BAB II

LANDASAN TEORI

2.1 Sistem Presensi

Absensi merupakan kegiatan atau rutinitas yang dilakukan oleh pegawai 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 adan pada suatu instansi. Pencatatan absensi disusun dan diatur sehingga mudah untuk dicari dan dipergunakan ketika diperlukan.

2.2 Wajah

Wajah merupakan bagian dari tubuh manusia yang selalu diperhatikan ketika melakukan interaksi dengan orang lain. Menurut kamus besar bahasa indonesia wajah merupakan bagian dari kepala; roman muka; raut muka. Biasanya wajah dipergunakan untuk mengekspresikan diri, penampilan dan identias seseorang.

Wajah dapat dikenali menggunakan teknologi biometrik dengan ciri-ciri fisik manusia seperti wajah dapat dipakai sebagai informasi berupa karakteristik pola wajah setiap individu dari pola wajah yang dapat diukur dan dianalisis untuk proses deteksi wajah.

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 Apriyana, dkk (2013) 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.

2.3.1 Pengolahan Citra

Menurut Adam Arif Budiman, Kevin Andreas Surbakti (2019) dalam “Perancangan aplikasi pengenalan objek menggunakan *image processing haar cascade* pada *Unmaned Aerial Vehicle* (UAV) untuk bencana alam”, *Image Processing* / Pengolahan citra merupakan suatu bentuk pengolahan atau pemrosesan sinyal dengan input berupa gambar *(image)* dan di transformasikan menjadi gambar lain sebagai keluarannya dengan teknik tertentu untuk memperbaiki kesalahan data sinyal gambar akibat dari transmisi dan akusisi sinyal, serta meningkatkan kualitas penampakan gambar agar mempermudah penglihatan manusia untuk menginterpretasikan dengan baik dan menganalisisan gambar.

Pengolahan citra merupakan suatu proses pengolahan citra atau menggunakan komputer agar sebuah citra dapat memiliki kualitas lebih baik dari sebelumnya. Pengolahan citra dapat memperbaiki kualitas suatu gambar atau citra sehingga dapat di interpretasikan dengan mudah oleh manusia dan komputer.

Menurut Apriyana, dkk (2013) ”Implementasi Raspberry Pi untuk Rancang Bangun Sistem Keamanan Pintu Ruang Server dengan Pengenalan Wajah Menggunakan Metode Triangle Face”, Pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik. Dengan istilah lain bahwa pengolahan citra merupakan proses pengolahan dan analisis citra yang banyak melibatkan persepsi visual. Selain itu tujuan pengolahan citra dikembangkan untuk :

1. Memperbaiki tampilan citra *(image enhancement)*

2. Mengurangi ukuran *file* citra dengan tetap mempertahankan kualitas citra *(image compression)*

3. Memulihkan citra ke kondisi semula *(image restoration)*

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*.

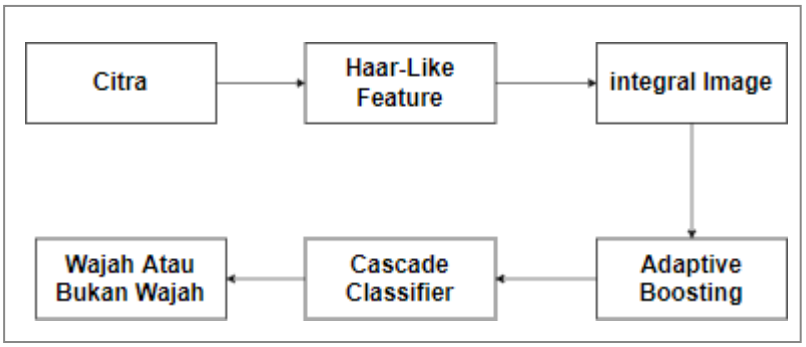
3. Sebuah *Cascade Classifier* yang menggabungkan banyak fitur secara efisien.

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 2.1 *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.

2.5.1 Haar-Like Feature

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 2.2 *Haar-Like Feature (Viola, 2004)*

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

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

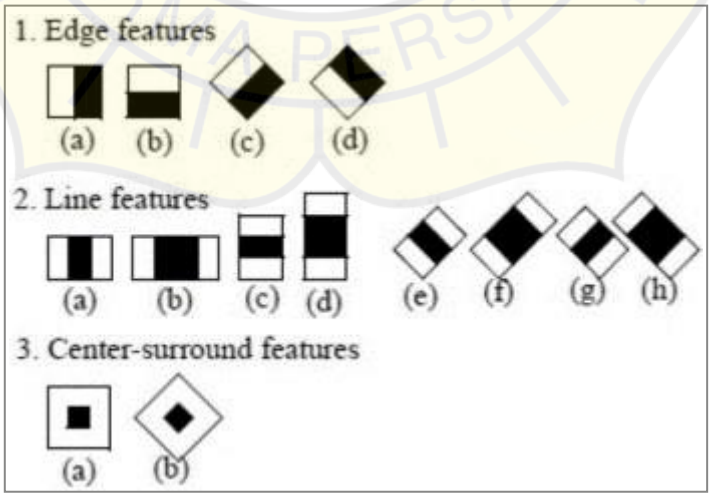
F Harr = Nilai fitur total
∑ F White = Nilai fitur pada daerah terang
∑ F Black = Nilai fitur pada daerah gelap

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

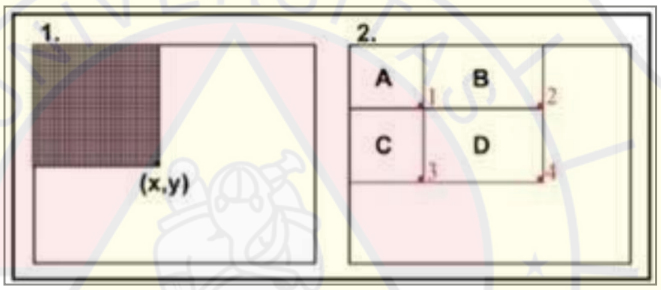


Gambar 2.3 *Variasi Fitur pada Haar (LienHart et al, 2002)*

2.5.2 Integral Image

Integral Image merupakan suatu teknik menghitung nilai fitur secara cepat dengan mengubah nilai dari setiap piksel menjadi suatu representasi citra baru.

Menurut Aries Suharso, "Pengenalan wajah menggunakan metode viola-jones dan eigenface dengan variasi posisi wajah berbasis webcam" (2016), *Integral Image* adalah sebuah citra yang nilai tiap pikselnya merupakan jumlah dari nilai *pixel* kiri atas hingga kanan bawah yang memungkinkan penghitungan pixel secara mudah dan berdasarkan jumlah seluruh *pixel* yang terkandung dalam batasan jendela fitur *haar* untuk distribusi fungsi kumulatif, sebagai mana disajikan dalam gambar 2.4



Gambar 2.4 *Integral Image (Evta, 2019)*

Berdasarkan Gambar 2.4, citra integral pada titik (x,y) (ii(x,y)) dapat dicari menggunakan persamaan (2.2) berikut :

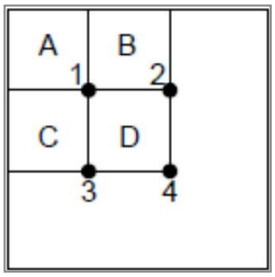
$$ii(x,y) = \sum x' \leq x,\ y' \leq y\ i(x',y') \qquad (2.2)$$

Keterangan :

II(x,y) = Citra integral pada lokasi x,y

I(x',y') = Nilai *pixel* pada citra asli

Perhitungan nilai dari suatu fitur dapat dilakukan dengan menghitung nilai citra integral pada empat buat titik seperti yang disajikan dalam Gambar 2.5

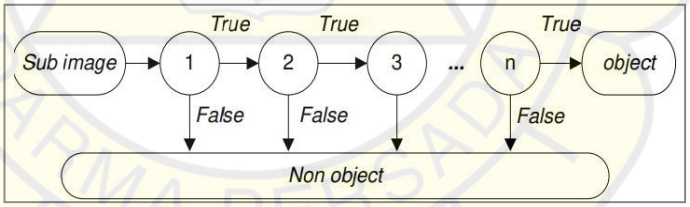


Gambar 2.5 *Perhitungan nilai fitur (Viola, 2004)*

Jika nilai *integral image* titik 1 adalah A, titik 2 adalah A+B, titik 3 adalah A+C dan titik 4 adalah A+B+C+D, maka jumlah *pixel* di daerah D dapat diketahui dengan cara 4+1-(2+3).

2.5.3 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 2.6 *Cascade Clasifier(Aries Suharso,2016)*

Dapat dijelaskan pada Gambar 2.6 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.

2.5.4 Adaptive Boosting

Menurut Dwisnanto Putro, Teguh Bharata Adji, Teguh Bharata Adji "Sistem deteksi wajah dengan menggunakan

metode viola-jones" (2012), *Adaptive Boosting* menggabungkan banyak *classifier* lemah untuk membuat sebuah *classifier* kuat. Dengan menggabungkan beberapa *Adaboost classifier* sebagai rangkaian filter yang cukup efisien untuk mengelompokan daerah gambar. *Classifier* lemah merupakan jawaban benar dengan tingkat keberatan yang kurang akurat. sebuah *classifier* lemah dapat dinyatakan sebagai berikut :

$$h_j(x) = \begin{cases} 1 & \text{if } p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases}$$

Keterangan

$h_j(x)$ adalah klasifikasi lemah

$p_j(x)$ adalah parity yang menunjukkan tanda pertidaksamaan ke j

θ_j adalah nilai ambang batas ke j

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

Javascript merupakan kode-kode program kecil yang digunakan membuat halaman *web* menjadi terlihat lebih dinamis dan menarik serta dapat juga membatasi aksi pengguna dalam web.

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"

2.7.1 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.*

2.8 WEBSITE

Website dalah sebuah kumpulan halaman pada suatu domain di internet yang dibuat dengan tujuan tertentu dan saling berhubungan serta dapat diakses secara luas melalui halaman depan (home page) menggunakan sebuah browser menggunakan URL untuk memudahkan tukar menukar dan menampilkan berbagai macam informasi teks, gambar, animasi, suara dan lain-lain.

Menurut Wendy Andriyan, Sarwan Septiawan, Annisa Aulya "Perancangan *website* sebagai media informasi dan peningkatan citra pada SMK dewi sartika Tangerang"(2020:79), *website* atau situs dapat diartikan sebagai kumpulan halaman yang menampilkan informasi, data, teks, gambar animasi, suara, video maupun gabungan dari semuanya yang bersifat statis maupun dinamis yang membentuk struktur yang saling terkait dan masing-masing dihubungkan dengan jaringan halaman.

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. Contoh sederhana dalam penggunaan PHP adalah sebagai berikut:

```
<?php
        Echo'Hello World':
>?
```

Dari contoh penulisan PHP di atas maka akan menghasilkan output “Hello World”. Perintah echo adalah perintah untuk melakukan cetak pada PHP.

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.

2.9.1 Struktur Leksikal

Menurut Kevin Tantroe, Peter Macintyre dan Rasmus Lerdorf dalam Programming PHP (2013, 1), Struktur leksikal bahasa pemrograman adalah seperangkat aturan dasar yang mengatur cara penulisan bahasa pemograman. Ini adalah sintaks level terendah dari bahasa dan menentukan hal-hal seperti nama variabel terlihat, karakter yang digunakan untuk sebuah komentar, dan penulisan perintah pemograman yang dipisahkan dengan perintah pemograman yang lainnya. Contoh struktur leksikal pemrograman php :

a) Case Sensitivity

Case Sensitivity adalah perbedaan variabel yang ditulis dengan huruf kecil dan besar dalam PHP. Nama-nama kelas dan fungsi yang ditentukan pengguna serta konstruksi bawaan dan kata kunci seperti echo, while, class dan lain lain merupakan *case-sensitive.*

Berikut contoh barus perintah yang memiliki variabel:

```
<?php
$x = 9;
$y = 10;
echo $X * $y;
?>
```

Maka hasilnya adalah :

***Notice**: undefined variable* X

Dalam PHP tidak membedakan adanya *Case Sensitive* walau pun variabel seperti $x dan $X itu dianggap sama dan tidak ada bedanya, namun karena menggunakan karakter yang berbeda pada echo saat penghasilan menggunakan variabel, maka variabel yang disebut harus sama dengan variabel yang dibuat, jika tidak hasil yang keluar adalah *undefined variable*.

2.9.2 Variabel

Variabel merupakan kode program yang menyatakan sebuah tempat penampung nilai-nilai tertentu yang dapat diubah-ubah. Nilai dalam variabel selanjutnya di pindahkan ke dalam *database* untuk di tampilkan atau di olah kembali. Penulisan variabel dalam php diawali dengan karakter “$” (dolar). Suatu variabel dapat memiliki nilai dari tipe apapun pada saat mengganti nilai variabel dengan nilai lain yang berbeda, sebagai contoh :

```
<?php
Fungsion perkalian () {
$nilai1 = 2;
$nilai2 = 5;
$hasil = $nilai1 * nilai2;
Echo “Hasil perkalian dari $nilai1 dan $nilai2 = $hasil”;
}
Perkalian ();
>?
```

Maka hasilnya adalah

Hasil perkalian dari 2 dan 5 = 10

2.9.3 Tipe Data

PHP menyediakan tipe data yang dapat dikelompoka menjadi beberapa bagian, yaitu:

Tabel 2.1 Tipe Data di PHP

Tipe	Contoh	Penjelasan
Integer	12345	Tipe data yang berisi bilangan bulat
Floating – Point	1.2345	Tipe data yang berisi nilai pecahan
Karakter	A,I,U,E	Tipe data berisi nilai karakter tunggal
String	“Meja”	Tipe data yang berisi teks atau Karekter
Array		Tipe data yang berisi nilai yang Sama
Objek		Tipe data yang berisi bilangan, variabel atau fungsi

2.10 Basis Data

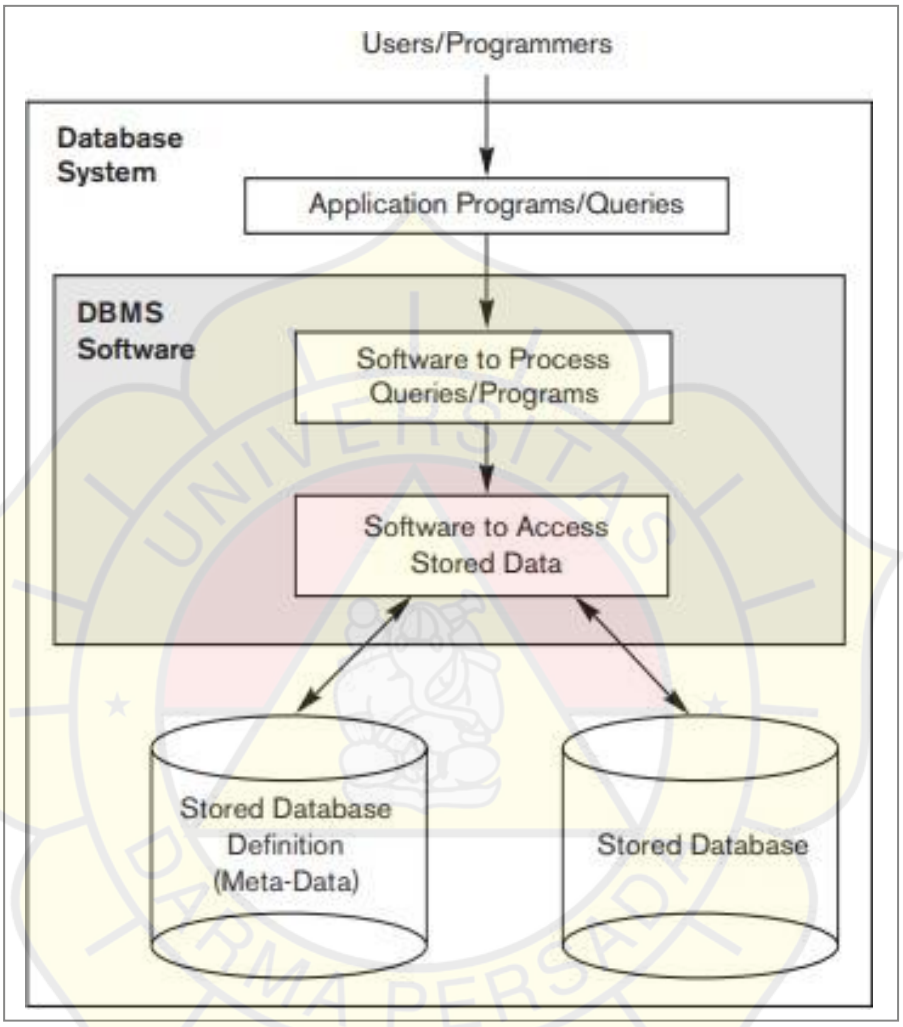
Menurut Ramez Elmasri, Shamkant B. *"Fundamentals of Database System 6th edition"* (2011, 4) Basis data merupakan kumpulan data terkait dengan data dan fakta yang diketahui akan direkam dan memiliki makna yang implisit.

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.

2.10.1 Database Management System

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 bersamaandalam DBMS.



Gambar 2.7 *Lingkup Basis Data ((Elmasri & B.Navathe, 2011)*

2.10.1.1 MySQL

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.

2.11 Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan salah satu standar bahasa yang banyak digunakan di dunia pemrograman untuk mendefinisikan *requirement*, membuat analisis dan desain serta menggambarkan arsitektur dalam pemrograman berorientasi objek. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuha sistem yang menggunakan diagram dan teks-teks pendukung.

Unified Modeling Language (UML) biasa digunakan untuk :

1. Menggambarkan batasan sistem dan fungsi sistem secara umum.
2. Menggambarkan proses atau kegiatan yang dilaksanakan secara umum.
3. Menggambarkan representasi struktur statis sebuah sistem dalam bentuk *class* diagram.
4. Menyatakan arsitektur implementasi fisik menggunakan *component and development*

Menurut Ade Hendini,"Pemodelan uml sistem informasi monitoring penjualan dan stok barang" (2016:108), *Unified Modeling Language* (UML) merupakan bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML adalah metodologi yang mengembangkan sistem berorientasi objek dan alat untuk mendukung pengembangan sistem.

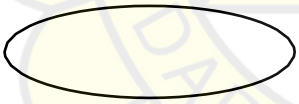
Adapun alat bantu yang digunakan dalam perencangan sistem berorientasi objek berbasis UML adalah sebagai berikut:

2.11.1 Use Case Diagram

Use case mendeskripsikan sebuah interaksi antara 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 yang akan dibuat.

Menurut Fitria Nur Hasanah dan Rahmania Sri untari dalam buku yang berjudul ”Rekayasa Perangkat Lunak” (2020:71), *Use case diagram* menjelaskan tentang apa yang akan diperbuat oleh sistem, dan mereperentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan teknik perekaman dan penggambaran fungsionalitas dari sebuah sistem seperti melakukan login ke sistem, mengbuat sebuah daftar belanja dan sebagainya. Simbol-simbol yang digunakan dalam *Use case diagram* yaitu:

Tabel 2.2 Notasi *Use Case Diagram(ade,2016)*

Simbol	Keterangan
Use Case 	Menggambarkan fungsionalitas dan interaksi antara sistem dengan aktor yang dinyatakan dengan menggunakan kata kerja
Actor 	Menggambarkan peran orang dan sistem yang mengaktifkan fungsi dari tugas-tugas yang berkaitan dengan peran pada konteks target sistem yang berinteraksi dengan *use case*.
Assosiasi	Penghubung interaksi aktor dengan *use case*

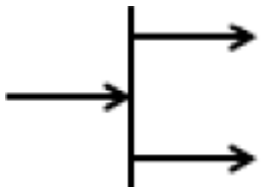
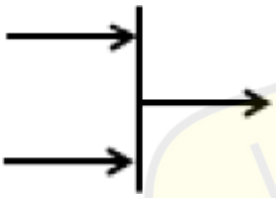
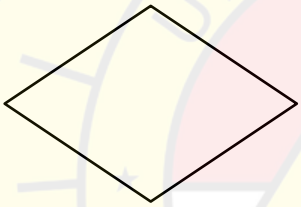
Include <<Include>>	Untuk menunjukan *use case* lain yang merupakan sebuah fungsi program
Extend <<Extend>>	Merupakan penambahan fungsional dari *use case* lain jika kondisi atau syarat terpehuni

2.11.2 Activity Diagram

Menurut Fitria Nur Hasanah dan Rahmania Sri untari dalam buku yang berjudul ”Rekayasa Perangkat Lunak” (2020:79), *Acivity diagram* menggambarkan *workflow* (alur kerja) atau alur aktivitas sebuah sistem yang sedang dirancang, dari bagaimana alur berawal, kondisi yang mungkin terjadi sampai alur tersebut berakhir. Simbol-simbol yang digunakan dalam *activity diagram* yaitu:

Tabel 2.3 Notasi *Activity Diagram(ade,2016)*

Simbol	Keterangan
Start Point 	Merupakan diagram aktivitas awal
End Point 	Merupakan akhir aktivitas sistem
Activity 	Menggambarkan suatu proses/kegiatan aktivitas sistem

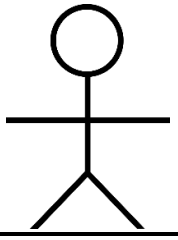
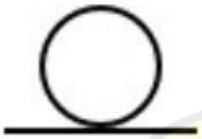
Fork 	Digunakan untuk menunjukan dekomposisi atau memecah satu kegiatan menjadi dua kegiatan paralel
Join 	Menunjukan kegiatan yang dilakukan seacara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
Decision 	Menggambarkan percabangan aktivitas lebih dari satu untuk pengambilan keputusan

2.11.3 Sequence Diagram

Menurut Ade Hendini,"Pemodelan uml sistem informasi monitoring penjualan dan stok barang" (2016:110), *Sequence Diagram* menggambarkan kelakuakn objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirim dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram* yaitu:

Tabel 2.4 Notasi *Sequence diagram(ade,2016)*

Simbol	Keterangan

Actor 	Menggambarkan sistem yang mengaktifkan fungsi dari tugas-tugas yang berkaitan dengan peran pada konteks target sistem yang berinteraksi dengan *use case* tapi tidak memiliki kontrol terhadap *use case*
Entity Class 	Kumpulan kelas berupa entitas yang membentuk gambaran awal sistem dan menjadi landasan menyusun basis data
Boundary Class 	Kumpulan kelas yang menggambarkan interaksi satu atau lebih dengan sistem
Control Class 	Suatu objek yang berisi logika aplikasi untuk menghubungkan antara boundary dengan tabel
Activation 	Operasi dari objek yang menggambarkan tempat mulai dan berakhirnya operasi

Lifetime	Garis yang mengubungkan objek sepanjang *lifetime* yang terdapat *activation*