar dan merekam berbagai macam format audio dan video.
2. *Surface Manager* untuk mengatur hak akses layer dari apliaksi.
3. *Graphic Library* termasuk didalamnya SGL dan OpenGL, untuk tampilan 2D dan 3D.
4. *SQLite* untuk mengatur relasi *database* yang digunkan pada aplikasi.
5. *SSI* dan *WebKit* untuk *browser* dan keamanan internet.

Pusaka-pustaka tersebut bukanlah aplikasi yang berjalan sendiri, namun hanya dapat digunakan oleh program yang berada di level atasnya. Sejak versi *Android* 1.5, pengembang dapat membuat dan mengunkan pustaka sendiri menggunkan *Native Devlopment Toolkit* (NDK).

3. Android Runtime

Pada *Android* tertanam paket pustaka inti yang menyediakan sebagian besar fungsi *android*. Inilah yang membedakan *android* dibandingkan dengan sistem operasi lain yang juga mengimplentasikan linux [14]. *Android Runtime* merupakan mesin virtual yang membuat apliaksi android menjadi lebih tangguh dengan paket pustaka yang telah ada. Dalam *Android Runtime* terdapat 2 bagian utama, diataranya:

1. Pustaka Inti, *android* dikembangkan melalui bahasa pemograman java, tapi *Android Runtime* bukanlah mesin virtual java. Pada inti *android* menyediakan hampir semua fungsi semua fungsi yang terdapat pada pustaka java serta bebrapa pustaka khusu *android*.

2. Mesin Virtual Dalvik, Dalvik merupakan sebuah mesin virtual yang dikembangkan oleh Dan Bornstein yang terinspirasi dari nama sebuah perkampungan yang berada di Iclend. Dalvik hanyalah interpreter mesin virtual yang mengksekusi file dalam format *Dalvik Exutable* (*.dex). Dengan format ini Dalvik akan mengoptimalkan efisiensi penyimpanan dan pengamatan memori pada file yang dieksekusi. Dalvik berjalan di atas kernel linux 2.6, dengan fungsi dasar seperti *threading* dan manajemen memori yang terbatas.

4. Aplication Framework

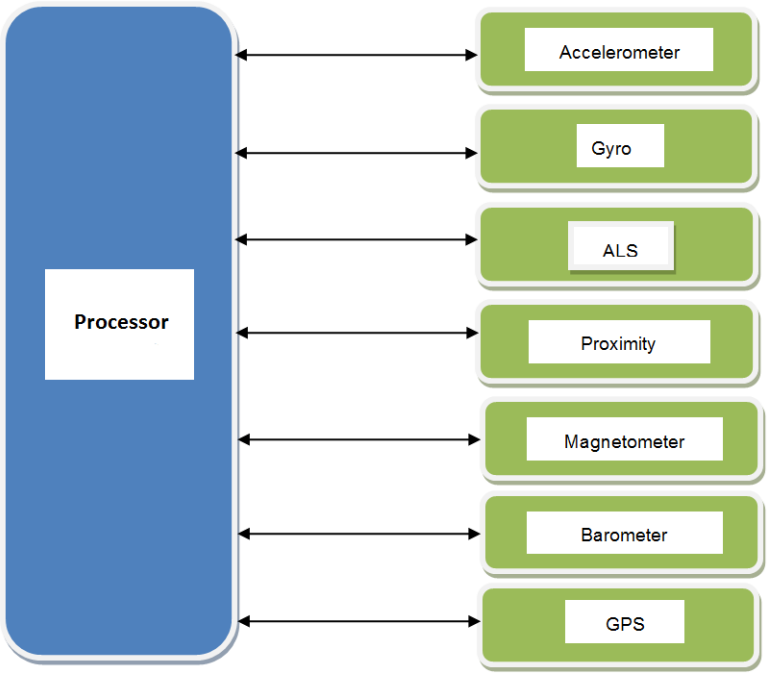
Kerangka aplikasi menyediakan kelas-kelas yang dapat digunakan untuk mengembangkan aplikasi *android*. Selain itu, juga menyediakan abstaksi untuk mengakses perangkat, serta mengatur tampilan *User Interface* dan sumber daya aplikasi [14].

5. Aplication Layer

Puncak dari diagram arsitektur *android* adalah lapisan apliaksi dan *widget*. Lapisan aplikasi merupakan lapisan yang paling tampak pada pengguna ketika menjalankan program [14]. Pengguna hanya akan melihat program ketika digunkan tanpa mengetahui proses yang terjadi dibalik layar aplikasi. Lapisan ini berjalan dalam *Android runtime* dengan menggunkan kelas dan *service* yang tersedia pada Framework aplikasi. Lapiasan aplaikasi android sangat berbeda dibandingkan dengan sitem operasi lainya. Pada android semua aplikasi, baik aplkasi inti (*native*) maupun aplikasi pihak ketiga berjalan diatas lapisan aplikasi dengan menggunkan pustka API (*Application Programming Interface*) yang sama.

2.6.2 Sensor *Android*

Ponsel dan tablet *android* berbasis processor mendukung berbagai sensor. Sensor-sensor tersebut digunakan untuk mendeteksi gerakan dan perubahan posisi, dan melaporkan parameter lingkungan sekitar. Blok diagram di Gambar 2.13 menampilkan konfiguransi sensor yang memungkinkan di perangkat *andorid* berbasis processor pada umumnya [15].



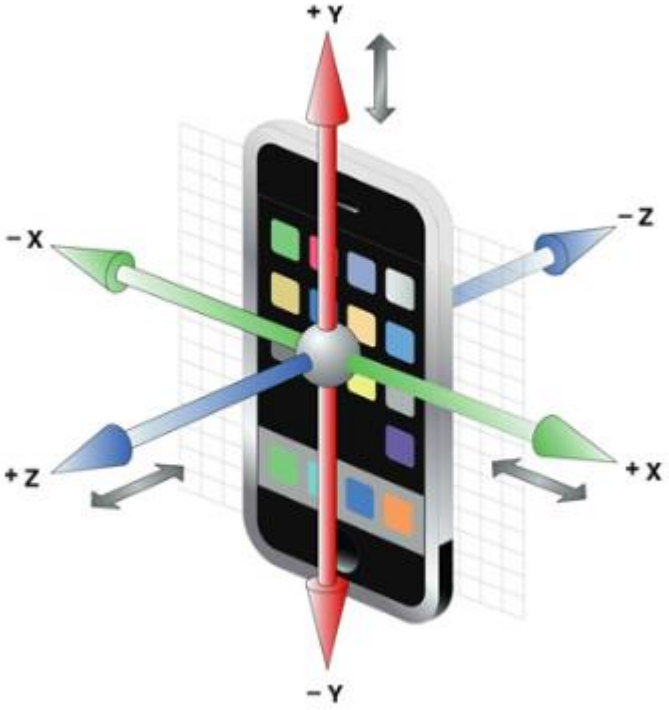
Gambar 2.13 Sensor *Android* [15]

2.7 *Accelerometer*

Accelerometer adalah sebuah sensor atau tranduser yang berfungsi untuk mengukur percepatan, mendeteksi dan mengukur getaran, ataupun untuk mengukur percepatan akibat gravitasi bumi. Pada penelitian ini, *accelerometer* digunakan untuk mengambil nilai sudut kemiringan dari suatu objek [16].

Pada *smartphone*, *Accelorometer* merupakan sensor yang membaca perangkat sehingga dapat mengubah tampilan layar dari posisi landscape ke potrait atau sebaliknya dengan cukup memiringkan badan ponsel secara otomatis.

Pada *smartphone* berbasis *android* terdapat sensor manager yang berfungsi untuk mengaktifkan sensor *accelorometer* dalam mencari nilai kordinat x,y,z dengan kemiringan handphone.



Gambar 2.14 Sumbu *Accelerometer* [16]

2.8 *Tools* Pendukung

Tools pendukung disini sangat penting dalam pembangunan sistem program terdapat beberapa *tools* perangakat pendukung untuk membuat aplikasi pada *smartphone* berbasis *mobile android*, bebrapa *tools* perangakat lunak pendukungnya sebagai berikut :

2.8.1 *Android* Studio

Android studio adalah lingkungan pengembangan terpadu (IDE) remi untuk pengembangan *platfrom android*, hal itu disampaikan pada tanggal 16 mei 2013 Google I/O, *android* studio sudah tersedia secara bebas dibawah lisensi Apache 2.0.

Android studio pada awalnya tahap preview versi 0.1 yang dipakai pada tanggal 1 mei 2013 dan memasuki tahap beta pada bulan 2014 dan memulai stabil diliris Desember 2014 dengan versi 1.0, berdasarkan jet Brains IDEA *Softweare*, android studio dirancang khusus untuk pengembangan android yang tersedia untuk Windows, Mac OS X dan Linux Sebagai pengganti Eclipse [17].

2.8.2 Microsoft Visio

Microsoft Viso adalah program aplikasi komputer yang sering digunakan untuk membuat diagram, diagram alir (*flowchart*), *brainstorm*, dan skema jaringan yang diliris oleh Microsoft Comporation. Aplikasi ini menggunkan grafik vektor untuk membuat diagram-diagramnya.

Viso aslinya bukanlah buatan Microsoft Corporation, melainkan buatan Visio Corporation, yang diakusisisi oleh Microsoft pada tahun 2000. Versi yang telah menggukan nama Microsoft Visio adalah Visio 2002, Visio 2003, Visio 2007, Visio 2013, dan Visio 2016 yang merupakan versi terbaru. Visio 2007 atau yang lebih baru Standar dan professional menawarkan lebih banyak pilihan template untuk pembuatan diagram yang lebih lanjut dan juga penataan letak (layout). Selain itu edisi Professional juga memudahkan pengguna untuk mengoneksikan diagram-diagram buatan mereka terhadap beberapa sumber data dan juga menampilkan informasi secra visual dengan menggunakan grafik [18].

2.8.3 XAMMP

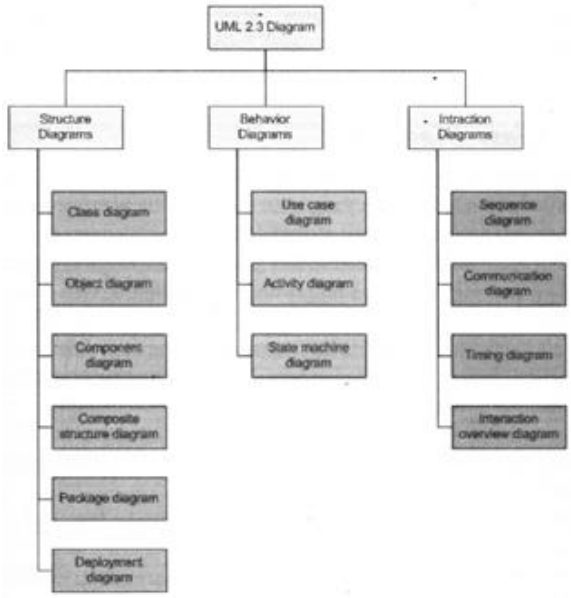
XAMPP merupakan paket php berbasis open source yang dikembangkan oleh sebuah komunitas Open Scource. Dengan menggunkan XAMPP kita tidak perlu lagi melakukan penginstalan program yang lain karena semua kebutuhan telah disediakan oleh XAMPP [19].

Beberapa paket yang telah disediakan adalah Apache, MySql, Php, Filezia, dan Phpmayadmin digunakan untuk membangun sebuah lokal untuk membuat simulasi sebuah website yang memiliki server side script sebelum dipublikasikan pada server yang sebenarnya.

2.9 UML (*Unified Modeling Language*)

Unified Modeling Language (UML) adalah termasuk kedalam rumpun jenis permodelan notasi grafis yang didukung oleh model-model tunggal. Permodelan ini berguna untuk membantu dalam menjelaskan dan merancang perangkat lunak yang dibangun dengan *object-oriented* (OO). UML merupakan standar terbuka yang dikelola oleh *Open Management Group* (OMG) yang berada dibawah

naungan perusahaan-perusahaan konsorsium terbuka. Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada Gambar 2.15 [19].



Gambar 2.15 Diagram UML [19]

2.9.1 OOAD

Konsep OOAD (*Object Oriented Analysis Design*) mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaitu :

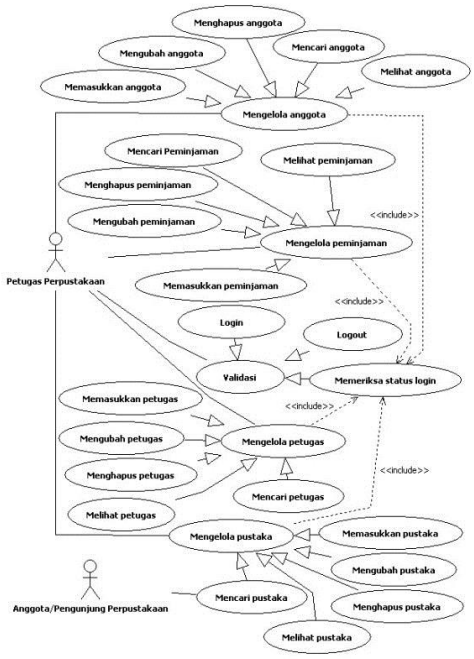
1. Analisis berorientasi objek (OOA)

Metode analisis yang memeriksa requirement (syarat/keperluan) yang harus dipenuhi sebuah sistem dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup perusahaan. *Unified Modeling Language* (UML) adalah bahasa untuk mengspesifikasi, memvisualisasi, membangun dan mendokumentasi artefacts (bagian dari informasi yang digunakan atau dihasilkan oleh proses pembuatan perangkat lunak), seperti pada pemodelan bisnis dan sistem non perangkat lunak lainya. UML merupakan bahasa standar untuk penulisan blueprint software yang digunakan untuk visualisasi, spesifikasi, pembentukan dan pendokumentasian alat-alat dari sistem perangkat lunak.

Berikut ini merupakan beberapa bagian dari UML adalah sebagi berikut :

a. *Use Case Diagram*

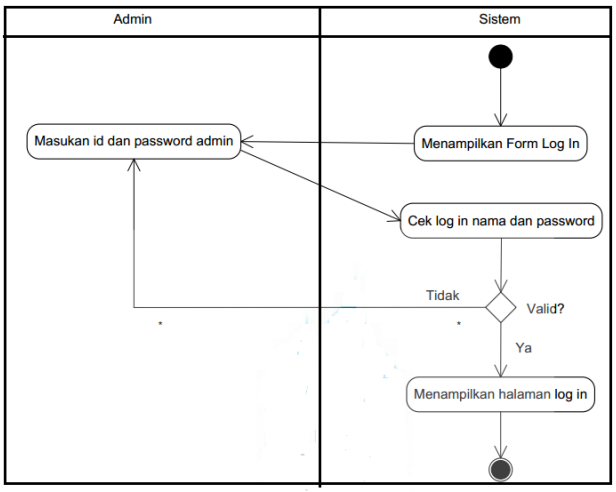
Use case diagram merupakan pemodelan untuk kekukan (behavior) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antra satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Contoh *Use Case Diagram* dapat dilihat pada Gambar 2.16.



Gambar 2.16 Contoh *Use Case Diagram*

b. *Activty Diagram*

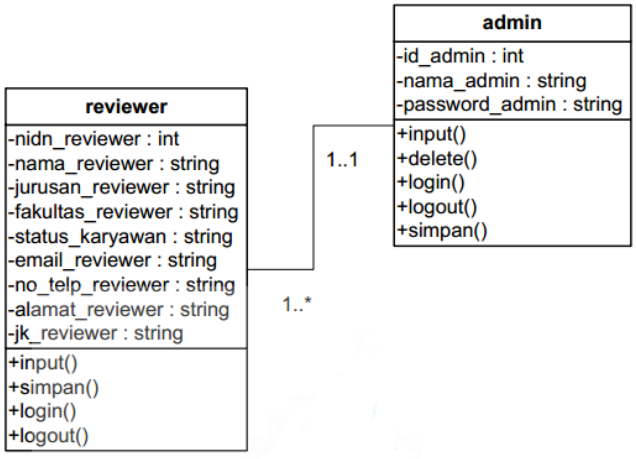
Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Contoh *Activity Diagram* dapat dilihat pada Gambar 2.17.



Gambar 2.17 Contoh *Activty Diagram*

c. *Class Diagram*

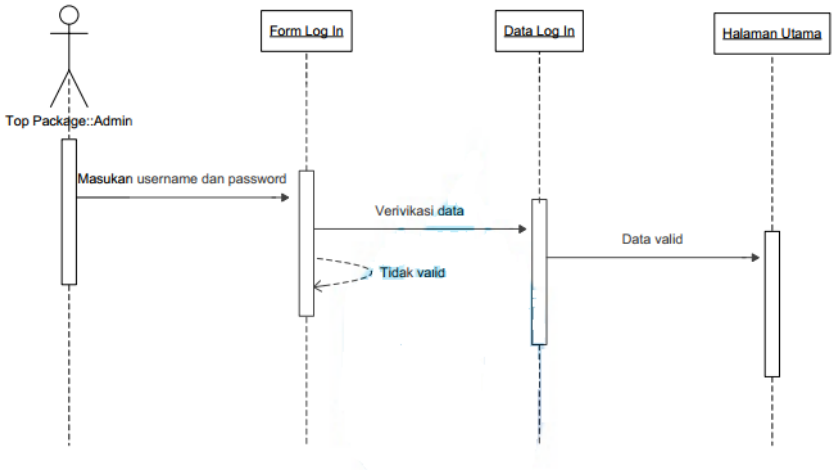
Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Contoh *Class Diagram* dapat dilihat pada Gambar 2.18.



Gambar 2.18 Contoh *Class Diagram*

d. *Sequence Diagram*

Sequence diagram menggambarkan kolaborasi dinamis antara sejumlah objek dan untuk menunjukan rangkaian pesan yang dikirim antara objek juga interaksi yang terjadi pada titik tertentu dalam eksekusi sistem. Contoh Sequence Diagram dapat dilihat pada Gambar 2.19.



Gambar 2.19 Contoh *Sequence Diagram*

2.10 *Database*

Database adalah koleksi data-data yang terkait secara logis dan deskipsi dari data-data tersebut, yang dirancang untuk memenuhi kebutuhan informasi dari suatu organisasi. "*Database is a self-describing collection intergrated tables*", yang berarti *database* adalah sebuah koleksi data yang menggambarkan intergrasi antara table yang satu dengan table yang lainya. "*Database is a self-describing*", disini dijelaskan bahwa struktur data saling terintergasi dalam suatu tempat ang dikenal sebagi kamus data atau metadata [20].

Pada umumnya data dalam *database* bersifat *intergrated* dan *shared*, maskudnya dari *intergatade* adalah *database* merupakan penggabungan beberapa file data yang berbeda, dengan membatasi pengulangan baik keseluruhan file ataupun sebagaian. Pengertian *shared* adalah data individu dalam *database* dapat digunakan secra bersamaan antara bebrapa pengguna yang berbeda.

2.10.1 My SQL

MySql adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa inggris: database management system) atau DBMS yang multithead, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah ini lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial

untuk untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL [21].

Tidak seperti PHP atau Apache yang merupakan software yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu MySQL AB. MySQL AB memegang penuh hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Filandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius [22].

2.10.2 SQLite

SQLite merupakan sebah sistem manajemen basisdata relasional yang bersifat ACID-Cmpliant dan memiliki ukuran pustaka kode yang relatif kecil, ditulis dalam bahasa C. SQLite merupakan proyek yang bersifat public domain yang dikerjakan oleh D. Ricard Hipp.

Tidak seperti pada paradigma client-server umumnya, inti SQLite bukan lah sebuah sistem yang mandiri yang berkomunikasi dengan sebuah program, melainkan sebagai bagian intergal dari sebuah program secara keseluruhan. Sehingga protokol komunikasi utama yang digunakan adalah melalui pemanggilan API secara langsung melalui bahasa pemograman. Mekanisme seperti ini tentunya membawa keuntungan karena dapat mereduksi *overhead*, *latency times*, dan secara keseluruhan lebih sederhana. Seluruh elemen basisdata (definisi data, tabel, indeks, dan data) disimpan sebagai sebuah file. Kesederhanaan dari sisi disain bisa diraih dengan cara mengunci keseluruhan file basis data pada saat transaksi dimulai [23].

2.11 Metode Pengujian

Pada Siklus hidup pengembangan perangkat lunak (*System Development Life Cycle: SDLC*) salah satu proses yang harus dilakukan adalah proses pengujian (*testing*). Pengujian perangkat lunak adalah suatu teknik yang digunakan menguji apakah sebuah perangkat lunak yang dihasilkan telah memenuhi kebutuhan proses bisnis pengguna atau masih belum. Menurut

Pressman, *Testing* adalah proses eksekusi suatu program untuk menemukan keslahan sebelum digunakan oleh pengguna akhir (*end-user*) [24].

2.11.1 BlackBox

Metode blackbox memfokuskan pada keperluan fungsional dari sofwear. Karna itu ujicoba backbox memungkinkan pengembang sofware untuk membuat himpunan kondisi input yang akan melatih whitebox, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainya, selain menggunakan metode whitebox [25].

Ujicoba blackbox berusaha untuk menemukan kesalahan dalam beberpa kategori, diantranya :

1. Fungsi-fungsi yang salah atau hilang
2. Kesalahan interface
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan performa
5. Kesalahan insialisasi dalam terminasi

2.11.2 Kuesioner

Kuesioner digunakan untuk mendapatkan sampel yang *representative* (mewakili), yang dapat menggambarkan populasinya. Penentuan jumlah *sample* kuesioner pada penelitian ini menggunakan rumus Slovin [26].

$$n = \frac{N}{1+N.e2} \quad (4)$$

n = Ukuran *sample*

N = Ukuran populasi

e = Toleransi kesalahan pengambilan *sample*

2.11.3 Skala Likert

Skala Likert ditemukan dan dikebangkan oleh Rensis Likert pada tahun 1932 dimana skala Likert merupakan metode skala bioplar karena dapat digunakan mengukur sikap, pendapat dan persepsi seseorang atau sekelompok orang tentang fenomena sosial [27]. Pada skala Likert, variable yang akan diukur dijabarkan menjadi indikator *variable.* Setiap jawaban dihubungkan dengan

bentuk pernyataan atau dukungan sikap yang diungkapkan dengan kata-kata misalnya untuk pernyataan positif.