

BAB 2

LANDASAN TEORI

2.1 Penerapan

Implementasi adalah bermuara pada aktivitas, aksi, tindakan atau adanya mekanisme suatu sistem, implementasi bukan sekedar aktivitas, tapi suatu kegiatan yang terencana dan untuk mencapai tujuan kegiatan[15].

Penerapan adalah suatu perbuatan mempraktekkan suatu teori, metode, dan hal lain untuk mencapai tujuan tertentu dan untuk suatu kepentingan yang diinginkan oleh suatu kelompok atau golongan yang telah terencana dan tersusun sebelumnya[16].

2.2 Deteksi

Deteksi adalah suatu proses untuk memeriksa atau melakukan pemeriksaan terhadap sesuatu dengan menggunakan cara dan teknik tertentu. Deteksi digunakan untuk berbagai masalah, misalkan pendeteksi wajah, dimana sistem ini mengidentifikasi berbagai wajah.

Tujuan dari deteksi adalah memecahkan suatu masalah dengan berbagai cara tergantung metode yang diterapkan sehingga menghasilkan sebuah solusi.

2.3 Manipulasi

Manipulasi adalah suatu proses rekayasa dari kenyataan yang dibuat seolah nyata. Manipulasi dapat berwujud lingkungan fisik, tugas, dan induksi. Manipulasi yang diterapkan misalnya memanipulasi data, memanipulasi uang, dan lainnya.

Tujuan manipulasi adalah untuk mempelajari hubungan sebab-akibat atau hubungan kausal antara variabel antededen penyebab dan variabel konsekuensi[17].

2.4 Wajah

Wajah adalah bagian depan dari kepala, pada manusia meliputi wilayah dari dahi hingga dagu, termasuk rambut, dahi, alis, pelipis, mata, hidung, pipi, mulut, bibir, gigi, kulit, termasuk dagu. Wajah terutama digunakan untuk ekspresi wajah, penampilan, serta identitas. Identitas pada dasarnya adalah suatu yang unik yang melekat pada diri kita dan dapat dikenali oleh diri sendiri dan orang lain sebagai hal pembeda diantara manusia lainnya. Sementara itu, Harry Gardiner melihat identitas

sebagai pendefinisian diri seseorang sebagai individu yang berbeda dalam perilaku, keyakinan dan sikap [H. W. Gardener, *Lives Across Cultures: Cross-Cultural Human Development*, 6th ed. London: Pearson, 2017.].

Salah satu hal yang dapat dikaitkan dengan pengenalan identitas adalah wajah. Wajah atau muka adalah bagian depan dari kepala, pada manusia meliputi wilayah dari dahi hingga dagu, termasuk rambut, dahi, alis, mata, hidung, pipi, mulut, bibir, gigi, kulit, dan dagu. Tidak ada manusia yang mempunyai wajah identik 100%, bahkan pada manusia kembar identik sekalipun. Karena kembar identik pun pasti mempunyai perbedaan wajah walaupun tidak signifikan.

Citra sebagai keluaran suatu sistem perekaman data dapat bersifat optik berupa foto, bersifat analog berupa sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu media penyimpanan[18]. Wajah dapat di deteksi melalui kontur atau struktur wajah dan tepi citra. Tepi citra merupakan perubahan signifikan dari nilai keabuan pada suatu citra.



Gambar 2.1 Wajah Khabib Nurmagomedov

2.5 Video Digital

Video merupakan teknologi untuk menangkap, merekam, memproses, dan menghasilkan gambar yang bergerak. Sedangkan digital merupakan suatu sistem otomatis dengan komputerisasi atau format yang dapat dibaca oleh komputer. Maka

Video digital adalah gambar bergerak hasil produk dari industri komputer dan oleh sebab itu dijadikan standar data digital. Jaringan komunikasi yang dikembangkan untuk menangani data dapat dengan senang hati membawa video digital dan audio yang menyertainya dalam jarak tak terbatas tanpa kehilangan kualitas[1].

2.6 Cloud Computing

Cloud Computing atau komputasi awan merupakan tren baru di bidang komputasi terdistribusi dimana berbagai pihak dapat mengembangkan aplikasi dan layanan berbasis SOA (Service Oriented Architecture) di jaringan internet[19]. Cloud computing adalah seperangkat layanan jaringan yang memberikan skalabilitas, jaminan Quality of Service (QoS), dapat dipersonalisasi, platform komputasi murah yang dapat diakses secara luas[20].

Berdasarkan kategori layanannya *cloud computing* dikategorikan sebagai berikut :

1. *Public Cloud*

Public Cloud dimiliki dan dioperasikan oleh penyedia layanan cloud pihak ketiga, yang memberikan sumber daya komputasi mereka, seperti penyimpanan dan server dengan melalui internet. Dengan public cloud, semua perangkat keras, perangkat lunak, dan infrastruktur pendukung dimiliki dan dikelola oleh penyedia. Mengakses layanan ini dan mengelola akun menggunakan browser web.

2. *Private Cloud*

Private cloud mengacu pada sumber daya komputasi awan yang digunakan secara eksklusif oleh satu bisnis atau organisasi. Private cloud dapat secara fisik terletak di pusat data perusahaan di tempat. Beberapa perusahaan juga membayar penyedia layanan pihak ketiga untuk meng-host cloud pribadi mereka. Private cloud adalah *cloud* di mana layanan dan infrastruktur dipertahankan pada jaringan pribadi.

3. *Hybrid Cloud*

Hybrid Cloud menggabungkan cloud publik dan pribadi, terikat bersama oleh teknologi yang memungkinkan data dan aplikasi untuk dibagikan di antara mereka. Dengan memungkinkan data dan aplikasi untuk berpindah antara *cloud* privat dan publik, *cloud hybrid* memberi bisnis fleksibilitas yang lebih besar, lebih

banyak opsi penyebaran, dan membantu mengoptimalkan infrastruktur, keamanan, dan kepatuhan Anda saat ini.

Cloud berdasarkan layanan yang diberikan dapat dikategorikan sebagai berikut :

1. *Infrastructure as a service (IaaS)*

Kategori paling mendasar dari layanan cloud computing. Dengan IaaS, Anda menyewa infrastruktur TI seperti server dan mesin virtual (VM), penyimpanan, jaringan, sistem operasi dari penyedia cloud dengan dasar pay-as-you-go.

2. *Platform as a Service (PaaS)*

Platform sebagai layanan mengacu pada layanan komputasi awan yang memasok lingkungan berdasarkan permintaan untuk mengembangkan, menguji, memberikan, dan mengelola aplikasi perangkat lunak. Contohnya seperti Microsoft Azure.

3. *Serverless Computing*

Berkaitan dengan PaaS, komputasi tanpa server berfokus pada membangun fungsionalitas aplikasi tanpa menghabiskan waktu terus-menerus mengelola server dan infrastruktur yang diperlukan untuk melakukannya. Penyedia cloud menangani pengaturan, perencanaan kapasitas, dan manajemen server untuk Anda. Arsitektur tanpa server sangat skalabel dan dikendalikan oleh peristiwa, hanya menggunakan sumber daya saat fungsi atau pemicu tertentu terjadi.

4. *Software as a service (SaaS)*

SaaS merupakan model bisnis untuk menyampaikan aplikasi dalam bentuk layanan. Penggunaan SaaS akan menghasilkan berbagai manfaat seperti penghematan biaya, ketangkasan (*agility*) yang lebih baik, dan mempercepat penyampaian layanan kepada pengguna, serta peningkatan fleksibilitas dalam berbagai skala untuk mendukung lebih banyak pengguna yang diperlukan[21].

2.6.1 Microsoft Azure

Microsoft Azure adalah rangkaian layanan Cloud Computing yang terus berkembang untuk membantu organisasi menghadapi tantangan bisnis. Dalam Microsoft Azure kita di bebaskan untuk membangun, mengelola, dan menyebarkan

Perangkat Lunak pada jaringan global yang masif menggunakan perangkat favorit dan Frameworks. Microsoft Azure menyediakan software as a service (SaaS), Platform as a service (PaaS) dan Infrastructure as a Service (IaaS) dan mendukung banyak bahasa pemrograman, tools, dan framework yang berbeda, termasuk perangkat lunak dan perangkat lunak pihak ketiga khusus dan Microsoft. Salah satu layanan yang tersedia di dalam cloud computing Microsoft Azure adalah layanan kecerdasan buatan bernama *Cognitive Service*.

2.6.2 *Cognitive Service*

Microsoft Azure Cognitive Service adalah layanan penyedia API dan SDK untuk membantu pengembang membangun aplikasi yang cerdas tanpa harus memiliki sumberdaya terkait AI secara langsung. Cognitive Service menyediakan akses ke pembelajaran mesin dan algoritma yang bisa diakses dengan mudah hanya dengan memanggil layanan API [Microsoft, "The Developer's Guide to Azure," 2019.[Online].Available:

<https://go.microsoft.com/fwlink/?LinkId=862819&clcid=0x409>. [Accessed: 26-Aug-2019].] . Berikut layanan yang diberikan oleh Cognitive Service diantaranya adalah :

1. *Vision API*

Layanan yang tersedia pada kategori vision API diantaranya adalah:

a. *Computer Vision*

Layanan Computer Vision memberi akses ke algoritma canggih untuk memproses gambar dan mengembalikan informasi.

b. *Custom Vision Service*

Layanan Custom Vision memungkinkan pengembang untuk membangun pengklasifikasi gambar kustom.

c. *Face API*

Face API menyediakan akses ke algoritma wajah tingkat lanjut, memungkinkan deteksi dan pengenalan atribut wajah.

d. *Form Recognizer*

Form Recognizer mengidentifikasi dan mengekstraksi pasangan nilai kunci dan data tabel dari dokumen formulir; kemudian menampilkan data terstruktur termasuk hubungan dalam file asli.

e. *Ink Recognizer*

Ink Recognizer memungkinkan pengembang untuk mengenali dan menganalisis data goresan tinta digital, bentuk, dan konten tulisan tangan, dan menghasilkan struktur dokumen dengan semua entitas yang dikenal.

f. *Video Indexer*

Video Indexer memungkinkan pengembang untuk mengekstrak wawasan dari video Anda.

2. *Speech API*

Layanan yang tersedia pada kategori *Speech API* diantaranya adalah:

a. *Speech Services*

Speech Services menambahkan fitur-fitur berkemampuan bicara ke aplikasi.

b. *Bing Speech*

Bing Speech API memberi pengembang cara mudah untuk membuat fitur yang mendukung ucapan di aplikasi Anda.

c. *Translator Speech*

Translator Speech adalah sebuah layanan mesin translasi.

3. *Language API*

Layanan yang tersedia pada kategori *Language API* diantaranya adalah:

a. *Language Understanding service (LUIS)*

Language Understanding service (LUIS) memberikan kemampuan kepada aplikasi yang dibangun untuk mengerti apa yang orang inginkan dari kata-kata yang mereka ucapkan

b. *Qna Maker*

QnA Maker memungkinkan pengembang membangun layanan tanya jawab dari konten semi-terstruktur.

2.7 Application Programming Interface (API)

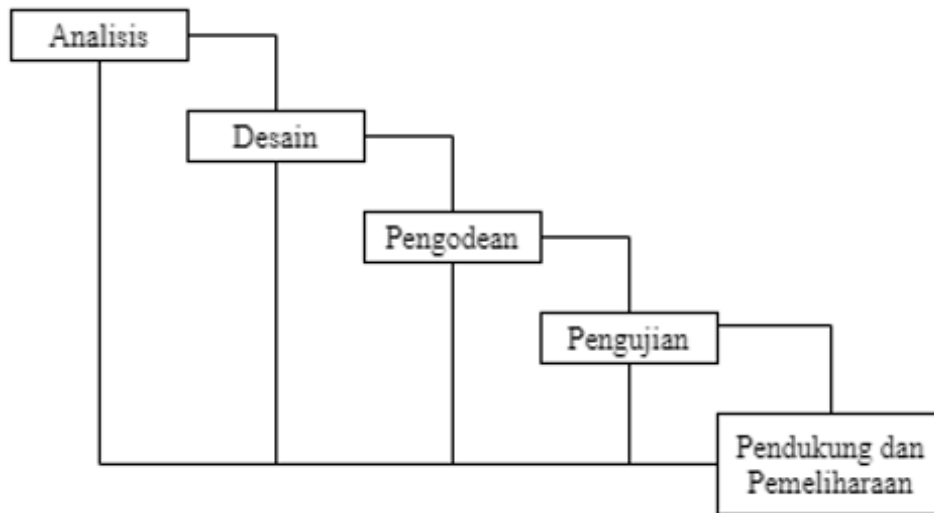
Application Programming Interface (API) merupakan software interface yang terdiri atas kumpulan instruksi yang membuat dua aplikasi yang berbeda platform dapat berkomunikasi satu sama lainnya[MuleSoft, “What is an API? (Application Programming Interface).” [Online]. Available: <https://www.mulesoft.com/resources/api/what-is-an-api>. [Accessed: 22-Mar-2019].]

Contoh dari penggunaan API adalah aplikasi pada platform mobile. Aplikasi mobile dapat menerima dan mengirim data ke server tanpa harus menulis aplikasi yang sama di sisi aplikasi mobile tersebut. Aplikasi mobile hanya perlu mengakses API yang sudah disediakan untuk mampu melakukan komunikasi data dengan server.

2.8 Metode Waterfall – Roger S

Metode air terjun atau yang sering disebut metode *waterfall* sering dinamakan siklus hidup klasik (*classic life cycle*), dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), permodelan (*modeling*), konstruksi (*construction*), serta penyerahan sistem ke para pelanggan/pengguna (*deployment*), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan (Pressman, 2012).

Tahapan metode *waterfall* dapat dilihat pada gambar di bawah ini.



Gambar 2.2 Metode Waterfall – Roger S

Penjelasan dari setiap tahapannya adalah sebagai berikut :

1. Analisis

Tahap ini bertujuan untuk memahami perangkat lunak yang diharapkan oleh pengguna dan batasan perangkat lunak tersebut. Informasi diperoleh melalui jurnal dan studi literatur. Informasi dianalisis untuk mendapatkan data yang dibutuhkan oleh pengguna.

2. Desain

Pada tahap Design ini, aplikasi yang akan dibangun adalah berbasis *Website*. Bahasa pemrograman yang digunakan dalam pembangunan aplikasi ini adalah Python. Database yang digunakan adalah MySQL.

3. Pengodean

Pada tahap ini, perancangan perangkat lunak direalisasikan sebagai serangkaian program atau unit program. Pengujian melibatkan verifikasi bahwa setiap fungsi memenuhi spesifikasinya

4. Pengujian

Tahap ini adalah tahap dimana aplikasi akan dilakukan pengujian internal apakah fungsionalitasnya sudah berjalan sesuai dengan yang diharapkan atau tidak.

5. Pendukung dan Pemeliharaan

Tahap akhir dalam model waterfall. Perangkat lunak yang sudah jadi, dijalankan serta dilakukan pemeliharaan. Pemeliharaan termasuk dalam memperbaiki kesalahan yang tidak ditemukan pada langkah sebelumnya. Perbaikan implementasi unit sistem dan peningkatan jasa sistem sebagai kebutuhan baru. Kesimpulan dan Saran[22].

2.9 Video Bloging

Vlog atau Video Blog, sesuai namanya adalah blog berbentuk video. Lebih jelasnya lagi, Vlog adalah satu video berisi mengenai opini, cerita atau kegiatan harian yang biasanya dibuat tertulis pada blog. Vlog ini bersifat untuk memberi informasi baik itu bersifat umum ataupun bersifat pribadi.

Vlog ialah salah satu genre komunikasi yang berpotensi untuk dikomentari, hingga mengundang kritik, perdebatan, serta diskusi interaktif. Beberapa Vlog juga disusun untuk saling bersahutan[23].

2.10 CyberStalking

Cyberstalking atau bisa disebut penguntitan dunia maya adalah serangan berbasis teknologi terhadap satu orang yang sudah menjadi sasaran karena alasan tertentu. Banyak bentuk dari *cyberstalking*, contohnya pelecehan, pembobolan rekening, menculik, dan banyak lagi kejahatan lainnya. *Cyberstalking* sering kali disertai dengan penguntitan waktu nyata atau offline[8]. *Cyberstalking* merupakan tindakan kriminal karena dapat merugikan orang lain.

Para penguntit melakukan tindakan ini secara terencana. Dari melihat seseorang yang dia kenal atau tidak, lalu mencari dengan menggunakan aplikasi. Dengan begitu data korban akan diketahui. Lalu penguntit mengambil data korban atau mengikuti korban selagi korban lengah penguntit akan bertindak[4].

2.11 Face Recognition

Pengenalan wajah merupakan sebuah sistem identifikasi pribadi yang menggunakan karakteristik wajah seseorang. Pengenalan wajah sendiri merupakan suatu cabang ilmu biometrik, yaitu suatu bidang keilmuan yang menggunakan karakteristik fisik dari seseorang untuk menentukan atau mengungkapkan identitasnya[24].

Evaluasi terbaru dari sistem pengenalan wajah komersial menunjukkan tingkat kinerja untuk verifikasi wajah dari sistem terbaik agar setara dengan pengenalan sidik jari untuk wajah depan yang diterangi secara seragam[25].

2.12 Open CV

Open Computer Vision (OpenCV) sendiri merupakan *library open source* yang tujuannya dikhususkan untuk melakukan pengolahan citra. Maksudnya adalah agar komputer mempunyai kemampuan yang mirip dengan cara pengolahan visual pada manusia. *OpenCV* telah menyediakan banyak algoritma visi komputer dasar. *OpenCV* juga menyediakan modul pendeteksian objek yang menggunakan algoritma Viola Jones[27]. *OpenCV* digunakan untuk pemrosesan dan penglihatan gambar, GUI, Struktur data, Gambar, dan I/O[28].

2.13 Image Processing

Image processing atau pengolahan citra merupakan suatu metode atau teknik yang digunakan untuk memproses citra atau gambar dengan memanipulasi menjadi data gambar untuk mendapatkan suatu informasi mengenai obyek yang sedang diamati[29].

Karena komputer merepresentasikan angka dengan menggunakan presisi hingga, angka-angka ini harus dikuantisasi agar dapat direpresentasikan secara digital. Pengolahan citra digital terdiri dari manipulasi angka presisi hingga tersebut[30].

2.14 Website

Website merupakan kumpulan dari halaman-halaman web yang berhubungan dengan file lain yang terkait. Home page adalah sebuah halaman yang pertama kali dilihat ketika seseorang mengunjungi website. Dari home page, pengunjung dapat mengklik hyperlink untuk ke halaman lain yang terdapat dalam website tersebut[31].

Website juga dapat dijadikan sarana untuk menampung aplikasi. Kelebihan menggunakan aplikasi website yaitu tidak perlu mengunduh seperti pengunduhan aplikasi native.

2.15 Laragon

Laragon adalah perangkat lunak yang memiliki bahasa pemrograman PHP, MySQL sebagai tempat penyimpanan database, dan apache sebagai web server yang digunakan untuk membangun local development environment pada Sistem Operasi windows[32].

Kelebihan menggunakan Laragon[32]:

1. Pretty URLs, project dapat diakses dengan app.test tanpa harus menggunakan localhost/app.
2. Portable, project dapat dipindahkan dengan mudahnya tanpa merusak sis
3. Isolated, sistem pada laragon terisolasi langsung dengan sistem operasi sehingga apa yang pengguna lakukan pada aplikasi ini tidak mempengaruhi komputer lokal pengguna.
4. Easy Operation, aplikasi ini otomatis memiliki banyak konfigurasi sehingga sangat mudah untuk digunakan.
5. Modern dan Powerful, aplikasi ini memiliki arsitektur yang modern sehingga mudah digunakan saat membangun web yang modern.

2.16 PHP

PHP merupakan suatu bahasa pemrograman Open Source yang digunakan secara luas terutama untuk mengembangkan web dan disimpan dalam bentuk HTML. PHP bersifat server side yang di dalam HTML artinya dalam suatu dokumen HTML dapat dimasukan skrip PHP[26].

Kelebihannya PHP lebih ringkas karena tidak perlu *mensetting* secara berlebihan. Karena konfigurasi pada database juga dapat dilakukan dengan lebih mudah. Bahasa pemrograman PHP ini open source, maka pengguna dapat menggunakannya dengan bebas dan gratis.

2.17 MySQL

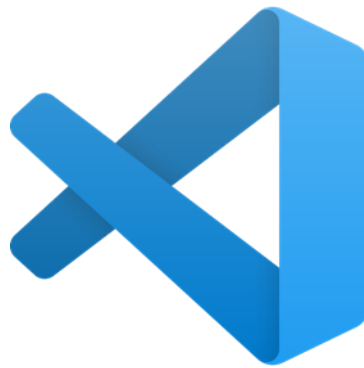
MySQL merupakan sebuah *database* yang dapat menerima dan mengirimkan data dengan sangat cepat, *multi-user*, serta menggunakan perintah standar SQL (*Structured Query Language*) baik digunakan *client server* maupun *server* []

- a. Kelebihan
 4. Bersifat *Open Source*.

5. Mendukung *Multi-User*.
 6. Mendukung integrase dengan Bahasa pemrograman lain.
 7. Tidak membutuhkan RAM besar.
 8. Struktur table yang fleksibel.
 9. Tipe data yang bervariasi.
- b. Kekurangan
1. *Technical Support* kurang bagus.
 2. Sulit mengelola *database* besar.

2.18 Visual Studio Code

Visual Studio Code atau biasa disebut VSCode merupakan source code yang dikembangkan oleh Microsoft untuk Windows, Linux dan MacOS. Ini termasuk dukungan untuk debugging, GIT Control yang disematkan, penyorotan sintaks, penyelesaian kode cerdas, cuplikan, dan kode refactoring. Hal ini juga dapat disesuaikan, sehingga pengguna dapat mengubah tema editor, shortcut keyboard, dan preferensi. Visual Studio Code gratis dan open-source.



Gambar 2.3 Visual Studio Code

2.19 Analisa dan Perancangan Berorientasi Objek

Dalam melakukan analisa dan perancangan sistem berorientasi objek penulis menggunakan UML (*Unified Modelling Language*) untuk pemodelannya. Sedangkan alat (*tool*) visual modeling yang akan digunakan untuk menggambarkan model analisa dan perancangan adalah Microsoft Visio 2007. Implementasi perangkat lunak menggunakan bahasa pemrograman PHP.

2.19.1 Unified Modelling Language

Duet mereka pada Oktober 1995 menghasilkan Unified Method versi 0.8, yang menjadi cikal bakal dari UML sebagai bahasa pemodelan standar untuk aplikasi object oriented.

Pada tahun 2002 lahirlah UML versi 2.0 dengan penambahan dan penggantian diagram menjadi 13 buah diagram. Diagram-diagram ini terbagi menjadi 3 kategori, yaitu:

- a. *Structural diagrams* : menggambarkan elemen dari spesifikasi yang mengabaikan waktu. Terdiri dari : *Class Diagram, Object Diagram, Component Diagram, Deployment Diagram, Composite Structure Diagram* dan *Package Diagram*.
- b. *Behavior Diagram* : menggambarkan ciri-ciri *behavior/method/function* dari sebuah sistem atau *business process*. Terdiri dari : *Use case Diagram, Activity Diagram, dan State Machine Diagram*.
- c. *Interaction Diagram* : bagian dari *behavior diagram* yang menggambarkan *object interactions*. Terdiri dari : *Communication Diagram, Interaction Overview Diagram, Sequence Diagram* dan *Timing Diagram*.

Karena UML sangat fleksibel, ada cara untuk melihat diagram UML berdasarkan kategori berikut :

- a. *Static Diagram* : menunjukkan segi statik dari sistem. Kategori ini sama dengan *structural diagram*.
- b. *Dynamic Diagram* : menunjukkan bagaimana sistem berkembang setiap waktu. Meliputi *state-machine diagram* dan *timing diagram*.
- c. *Functional Diagram* : menunjukkan detail dari perilaku (*behavior*) dan algoritma bagaimana sistem memenuhi perilaku yang diinginkannya.

Kategori ini termasuk *Use Case, interaction* dan *activity diagram*.

2.19.2 Use Case Diagram

Use case diagram menggambarkan kebutuhan sistem dari sudut pandang *user*. Digunakan untuk menggambarkan hubungan antara internal sistem dan eksternal sistem atau hubungan antara *use case* dan aktor[33].

1. Aktor

Aktor adalah sesuatu (entitas) yang berhubungan dengan sistem dan berpartisipasi dalam use case. Actor menggambarkan orang, sistem atau entitas eksternal yang secara khusus membangkitkan sistem dengan input atau masukan kejadian-kejadian, atau menerima sesuatu dari sistem. Actor dilukiskan dengan peran yang mereka mainkan dalam use case, seperti Staff, Kurir dan lain-lain.



Dalam use case diagram terdapat satu aktor pemulai atau initiator actor yang membangkitkan rangsangan awal terhadap sistem, dan mungkin sejumlah aktor lain yang berpartisipasi atau participating actor. Akan sangat berguna untuk mengetahui siapa aktor pemulai tersebut.

2. Use Case

Use case yang dibuat berdasarkan keperluan aktor merupakan gambaran dari “apa” yang dikerjakan oleh sistem, bukan “bagaimana” sistem mengerjakannya. *Use case* diberi nama yang menyatakan apa hal yang dicapai dari interaksinya dengan aktor.

Dalam UML *Use Case* dinotasikan dengan gambar.



3. Relationship

Relasi atau *Relationship* digambarkan sebagai bentuk garis antara dua simbol dalam use case diagram. Relasi antara actor dan use case disebut juga dengan asosiasi (*association*). Asosiasi ini digunakan untuk menggambarkan bagaimana hubungan antara keduanya.

Relasi- relasi yang terjadi pada *use case diagram* bisa antara *actor* dengan *use case* atau *use case* dengan *use case*.



Relasi antara *Use Case* dengan *Use Case*.

- a. *Include*, pemanggilan *use case* oleh *use case* lain atau untuk menggambarkan suatu *use case* termasuk di dalam *use case* lain (diharuskan). Contohnya adalah pemanggilan sebuah fungsi program. Digambarkan dengan garis lurus berpanah dengan tulisan <<include>>.
- b. *Extend*, digunakan ketika hendak menggambarkan variasi pada kondisi perilaku normal dan menggunakan lebih banyak *control form* dan mendeklarasikan *ekstension* pada *use case* utama. Atau dengan kata lain adalah perluasan dari *use case* lain jika syarat atau kondisi terpenuhi. Digambarkan dengan garis berpanah dengan tulisan <<extend>>.
- c. *Generalization/Inheritance*, dibuat ketika ada sebuah kejadian yang lain sendiri atau perlakuan khusus dan merupakan pola berhubungan *base parent use case*. Digambarkan dengan garis berpanah tertutup dari *base use case* ke *parent use case*.

2.19.3 Activity Diagram

Diagram aktivitas menggambarkan proses bisnis dan urutan aktivitas-aktivitas yang mendukung penggambaran tindakan sistem baik yang bersifat

kondisional maupun paralel. Tindakan kondisional dilukiskan dengan cabang (*branch*) dan penyatuan (*merge*).

Sebuah *branch* memiliki sebuah *transition* masuk atau yang disebut dengan *incoming transition* dan beberapa *transition* keluar atau yang disebut dengan *outgoing transition* dari *branch* yang berupa keputusan-keputusan. Hanya satu dari *outgoing transition* yang dapat diambil, maka keputusan-keputusan tersebut harus bersifat *mutually exclusive*. [*else*] digunakan sebagai keterangan singkat yang menunjukkan bahwa *transition* “*else*” tersebut harus digunakan jika semua keputusan yang ada pada *branch* salah.

Sebuah *merge* memiliki banyak *input transition* dan sebuah *output*. *Merge* menandakan akhir dari suatu kondisi yang diawali dengan sebuah *branch*. Selain *branch* dan *merge*, di dalam diagram aktivitas terdapat pula *fork* dan *join*. *Fork* memiliki satu *incoming transition* dan beberapa *outgoing transition*. Sedangkan pada *join*, *outgoing transition* diambil atau digunakan hanya ketika semua *state* pada *incoming transition* telah menyelesaikan aktivitasnya.

2.20 Sequence Diagram

Diagram yang menggambarkan bagaimana objek berinteraksi dengan objek lainnya melalui pesan (*message*) yang disampaikan, disusun dalam urutan kejadian atau waktu dan secara khusus berasosiasi dengan *use case*[33].

Tujuan dari penggunaan *sequence diagram* ini untuk mengkomunikasikan requirement kepada tim teknis karena diagram ini dapat lebih mudah untuk dielaborasi menjadi model design, mengembangkan model deskripsi use-case menjadi spesifikasi design, dan memfokuskan pada identifikasi method di dalam sebuah system.

2.21 Class Diagram

Class diagram merupakan bagian yang paling penting dalam analisa dan perancangan berorientasi objek. Dalam UML diagram kelas digunakan untuk memodelkan *static structure* dari sistem informasi[33].

Kelas merupakan himpunan dari objek yang sejenis yang mempunyai atribut dan perilaku (*behaviors/method*) yang sama. Atribut adalah sebuah nilai data karakteristik yang dimiliki oleh objek sebuah kelas sedangkan *method* adalah

perilaku atau operasi yang dikenakan oleh suatu kelas. Pada gambar kelas terdapat tiga bagiannya.

Diagram kelas menggambarkan struktur objek sistem, dimana diperlihatkan hubungan antar mereka. Diagram kelas (*Class Diagram*) merupakan pondasi untuk *component diagram* dan *deploelas* merupakan himpunan dari obyek yang sejenis yang mempunyai atribut dan perilaku (behaviors/method) yang sama. Atribut adalah sebuah nilai *oyment diagram*.

Secara garis besar terdapat 3 jenis *Class*. Ketiganya dikelompokkan berdasarkan fungsi dan karakternya masing-masing, yaitu :

a. *Entity Class Diagram*

Merupakan paket utama dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk sistem dan menjadi landasan untuk menyusun basis data pada model data konseptual.



b. *Control Class Diagram*

Berisi kumpulan kelas yang menjadi kontrol program termasuk koneksi dengan basis data dan merupakan kelas perantara atau penghubung antara *entity class* dengan kelas antar muka pemakai (*interface*).



c. *Boundary Class Diagram*

Berisi kumpulan kelas yang menjadi *interface* antara pemakai (*user*) dengan sistem, seperti tampilan form untuk pencetakan.



Umur	Apakah akurasi deteksi dan manipulasi wajah cukup akurat?	Apakah aplikasi ini dapat membantu para editor menyamarkan wajah secara otomatis?	Apakah aplikasi ini mudah digunakan?	Apakah hasilnya sesuai dengan yang di inginkan?
21-25	Sangat Setuju	Sangat Setuju	Sangat Setuju	Setuju
21-25	Setuju	Ragu-Ragu	Setuju	Setuju
21-25	Ragu-Ragu	Setuju	Setuju	Setuju
26-30	Setuju	Setuju	Ragu-Ragu	Setuju
21-25	Setuju	Sangat Setuju	Sangat Setuju	Sangat Setuju
21-25	Setuju	Setuju	Setuju	Setuju
21-25	Setuju	Sangat Setuju	Sangat Setuju	Setuju
21-25	Setuju	Setuju	Setuju	Ragu-Ragu
21-25	Setuju	Setuju	Setuju	Setuju
21-25	Setuju	Sangat Setuju	Sangat Setuju	Setuju
21-25	Setuju	Setuju	Sangat Setuju	Setuju
21-25	Setuju	Setuju	Setuju	Setuju
21-25	Setuju	Setuju	Setuju	Setuju
15-20	Setuju	Setuju	Setuju	Setuju
21-25	Setuju	Setuju	Setuju	Setuju
31-35	Setuju	Setuju	Setuju	Setuju
26-30	Setuju	Ragu-Ragu	Ragu-Ragu	Setuju
21-25	Setuju	Setuju	Sangat Setuju	Setuju
21-25	Ragu-Ragu	Setuju	Ragu-Ragu	Setuju
15-20	Setuju	Setuju	Sangat Setuju	Setuju
15-20	Setuju	Setuju	Setuju	Setuju
21-25	Setuju	Sangat Setuju	Sangat Setuju	Setuju
15-20	Ragu-Ragu	Setuju	Sangat Setuju	Ragu-Ragu
21-25	Ragu-Ragu	Sangat Setuju	Sangat Setuju	Ragu-Ragu
21-25	Setuju	Sangat Setuju	Sangat Setuju	Setuju
21-25	Setuju	Sangat Setuju	Sangat Setuju	Ragu-Ragu