BAB II

LANDASAN TEORI

2.1. Penelitian sebelumnya yang berkaitan dengan topik terkait

2.1.1. Sistem Deteksi Wajah Dengan Modifikasi Metode Viola Jones

Menurut Adinda Rizkita Syafira, deteksi wajah manusia merupakan salah satu topik yang paling banyak dipelajari di bidang *computer vision*. Tujuan deteksi wajah adalah untuk mengetahui ada atau tidaknya wajah pada suatu gambar. Meskipun tampak mudah dilakukan manusia, ternyata pendeteksian wajah sangat rumit dilakukan oleh computer karna terdapat beberapa kesulitan yang terkait dengan lokasi, sudut pandang, cahaya dan oklusi. Penelitian ini

Penelitian ini menerapkan metode Viola Jones untuk membangun sistem deteksi wajah dengan bahasa pemrograman Python. Metode Viola Jones merupakan salah satu metode deteksi wajah dengan tingkat akurasi yang tinggi dan komputasi yang cepat. Metode Viola Jones menggunakan fitur Haar sebagai deskriptor kemudian menggabungkan Integral Image dan AdaBoost untuk mencari dan melakukan seleksi nilai fitur dan membentuk Cascade Classifier. Classifier tersebut yang akan digunakan untuk mendeteksi wajah pada gambar. Penelitian ini juga mengevaluasi tingkat akurasi sistem dengan cara memodifikasi nilai-nilai parameter yang ada di metode Viola Jones. Dari hasil pengujian menggunakan K-fold cross validation didapat hasil akurasi tertinggi sebesar 90,9% untuk gambar wajah dan 75,5% untuk gambar bukan wajah.

2.1.2. Sistem Pengenalan Wajah dengan Algoritma *Haar Cascade* dan *Local Binary Pattern Histogram*

Haar-Like Feature merupakan fitur yang digunakan dalam metode Viola-Jones atau dapat disebut fitur gelombang tunggal bujur sangkar (interval tinggi dan interval rendah), dan dua dimensi disebut sebagai terang dan gelap. Cara menentukan *Haar-Like Feature* dengan mengurangi rata-rata piksel pada daerah gelap dari rata-rata piksel pada daerah terang.

Menurut Sayeed Al-Aidid, Daniel S. Pamungkas, "Sistem pengenalan wajah dengan algoritma *haar cascade dan local binary pattern histogram*" (2018), secara umum haar-like feature digunakan untuk mendeteksi objek pada image digital dengan memproses gambar dalam kotak-korak, dimana dalam satu kotak terdapat beberapa pixel kemudian diproses dan menghasilkan perbedaan nilai yang menandakan daerah gelap dan terang yang nantinya dijadikan dasar pemrosesan gambar. Contoh *Haar-Like Feature* disajikan dalam Gambar 2.2

Gambar 2.1 Haar-Like Feature (Viola, 2004)

Nilai *Haar-Like Feature* diperoleh dari selisih jumlah nilai piksel daerah gelap dan jumla nilai piksel daerah terang :

$$\textbf{F Harr} = \sum \textbf{F white} - \sum \textbf{F Black} \qquad \textbf{(2.1)}$$

F Harr = Nilai fitur total

$\sum$ F White = Nilai fitur pada daerah terang

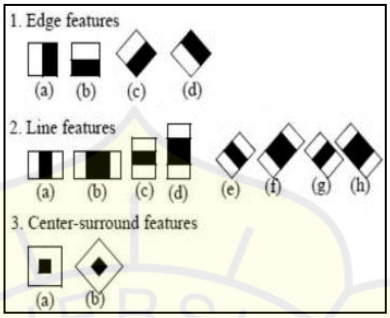
$\sum$ F Black = Nilai fitur pada daerah gelap

Setiap *Haar-Like Feature* terdiri dari gabungan kotak-kotak hitam dan putih. Ada tiga tipe kotak *feature* dalam *Haar*:

a. Tipe two-rectangle feature (horizontal, vertikal)

b. Tipe three-rectangle feature

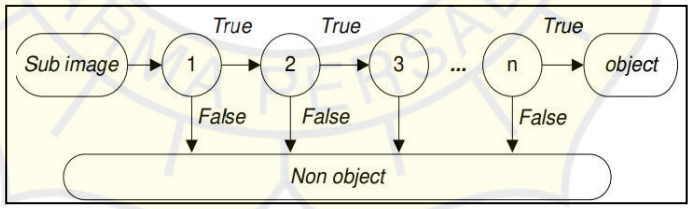
c. Tipe four-rectangle feature



Gambar 2.2 Variasi Fitur pada Haar (LienHart et al, 2002)

2.1.3. Cascade Classifier

Cascade classifier merupakan metode yang digunakan untuk mengkombinasikan classifier yang kompleks pada struktur bertingkat yang dapat meningkatkan kecepatan dan keakuratan pendeteksian objek dengan memfokuskan pada daerah citra yang berpeluang saja. Struktur casecade classifier seperti berikut.



Gambar 2.3 Cascade Clasifier (Aries Suharso,2016)

Dapat dijelaskan pada Gambar 2.3 proses penyeleksian objek yang diasumsikan sebagai sub image di evaluasi oleh classifier pertama, jika berhasil melewati classifier pertama maka mengindikasikan sub image berpotensi terkandung objek dan dapat dilanjutkan pada classifier selanjutnya sampai melewati keseluruhan

classifier, maka dapat disimpulkan terdapat objek tersebut terdeteksi. jika tidak, proses evaluasi tidak dilanjutkan dan disimpulkan tidak terdapat objek.

Menurut Viola, Paul dan Michael Jones "*Rapid Object Detection using a Boosted Cascade of Simple Features*" (2001), Cascade classifier adalah sebuah rantai stage classifier, dimana setiap stage classifier digunakan untuk mendeteksi apakah didalam image sub window terdapat obyek yang diinginkan (object of interest). Stage classifier dibangun menggunakan algoritma adaptive-boost (AdaBoost). Algoritma tersebut mengkombinasikan performance banyak weak classifier untuk menghasilkan strong classifier. Weak classifier dalam hal ini adalah nilai dari haar-like feature.

2.1.4. A comparative study on Convolutional Neural Network and Viola-Jones Algrithm for faster Face Recognition

Menurut Rubeena M (2020), dalam penelitiannya yang berjudul "*A comparative study on Convolutional Neural Network and Viola-JonesAlgrithm for faster Face Recognition",* Meskipun terdapat kesamaan antara algoritme Viola-Jones dan Convolutional Neural Networks, keduanya berbeda di beberapa area. Algoritma Viola-Jones tidak mampu mendeteksi wajah pada berbagai posisi. Jika wajah tidak ditampilkan dalam posisi menghadap ke depan dengan pencahayaan yang tepat, keakuratan hasil akan menurun drastis. Sementara kedua metode bekerja dalam serangkaian langkah, langkah-langkah dalam metode Viola-Jones diatur sedangkan langkah-langkah dalam CNN jauh lebih terstruktur. Algoritme Viola-Jones menemui masalah dalam rentang posisi seperti tampilan samping dan untuk pencahayaan redup karena fitur mirip Haar mereka tidak memetakan dengan baik ke berbagai posisi. Sebaliknya, CNN memiliki kemampuan untuk mendeteksi

wajah di posisi yang berbeda termasuk tampilan samping dan skenario pencahayaan yang berbeda, dan oleh karena itu CNN jauh lebih beragam dalam menangani masukan dengan benar. Convolutional Neural Networks membutuhkan beberapa pass through dan karenanya menyimpan informasi dalam jumlah besar, CNN membutuhkan lebih banyak ruang secara lokal daripada algoritma Viola-Jones.

2.2. Sistem Presensi

Absensi merupakan kegiatan atau rutinitas yang dilakukan oleh karyawan suatu instansi untuk membuktikan dirinya hadir atau tidak hadir dalam bekerja sebagai penerapan disiplin untuk pegawai.

Menurut Eko Budi Setiawan dan Bobi Kurniawan (2015), suatu pendataan kehadiran merupakan bagian dari aktifitas pelaporan yang ada pada suatu instansi. Pencatatan absensi disusun dan diatur sehingga mudah untuk dicari dan dipergunakan ketika diperlukan.

2.3. Definisi Citra

Citra (*image)* adalah suatu gambar pada bidang dua dimensi yang disusun dari banyak *pixel (Picture Element)* yang menyusun sebuah foto digital. Setiap *pixel* membawa informasi yang dapat menunjukan warna *(hue)*, kekuatan warna *(satruation)*, dan intensitas cahaya *(brightness)*. Sebuah gambar dapat disebut dengan citra bila gambar yang diperoleh merupakan hasil proses pada kamera, komputer, *scanner* atau perangkat citra lainnya.

Menurut Indra Wijaya, dkk (2017) dalam penelitiannya yang berjudul ” Implementasi Raspberry Pi untuk Rancang Bangun Sistem Keamanan Pintu Ruang Server dengan Pengenalan Wajah Menggunakan Metode *Triangle Face*”, Citra *(image)* merupakan salah satu komponen multimedia yang memegang peranan

penting sebagai bentuk informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu citra kaya dengan informasi, Citra adalah suatu representasi, kemiripan, atau imitasi dari suatu objek.

2.4. Sistem Pengenalan Wajah *(Face Recognition)*

Dalam penelitian oleh Ayush Bijoura, dkk (2020) dengan judul "*Facial Recognition Using Deep Learning"* menjelaskan bahwa *face recognition* dapat didefinisikan sebagai proses mengidentifikasi dan memverifikasi orang dalam foto dengan wajah mereka. Proses ini terdiri dari deteksi, penyelarasan, ekstraksi fitur, dan tugas pengenalan dalam program komputer yang membantu dalam menemukan wajah di foto.

Menurut Sanjaya (2018), Pendeteksi wajah yang digunakan pada penelitian ini menggunakan metode yang digunakan oleh *Paulus Viola* dan *Michael Jones* yang diterbitkan pada tahun 2001. Metode tersebut biasanya disebut sebagai metode Viola Jones, pendekatan yang dilakukan untuk mendeteksi objek-objek dalam suatu image adalah dengan menggabungkan tiga konsep kunci berikut, yaitu:

1. Mengintegrasi deteksi *Integral Image* secara cepat.
2. Pendeteksian data sampel dengan *Haar Fiture*.
3. Sebuah *Cascade Classifier* yang menggabungkan banyak fitur secara efisien.

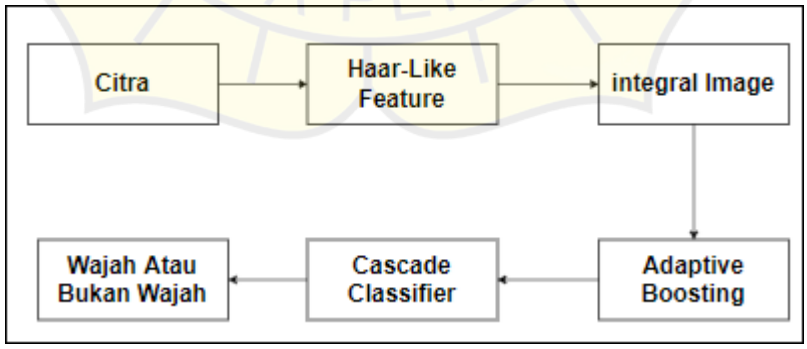
Hal terpenting dalam mengenali wajah adalah seberapa jauh jarak kamera untuk mengenali antara mata, dan berapa lebarnya dan panjang hidung, lebar dagu, hal-hal lain. Setelah ini, data yang diperoleh dibandingkan dengan data-data yang tersedia dalam database, semua parameter ini cocok, kemudian orang tersebut dapat

ter-identifikasi. Dengan cara ini dapat digunakan keamanan untuk menentukan orang tersebut berwenang atau tidak.

2.5. Sistem Kerja Algoritma Viola Jones

Viola Jones adalah algoritma pendeteksian objek menggunakan pembelajaran mesin yang digunakan untuk mengidentifikasi objek dalam gambar atau video, dan berdasarkan konsep fitur dari Paul Viola dan Michael Jones dalam makalah mereka "*Rapid Object Detection Using a Boosted Cascade of Simple Features*"(2001), melalui pendekatan berbasis pembelajaran mesin atau *mechine learning* dimana fungsi *cascade* dilatih dari banyak gambar positif dan negatif dan digunakan untuk mendeteksi objek dalam gambar lain.

Karakteristik dari algoritma *Viola-Jones* adalah dengan adanya klasifikasi bertingkat yang terdiri dari tiga tingkatan yang dari setiap tingaktan menghasilkan *output*s subcitra yang diyakini bukan wajah. Hal ini dilakukan sebab tidak sulit untuk menentukan subcitra yang bukan wajah dibandingkan menilai apakah subcitra tersebut merupakan wajah. Alur kerja dari klasifikasi bertingkat dapat digambarkan pada gambar berikut.



Gambar 2.4 algoritma viola-jones (Aries Suharso,2016)

Berikut ini penjelasan skema dari proses yang dilakukan dalam pendeteksian target wajah dengan algoitma metode *Viola-Jones:*

1. Pada tahap pertama, tiap subcitra akan proses membaca sampel citra wajah yang menghadap ke kamera.
2. Hasil tahap pertama berupa citra wajah akan dibaca dan ditentukan *Haar-Like Feature* untuk mendeteksi objek pada citra digital dan diproses menjadi beberapa kotak yang terdiri dari beberapa *pixel.* Selanjutnya *pixel* yang diproses akan menghasilkan nilai ambang *(threshold)* yang mengindikasikan daerah gelap dan daerah terang untuk dipakai sebagai dasar dalam proses pengolahan citra (*image processing*).
3. Kemudian pada tahap kedua citra akan diproses *Integral Image* untuk menyatakan ada atau tidak fitur *Haar-like feature* dalam citra pada skala yang berbeda secara efisien. Nilai *integral image* pada tiap-tiap pixel adalah hasil penjumlahan dari semua pixel-pixel dari atas sampai ke bawah. Penjumlahan diawali dari pojok kiri atas sampai pojok kanan bawah.
4. Selanjutnya dalam tahap ini digunakna sebuah metode *machine learning* untuk melakukan *supervised learning* yang disebut *Adaptive Boosting* atau disingkat *AdaBoost. AdaBoost* menggabungkan beberapa *classifier* lemah menjadi *classifier* yang lebih kuat dan menjadikan *AdaBoost classifier* menjadi rangkaian filter yang akan cukup efektif melakukan proses penggolongan pada daerah citra. Selama proses filter, jika sebuah filter gagal dalam meloloskan suatu objek citra, maka objek tersebut langsung diklasifikasikan sebagai bukan wajah. Namun jika filter berhasil meloloskan semua proses pada rangkaian filter, maka dapat didapati bahwa objek tersebut sebagai wajah.
5. Tahapan berikutnya untuk citra yang memenuhi proses klasifikasi *AdaBoost*

maka akan di filter pada *Casecade Classifier* dengan nilai bobot yang diberikan pada proses *AdaBoost*. Filter yang memiliki nilai bobot paling besar akan diletakan pada urutan pertama dengan tujuan untuk menghapus daerah citra secepat mungkin.

6. Tahap terakhir adalah menampilkan objek citra tersebut apakah terdeteksi sebagai wajah atau bukan wajah x adalah dimensi sub-image, misal 24x24.

2.6. HTML (Hyper Text Markup Language)

HTML yang merupakan singkatan dari *Hyper Text Markup Language* adalah serangkaian kode program yang merupakan dasar dari representasi visual sebuah halaman Web. Didalam HTML berisi informasi yang disimpan dalam *tag-tag* tertentu yang digunakan untuk melakukan format pada informasi yang dimaksud.

Menurut Ahmad Hidayat, Faisal (2019:3) HTML *(Hypertext Markup Language)* adalah merupakan sebuah bahasa pemrograman terstruktur yang dikembangkan untuk membuat halaman *website* yang dapat diakses dan ditampilkan menggunakan web browser. HTML terdiri dari tag-tag yang mendefinisikan elemen tertentu pada sebuah halaman *website*.

2.7. Javascript

Bahasa yang digunakan untuk meningkatkan interaksi pengguna karena dapat membuat halaman *web* menjadi terlihat lebih interaktif dan menarik merupakan definisi dari *Javascript*.

Menurut Ade Suryadi, Yuli Siti Zulaikhah “Rancang bangun sistem pengelolaan arsip surat berbasis web menggunakan metode *waterfall*”(2019:15), *Javascript* merupakan skrip paling banyak digunakan dalam pemrograman *web*

pada sisi *client*. Dengan adanya *javascript* sebuah web akan menjadi lebih hidup, cepat dan tampil lebih menawan dengan sebuah grafik animasi"

2.8. *Boostrap*

Menurut Jodi Martin, Andeka Rocky Tanaamah "Perancangan dan implementasi sistem informasi penjualan berbasis desktop website menggunakan *framework bootstrap*" (2017:59), *Boostrap* merupakan gabungan jenis *framework* dari *CSS* dan *javascript* yang berisikan *script* untuk membangun sebuah *website* secara mudah dan responsif. *Bootstrap* terdiri dari CSS dan HTML untuk menghasilkan *grid, layout, typography, table, form, navigation,* dan lain-lain. *Boostrap* sendiri sudah menjadi aplikasi *open-source* agar dapat digunakan oleh siapa saja tanpa harus membeli lisensi dan dapat berjalan pada web browser umum seperti *Chrome, Firefox, Safari Opera, dan Internet Explorer.*

2.9. PHP

PHP merupakan bahasa pemrograman yang ditujukan untuk membuat aplikasi berbasis web. Ditinjau dari pemrosesannya, PHP tergolong sebagai server side, yaitu pemrosesan yang dilakukan di server.

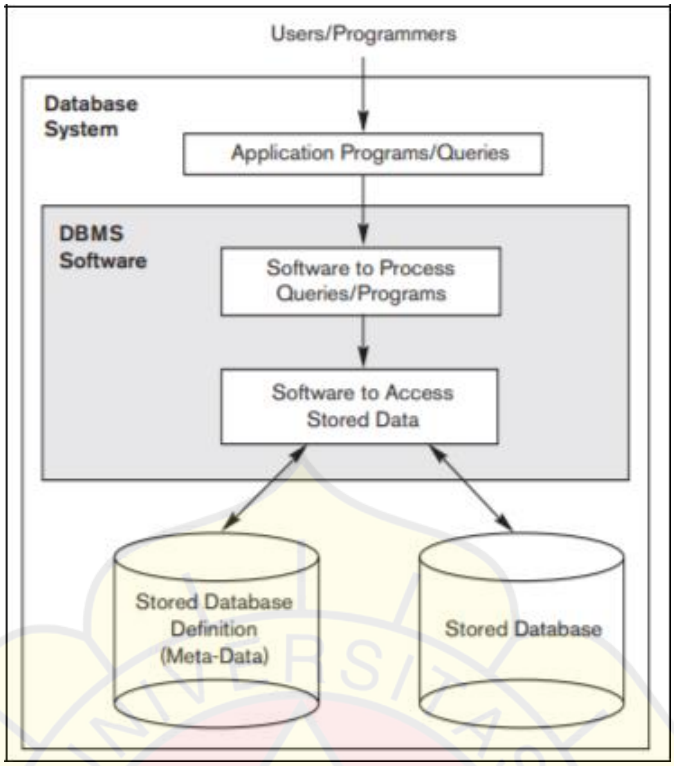
Menurut Supono dan Vidianry Putratama dalam buku yang berjudul "Pemrograman WEB dengan menggunakan PHP dan *framework codeigniter*" (2018, 3), PHP (Hypertext *Preprosesor*) merupakan bahasa pemrograman untuk menerjemahkan kode pemrograman menjadi kode mesin yang di mengerti komputer dan bersifat *server-side* yang berarti pengerjaan kode program dilakukan di server kemudian hasilnya akan dikirimkan ke browser sehingga halaman web tidak lagi berisfat statis namun menjadi dinamis dan dapat ditambahkan ke dalam HTML

2.10. Database Management System

Menurut Ramez Elmasri, Shamkant B. *"Fundamentals of Database System 6th edition"* (2011, 4) Basis data merupakan kumpulan data terkait dengan data dan fakta yang diketahui akan direkam dan memiliki makna yang implisit.

Dalam implementasi basis data, dikenal istilah query yaitu pencarian informasi tertentu yang spesifik. Bahasa query paling umum digunakan adalah SQL. SQL atau *Standard Query Language* membangun dirinya sebagai bahasa relational-database standar. Ada beberapa versi SQL. Salah satunya dibuat oleh IBM yang awalnya dikenal dengan nama *Sequel*. Bahasa Sequel mengalami perkembangan dan perubahan nama menjadi *Structured Query Language*. Pada tahun 1986, ANSI (*American National Standards Institute*) dan ISO (*International Standars Organization*) mempublikasikan SQL standar, yaitu SQL 86 lalu pada tahun-tahun berikutnya peningkatan terhadap SQL dilakukan hingga mencapai versi terbaru yaitu SQL 2000.

Menurut Ramez Elmasri, Shamkant B. *"Fundamentals of Database System 6th edition"* (2011, 5) *Database Management System* (DBMS) merupakan kumpulan program yang digunakan pengguna untuk membuat dan memelihara *database*. *Database Management System* adalah sistem perangkat lunak untuk keperluan umum seperti membuat, mengubah dan berbagi antar pengguna aplikasi. Membangun database merupakan proses menyimpan data pada beberapa media penyimpanan pada DBMS. Mengubah *database* mengambil, memperbarui *database* untuk menghasilkan laporan dari data. berbagi *database* memungkinkan banyak pengguna dan program mengakses *database* diwaktu yang bersamaan dalam DBMS.



Gambar 2.5 Lingkup Basis Data ((Elmasri & B.Navathe, 2011)

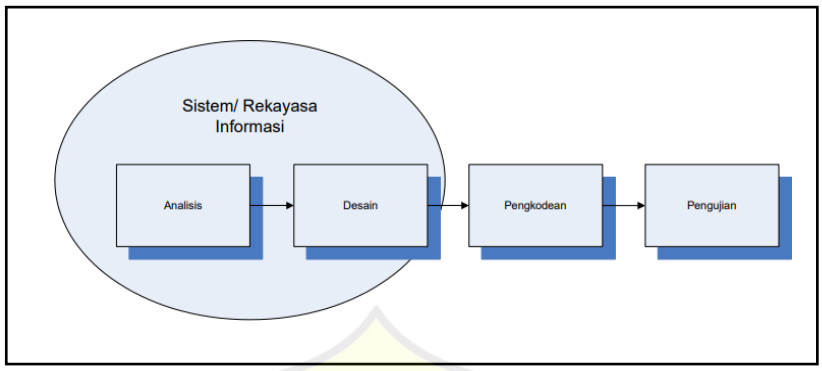
2.11. MySQL

Menurut Supono dan Vidianry Putratama dalam buku yang berjudul ”Pemrograman WEB dengan menggunakan PHP dan *framework codeigniter*” (2018, 96), *MySQL* adalah sistem manajemen *databse* yang bersifat *Open Source* dan mendukung beberapa fitur sepertu *multitrhearded, multiuser dan Database Management System* (DBMS) untuk keperluan sistem *database* yang cepat, andal dan mudah digunakan.

2.12. Model Waterfall

Menurut Juniardi Dermawan, Sari Hartini (2017:143) ”Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linear*) dan juga sering disebut dengan alur hidup klasik (*classic life cycle*). Model air terjun ini

menyediakan pendekatan alur hidup perangkat lunak secara terurut dimulai dari analisis, desain, pengodean, pengujian, dan tahap pendukung (*support*)".



Gambar 2.6 Metodologi Waterfall (Juniardi Dermawan, Sari Hartini ,2017)

Metode *Waterfall* memiliki tahapan-tahapan sebagai berikut :

1. Analisis Kebutuhan Perangkat Lunak

   Proses pengumpulan kebutuhan dilakukan secara matang untuk menyesuaikan kebutuhan perangkat lunak agar dapat dipahami perangkat lunak seperti apa yang dibutuhkan oleh user.

2. Desain

   Desain perangkat lunak adalah proses multi langkah yang berfokus pada desain pembuatan program perangkat lunak termasuk sturuktur data, arsitektur perangkat lunak, representasi antarmuka, dan prosedur pengodean. Tahap ini mentranslasi kebutuhan perangkat lunak dari tahap analisis kebutuhan ke representasi desain agar dapat diimplementasikan menjadi program pada tahap selanjutnya.

3. Pembuatan Kode Program

   Desain harus ditranskasikan ke dalam program perangkat lunak. Hasil dari tahap ini adalah program computer sesuai dengan desain yang telah dibuat pada tahap desain.

4. Pengujian

   Pengujian fokus pada perangkat lunak secara dari segi *logic* dan fungsional dan memastikan bahwa semua bagian sudah diuji. Hal ini dilakukan untuk meminimalisir kesalahan (*error*) dan memastikan keluaran yang dihasilkan sesuai dengan yang diinginkan.

5. Pendukung (*Support*) atau Pemeliharaan (*Maintenance*)

   Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan ke *user*. Perubahan bisa terjadi karena adanya kesalahan pada perangkat lunak yang mucul dan tidak terdeteksi padasaat pengujian. Tahap pendukung atau pemeliharaan dapat mengulangi prosespengembangan perangkat lunak yang sudah ada, tapi tidak untuk membuat perangkat lunak yang baru.

2.13. UML

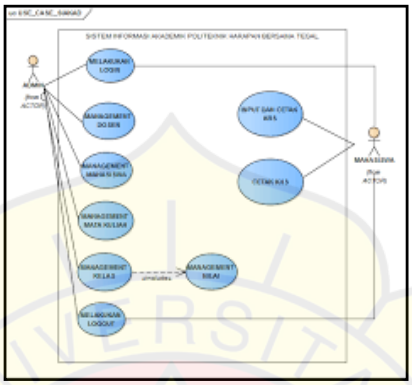
Menurut M Teguh Prihandoyo (2018:127) "*Unifield Modeling Language* merupakan salah satu metode pemodelan visual yang digunakan dalam perancangan dan pembuatan sebuah *software* yang berorientasikan pada objek .UML merupakan sebuah standar penulisan atau semacamnya blue print dimana didalamnya termasuk sebuah bisnis proses, penulisan kelas-kelas dalam sebuah bahasa yang spesifik". Terdapat beberapa jenis-jenis diagram di dalam UML sebagai berikut:

2.13.1. Use Case Diagram

Menurut M Teguh Prihandoyo (2018:127) "*Use Case* Merupakan gambarandari fungsionalitas yang di harapkan dari sebuah sistem dan merepresentasikan sebuah interaksi antara aktor dan sistem. Didalam *use case* terdapat aktor yang merupakan

pekerjaan di sistem". Diagram *Use Case* bersifat statis dan sangat penting untuk mengorganisasi dan memodelkan perilaku sistem yang dibutuhkan serta diharapkan pengguna.

Contoh *Diagram Use Case* :

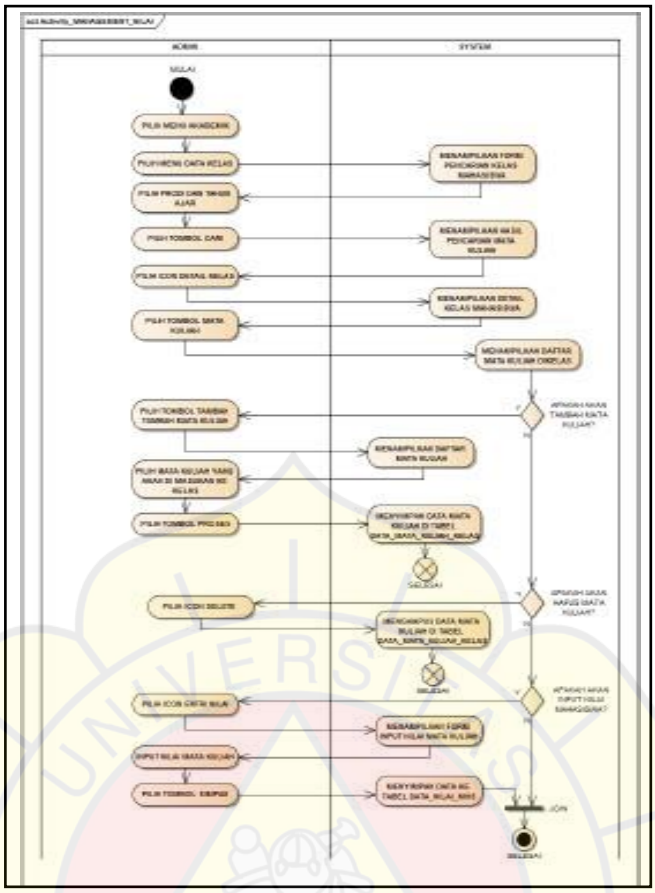


Gambar 2.7 *Diagram Use Case* (M Teguh Prihandoyo, 2018)

2.13.2. Activity Diagram

Menurut M Teguh Prihandoyo (2018:127) "*Activity* Diagram Merupakan gambaran air dari aktivitas-aktivitas didalam sistem yang berjalan". *Activity* diagram ini bersifat dinamis dalam pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek.

Contoh *Diagram Activity*:

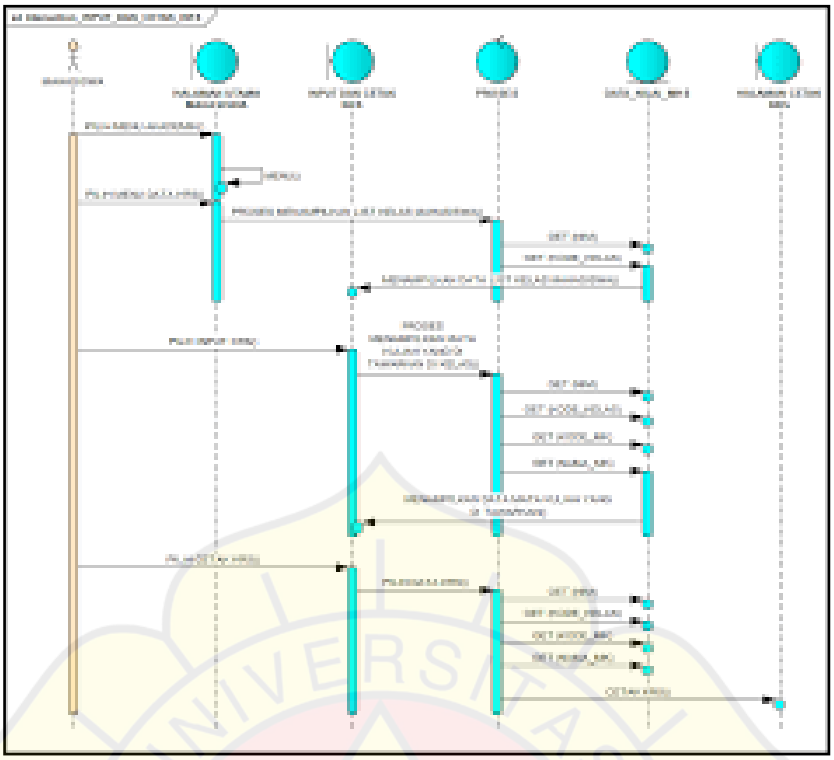


Gambar 2.8 *Diagram Activity* (M Teguh Prihandoyo, 2018)

2.13.3. Sequance Diagram

Menurut M Teguh Prihandoyo (2018:127) "*Sequence* Diagram adalah Menggambarkan interaksi antar objek didalam dan di sekitar sistem yang berupa message yang digambarkan terhadap waktu". *Sequence* Diagram bisa menggambarkan aktivitas objek pada *use case* dengan mendeskripsikan waktu hidup objek serta pesan yang dikirimkan serta diterima antar objek.

Contoh *Diagram Sequence*:



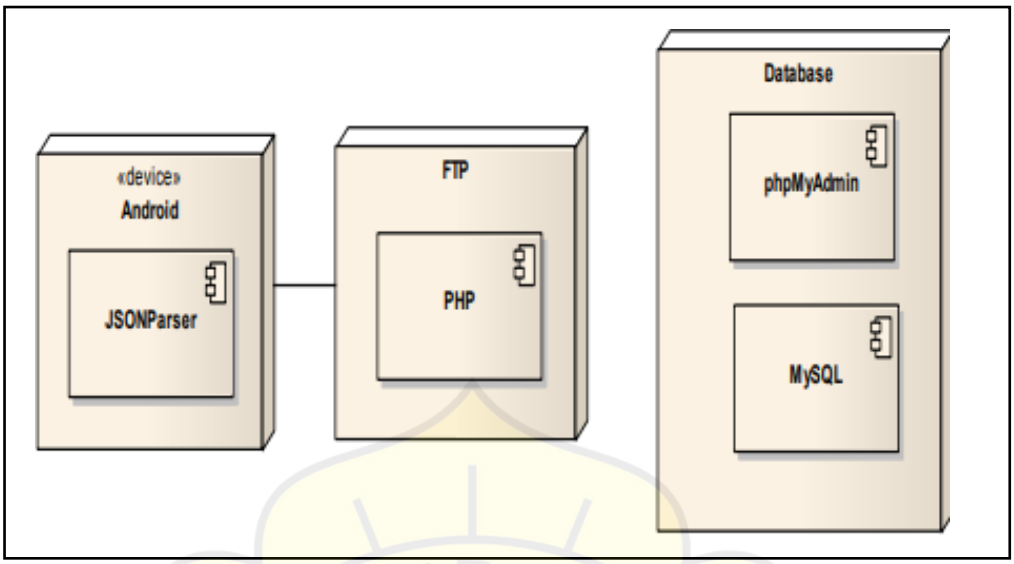
Gambar 2.9 *Diagram Sequence* (M Teguh Prihandoyo, 2018)

2.13.4. Deployment Diagram

Menurut Kundang Karsono (2016:4) "*Deployment* Diagram adalah penggambaran tugas-tugas kongkrit dari setiap *node/software* yang terlibat dalam jaringan sistem, menampilkan keseluruhan *node* dalam jaringan serta hubungan dari *node-node* tersebut termasuk proses-proses yang terlibat didalamnya".

Diagram Deplomen (*Deployment* Diagram) ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*). Menurut simpul-simpul beserta komponen-komponen yang ada di dalamnya. Diagram *deployment* berhubungan erat dengan diagram komponen dimana diagram ini memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing*).

Contoh *Diagram Deployment*:



Gambar 2.10 *Diagram Deployment* (Kundang Karsono, 2016)

2.13.5. Implementasi & Testing

Tahapan inilah ialah tahapan secara nyata dalam mengerjakan sesuatu sistem.Dalam artian pemakaian komputer hendak dioptimalkan dalam tahapan ini. Sehabis pengkodean berakhir hingga akan dicoba testing terhadap sistem yang sudah terbuat. Tujuan testing merupakan mendeteksi kesalahan-kesalahan terhadap sistem tersebut serta setelah itu dapat diperbaiki.

TEKNOLOGI INFORMASI

UNIVERSITAS DARMA PERSADA