

## BAB II

### LANDASAN TEORI

#### 2.1 Tinjauan Terhadap Penelitian Sebelumnya

##### 2.1.1 Pengenalan Warna Berbasis Android dengan Menggunakan Metode

##### *Template Matching*

Ringkasan:

Pada penelitian ini proses pengenalan warna menggunakan teknologi yang sama pada penelitian sebelumnya tetapi metode yang digunakan berbeda yaitu metode yang digunakan template matching metode ini juga memiliki algoritma yang sama serta digunakan untuk mendeteksi bentuk sebuah objek meskipun hanya digunakan untuk itu pada tugas akhir ini metode tersebut dicoba untuk mendeteksi warna dari sebuah objek, metode template matching merupakan metode yang sederhana dan banyak digunakan untuk mengenali pola. Metode ini bekerja dengan cara mengevaluasi pola citra yang akan dibandingkan dengan pola citra template pada basis data. Kelemahan metode ini adalah terbatasnya model yang akan dijadikan template sebagai pembanding pada basis data seperti bentuk, ukuran, dan orientasi. Oleh karena itu peneliti membangun media untuk pengenalan warna berbasis android dengan metode template matching. (Kurniawan, Indra dan Dewi, Nurcahyani (2018). Pengenalan Warna Berbasis Android dengan Menggunakan Metode *Template Matching*. Volume 7, Nomor 1, Mei 2018)

### 2.1.2 Deteksi Kesegaran dan Kualitas Telur Berdasarkan *Color Matching* dan *Template Matching*.

Ringkasan:

Kualitas dan kesegaran telur dapat dilihat dari ketinggian telur tersebut sedangkan warna kuning telur dengan menggunakan alat yang disebut *Yolk Color Fan* sebagai acuan dimana semakin tinggi nilai yolk maka semakin tinggi nilai kandungan karotein. Namun, untuk mengukur ketinggian telur maupun mencocokkan warna kuning telur dengan menggunakan alat ukur dan *Yolk Color Fan* secara kasat mata, hasil yang didapat akan bersifat subjektif karena disebabkan beberapa faktor seperti cahaya dan kemampuan penglihatan seseorang. Dalam tugas akhir ini penulis akan membahas bagaimana cara mendeteksi kualitas dan kesegaran dari tinggi dan warna kuning telur ayam negeri. Pada tugas akhir ini penulis menggunakan metode *Color Matching* dan *Template Matching*. Tujuan adanya tugas akhir ini, penelitian ini dapat mempermudah mendeteksi kualitas dan kesegaran kuning telur dengan hasil yang lebih efektif dan akurat. Sistem yang dibuat untuk kualitas kuning telur dengan metode *Color Matching* Mendapatkan Akurasi sebesar 87% sedangkan untuk kualitas dan kesegaran telur dengan metode *Template Matching* sebesar 76%. (Utami, Devi dan Dr. Ir. Hidayat, Bambang, DEA (2016). Deteksi Kesegaran dan Kualitas Telur Berdasarkan *Color Matching* dan *Template Matching*)

### 2.1.3 Identifikasi Uang Kertas Berdasarkan Warna Dengan Metode Template Matching

Ringkasan:

Dengan berkembangnya teknologi, media penyampaian informasi pun semakin beragam. Salah satunya adalah yang berkaitan dengan pengenalan objek. Aplikasi object recognition yang telah ada salah satunya adalah proses identifikasi citra pada uang kertas berdasarkan RGB. Dalam proyek akhir ini akan merancang bangun untuk mengidentifikasi mata uang kertas berdasarkan RGB. Dalam pembagian warna umumnya harus dilakukan inialisasi jumlah yang diinginkan terlebih dahulu. Untuk itu dalam proyek akhir ini menggunakan aplikasi java untuk menyelesaikan masalah tersebut. (Yehuda, Yohanes (2013). Identifikasi Uang Kertas Berdasarkan Warna Dengan Template Matching)

## 2.2 Pengertian Buah Jeruk.

Jeruk manis mempunyai rasa yang manis, kandungan air yang banyak dan memiliki kandungan vitamin C yang tinggi (berkisar 27-49 mg/100 gram daging buah). Vitamin C bermanfaat sebagai antioksidan dalam tubuh, yang dapat mencegah kerusakan sel akibat aktivitas molekul radikal bebas (Kusuma retno dkk, 2013). Sari buah jeruk manis mengandung 40-70 mg vitamin C per 100 ml, tergantung jenis jeruknya. Makin tua buah jeruk, umumnya kandungan vitamin C semakin berkurang, tetapi rasanya semakin manis (Anonim, 2007).

Dinegara berkembang, kulit dari jeruk sudah diolah menjadi parfum karena memiliki bau yang khas. Selain itu, kandungan gizi dalam jeruk ini juga dapat digunakan untuk terapi kesehatan seperti pertahanan tubuh, anti kanker, memerangi infeksi virus, dan menurunkan kadar kolesterol dalam tubuh (Ball, 1997). (Syani Juari (2012). Analisis Keputusan Dan Kepuasan Konsumen Jeruk Medan di Pasar Baru Bogor)

### **2.2.1 Jeruk Medan**

Jeruk Medan merupakan jeruk yang berasal dari Sumatera Utara, dengan nama ilmiah *Citrus sinensis* (L). Jeruk Medan termasuk varietas siam. Ciri-ciri Jeruk Medan adalah berukuran sedang, tangkainya kuat, bentuknya yang lebih pipih (gepeng), berwarna kuning cerah, harum, dan kulitnya tebal. Keunggulan Jeruk Medan adalah kulit buahnya yang tebal sehingga dapat disimpan dalam jangka waktu yang lebih lama sehingga berpeluang untuk diekspor ke mancanegara. Keunggulan Jeruk Medan yaitu kulit buahnya yang lebih tebal sehingga dapat disimpan dalam jangka waktu yang lebih lama dan berpeluang untuk diekspor ke mancanegara. (Syani Juari (2013). Analisis Keputusan Dan Kepuasan Konsumen Jeruk Medan di Pasar Baru Bogor).

### 2.2.2 Kelayakan Jeruk

Umur buah/tingkat kematangan buah yang dipanen, kondisi saat panen, dan cara panen merupakan faktor terpenting yang mempengaruhi mutu jeruk. Umur buah yang optimum untuk dipanen adalah sekitar 8 bulan dari saat bunga mekar. Ciri-ciri buah yang siap dipanen : jika dipijit tidak terlalu keras; bagian bawah buah jika dipijit terasa lunak dan jika dijentik dengan jari tidak berbunyi nyaring, warnanya menarik (muncul warna kuning untuk jeruk siam), dan kadar gula (PTT) minimal 10%. Kadar gula dapat ditentukan dengan alat hand refraktometer di kebun.

Dalam satu pohon, buah jeruk tidak semuanya dapat dipanen sekaligus, tergantung pada kematangannya. Jeruk termasuk buah yang kandungan patinya rendah sehingga bila dipanen masih muda tidak akan menjadi masak seperti mangga. Jika panen dilakukan setelah melampaui tingkat kematangan optimum atau buah dibiarkan terlalu lama pada pohon, sari buah akan berkurang dan akan banyak energi yang dikuras dari pohon sehingga mengganggu kesehatan tanaman dan produksi musim berikutnya. Panen yang tepat adalah pada saat buah telah masak dan belum memasuki fase akhir pemasakan buah. Dalam penyimpanan, rasa asam akan berkurang karena terjadi penguraian persenyawaan asam lebih cepat dari pada peruraian gula. (Ir. Sutopo, Msi (2011). Panen dan Pascapanen Jeruk)

### 2.3 Pengolahan Citra

Pengolahan citra adalah satu cabang dari ilmu Informatika. Pengolahan citra berkuat pada usaha untuk melakukan transformasi suatu citra atau gambar menjadi citra lain dengan menggunakan teknik tertentu.

Sebuah citra digital terdiri dari sejumlah elemen yang berhingga, di mana masing-masing mempunyai lokasi dan nilai tertentu. Elemen-elemen ini disebut sebagai picture element, image element, pels, atau pixels. Ada tiga level pengolahan citra, yaitu:

1. *Low-level process*, meliputi operasi dasar seperti image pre-processing.
2. *Mid-level process*, meliputi segmentasi (membagi sebuah gambar dalam region atau object), mendeskripsikan objek tersebut untuk direduksi dalam bentuk yang diinginkan dan klasifikasi dari objek tersebut. Input dari proses ini berupa gambar, dan outputnya berupa atribut yang diambil dari gambar (misal: *edge*, *countour*, dan identitas dari objek tertentu).
3. *High-level process*, meliputi pemberian arti dari suatu rangkaian objek-objek yang dikenali dan akhirnya menampilkan fungsi-fungsi kognitif secara normal sehubungan dengan penglihatan.

Tabel 2.1 Format Citra

Ekstensi	Nama	Keterangan
Jpg/jpeg	<i>Joint Photographic Experts Group</i>	JPEG biasanya digunakan untuk foto atau citra di website. JPEG menggunakan kompresi tipe lossy. Kualitas JPEG 2000 bisa bervariasi tergantung setting kompresi yang digunakan. Kompresi JPEG berbasis CT(Discrete Cosine Transform)
jp2/jpg2/j2k	<i>Joint Photographic Experts Group 2000</i>	Merupakan pengembangan dari JPEG yang berbasis transformasi wavelet. Format ini mendukung kompresi tipe lossless dan lossy. Namun, 9 support JPEG 2000 dalam berbagai aplikasi

		masih kurang, disebabkan kebutuhan hardware yang tangguh dan paten.
Pbm	Portable Bitmap Format	Merupakan format citra hitam putih yang sederhana. PBM memerlukan 1 bit tiap pixel. Tidak seperti format citra lainnya, format PBM merupakan plain text yang bisa diolah dengan menggunakan pengolah text. Format PBM merupakan bagian dari PNM (Portable Pixmap File Format).
Pgm	Portable Graymap Format	Merupakan format citra abu-abu yang sederhana. Format PGM memerlukan 8 bit tiap pixel. PGM merupakan citra mentah dengan

		<p>kompresi tipe 10 lossless. Format PGM merupakan bagian dari PNM (Portable Pixmap File Format).</p>
Ppm	Portable Pixmap Format	<p>Merupakan format citra berwarna yang sederhana. PPM merupakan citra mentah dengan kompresi tipe lossless. Format PPM merupakan bagian dari PNM (Portable Pixmap File Format)</p>
Png	Tagged Image File Format	<p>Merupakan format citra yang sudah digunakan sejak dulu. Mendukung kompresi tipe lossless dan lossy.</p>

(Budiharto, Widodo. 2016, Machine Learning & Computational Intelligence; CV.

ANDI OFFSET).

### 2.3.1 Tujuan Pengolahan Citra Digital

- Memperbaiki kualitas gambar dilihat dari aspek radiometrik (peningkatan kontras, transformasi warna, restorasi citra) dan dari aspek geometrik (rotasi, tranlasi, skala, transformasi geometrik).
- Melakukan proses penarikan informasi atau deskripsi objek atau pengenalan objek yang terkandung pada citra.
- Meakukan kompresi ataureduksi data untuk tujuan penyimpanan data, transmisi data, dan waktu proses data.

(Budiharto, Widodo. 2016, Machine Learning & Computational Intelligence; CV. ANDI OFFSET).

### 2.3.2 *Supervised Learning*

*Supervised Learning* yaitu metode belajar dengan adanya latihan dan pelatih. Banyak teknik dalam pattern recognition masuk dalam kategori ini. Sebagai contoh regresi, analisis diskriminan (LDA), *Artificial Neural Network*, (*ANN*), dan *Support Vector Machine (SVM)*, dan masih banyak lagi. Dalam pendekatan ini, untuk menemukan fungsi keputusan, fungsi pemisah atau fungsi regresi, kita menggunakan beberapa contoh data yang mempunyai output atau label selama proses training. (Budiharto, Widodo. 2016, Machine Learning & Computational Intelligence; CV. ANDI OFFSET).

### 2.3.2 Ruang Warna RGB

Salah satu model warna citra yang banyak dipergunakan dalam penelitian pengolahan citra digital adalah model warna RGB. Sutoyo (Sutoyo, 2009) menjelaskan setiap piksel pada citra warna mewakili warna dasar yang merupakan kombinasi dari tiga warna dasar merah (Red), hijau (Green) dan biru (Blue), dimana setiap warna mempunyai gradasi sebanyak 256 warna dengan variasi intensitas cahaya antara 0 sampai 255. Model ini disebut model warna RGB. Variasi dari gabungan ketiga intensitas cahaya inilah yang akan menghasilkan variasi warna-warna yang berbeda-beda. Model warna RGB dihasilkan dari tiga kombinasi warna utama (Red/merah, Green/hijau, dan Blue/biru) yang diturunkan menjadi nama model, dan pada spectrum cahaya yang dikombinasikan akan menghasilkan sebuah warna.

Warna RGB merupakan prinsip warna yang digunakan oleh media elektronik seperti televisi, monitor komputer, dan juga scanner. Oleh karena itu, warna yang ditampilkan RGB selalu terang dan menyenangkan, karena memang di setting untuk display monitor, bukan untuk cetak, sehingga lebih leluasa dalam bermain warna. Tapi bukan berarti RGB bebas masalah karena tampilan warna RGB akan selalu terikat dengan kapasitas/kemampuan grafis komputer yang menyandangnya. Jadi apabila komputer yang kita pakai mempunyai graphic card yang bagus serta monitor LCD, maka tampilan warna RGBnya akan jauh lebih bagus dibanding monitor tabung dengan graphic card yang biasa-biasa saja. (Budiharto, Widodo. 2016, Machine Learning & Computational Intelligence; CV. ANDI OFFSET).

### 2.3.3 Grayscale

Grayscale adalah suatu istilah untuk menyebutkan satu citra yang memiliki warna abu-abu, hitam, dan putih. Grayscale adalah koleksi atau kisaran corak monokromik (abu-abu), mulai dari putih murni di ujung yang paling terang hingga hitam murni di ujung yang berlawanan.

Grayscale hanya berisi informasi pencahayaan (kecerahan) dan tidak ada informasi warna; itulah sebabnya pencahayaan maksimum putih dan pencahayaan hitam nol; segala sesuatu di antaranya adalah warna abu-abu. Itulah sebabnya gambar grayscale hanya berisi nuansa abu-abu dan tidak ada warna. Grayscale juga dikenal sebagai akromatik.

Gambar digital biasanya berisi informasi warna dan pencahayaan atau grayscale. Jika Anda menghapus informasi warna, Anda dibiarkan dengan grayscale, menghasilkan gambar hitam putih. Grayscale adalah aspek penting dari gambar, dan itu adalah satu-satunya bagian yang tidak dihapus; jika tidak, gambar hitam murni akan menghasilkan informasi warna apa pun yang ada.

Gambar digital terdiri dari kelompok tiga piksel dengan warna merah, hijau dan biru (RGB), juga disebut saluran dalam pencitraan digital. Setiap saluran juga mengandung nilai luminansi untuk menentukan seberapa terang atau gelap warnanya. Untuk mendapatkan gambar grayscale, informasi warna dari setiap saluran dihapus, hanya menyisakan nilai pencahayaan, dan itulah sebabnya gambar menjadi pola area gelap dan terang tanpa warna, pada dasarnya gambar hitam dan putih.

Sebagian besar aplikasi perangkat lunak pencitraan digital, bahkan yang paling dasar, dapat mengubah gambar menjadi grayscale. Ini juga sangat penting saat mencetak, karena hanya menghabiskan tinta hitam, tidak seperti pencetakan berwarna, yang menghabiskan ketiga warna cetak (cyan, magenta, dan kuning) serta hitam. (Budiharto, Widodo. 2016, Machine Learning & Computational Intelligence; CV. ANDI OFFSET).

#### **2.4 Computer Vision**

Pada masa ini, perhatian difokuskan pada kemampuan komputer dalam mengerjakan sesuatu yang dapat dilakukan oleh manusia. Dalam hal ini, komputer tersebut dapat meniru kemampuan kecerdasan dan perilaku manusia. Salah satu cabang dari kecerdasan buatan ialah Computer Vision. Menurut Forsyth dan Ponce, Computer Vision adalah:

*Extracting description of the world from picture or sequences of pictures.*

(Budiharto, Widodo. 2016, Machine Learning & Computational Intelligence; CV. ANDI OFFSET).

#### 2.1.4 Template Matching

Template Matching adalah salah satu teknik dalam deteksi objek pada citra digital yang berfungsi untuk mencocokkan tiap-tiap bagian dari suatu citra. Algoritma ini berkerja dengan cara mengevaluasi pola citra yang akan dibandingkan dengan pola citra template pada basis data. Pendekatan dalam pendeteksian objek ini menggunakan pendekatan berbasis template.

Dari sejumlah penelitian prinsip Metode Template Matching memiliki karakteristik antara lain :

1. Relatif mudah untuk diaplikasikan dalam teknik pengolahan citra digital.
2. Hasilnya relative akurat karena mendeteksi kesalahan hingga ukuran piksel.
3. Walaupun demikian, metode ini cukup rentan terhadap orientasi antara citra acuan (template) dengan citra yang akan diidentifikasi, yang meliputi : ukuran, posisi dan kualitas citra.
4. Untuk mendapatkan hasil yang maksimal maka metode ini sangat tergantung pada teknik pengolahan citra digital yang lain seperti enhacement, filtering, dan masih banyak lagi.

(Budiharto, Widodo. 2016, Machine Learning & Computational Intelligence; CV. ANDI OFFSET).

### 2.1.5 ORB

Fitur ini diusung atas detector keypoint yang dikenal dengan FAST, dan dikembangkan menggunakan descriptor BRIEF. FAST adalah pemilihan algoritme untuk menemukan keypoints dalam keadaan real-time yang cocok dengan fitur visual. Parameter yang diambil oleh FAST adalah intensitas permulaan antara tengah pixel dan yang berada dalam lingkaran.

## 2.5 Pemrograman Android

### 2.5.1 Android

Android adalah system operasi berbasis linux yang dimodifikasi untuk perangkat bergerak (*mobile devices*) yang terdiri dari system operasi, *middleware*, dan aplikasi utama. Awalnya, Android dikembangkan oleh Android Inc. Perusahaan ini kemudian dibeli oleh Google pada tahun 2005. System aplikasi Android kemudian diluncurkan bersamaan dengan dibentuknya organisai Open Source pada tahun 2007. Selain Google, beberapa naa besar juga ikut dalam *Open Handset Alliance*, antara kain Motorola, Samsung, LG, Sony Ercson, T-Mobile, Vodafone, Toshiba, dan Intel. (Juhara, P. Zamrony. 2016, Panduan Lengkap Android, CV ANDI OFFSET).

### 2.5.2 Model Aplikasi Android

Android adalah OS berbasis Linux *multiuser*. Tiap aplikasi yang diinstall adalah user terppisah dan akan menerima user ID unik. Android juga mengatur file permission semua file pada aplikasi sehingga hanya aplikasi tersebut yang dapat mengaksesnya. Aplikasi Android ditulis menggunakan bahasa pemrograman Java dan akan dijalankan disebuah VM (bernama Dalvik VM) seperti aplikasi

Java lainnya. Aplikasi pada platform Android terdiri dari satu atau lebih *Activity* di mana tiap *Activity* adalah sebuah entry point. Tiap *Activity* terdiri dari satu *Thread* atau lebih.

Diantara banyak *Activity*, sebuah *Activity* harus bertindak sebagai *Activity* utama (main launcher) yang akan dieksekusi oleh aplikasi ketika pengguna memuat aplikasi. Namun, aplikasi tidak dapat mengansumsikan bahwa akan selalu dimulai dari *Activity* utama. Hal ini bias disebabkan karena *Activity* lain yang ada pada aplikasi berbeda dapat saja meminta sembarang *Activity* pada aplikasi lain untuk dijalankan. (Juhara, P. Zamrony. 2016, Panduan Lengkap Android, CV ANDI OFFSET)

### **2.5.3 Arsitektur *Android***

Secara umum, arsitektur *Android* terdiri dari lapisan software, yaitu lapisan framework, Android run time sebagai lapisan perantara (middleware), serta kernel Linux yang membungkus hardware di bawahnya. Berikut penjelasan arsitektur *Android*:

#### **a. Aplikasi**

Aplikasi yang anda buat berada pada level tertinggi dalam hierarki struktur software di *Android* seperti telepon, pembaca email, program SMS, kontak, browser, dan kalender. Aplikasi umumnya ditulis dalam bahasa pemograman Java.

b. Framework Aplikasi

Dibawah aplikasi terdapat sejumlah software pendukung, meliputi layanan pengelola Activity, View, sumber daya, pemberitahuan (notifikasi), dan lain-lain.

c. Pustaka

Android menyediakan sejumlah pustaka grafis 2D dan 3D, multimedia playback, browser engine, pencetakan font, database, dan lain-lain. Aplikasi yang memanfaatkan pustaka fungsi ini melalui lapisan framework aplikasi.

d. Android *Runtime*

Android menyediakan pustaka inti bagi pemograman Java. Dibawahnya terdapat Dalvik Virtual Machine (Dalvik VM) yang akan menjalankan aplikasi. Tiap aplikasi akan dijalankan pada proses terpisah dengan VM yang berbeda sehingga terisolasi satu sama lain. Aplikasi masih dapat berkomunikasi dengan aplikasi lain melalui mekanisme yang disediakan framework aplikasi. Dalvik VM bergantung pada lapisan dibawahnya (kernel Linux) untuk multi-threading dan pengelolaan memori tingkat rendah.

e. Kernel Linux

Kernel Linux bertanggung jawab menyediakan layanan dasar seperti keamanan, pengelolaan proses, pengelolaan file, pengelolaan sumber daya memori, dan hardware.

(Juhara, P. Zamrony. 2016, Panduan Lengkap Android, CV ANDI OFFSET)

#### 2.5.4 Komponen Aplikasi *Android*

Aplikasi *Android* ditulis dengan bahasa pemrograman Java. Semua file disatukan dalam satu paket berupa file berekstensi APK, sebuah file yang dapat didistribusikan. Tiap file APK adalah sebuah aplikasi tunggal. Komponen *Android* terdiri dari beberapa jenis, antara lain:

a) *Activity*

*Activity* adalah suatu istilah yang digunakan dalam pemrograman *Android* untuk mengacu pada satuan interaksi dan pengguna melalui antarmuka grafis (graphical user interface), GUI. Sebagai satuan interaksi, *Activity* adalah tampilan yang Anda lihat di layar seperti Windows atau kotak dialog pada pemrograman aplikasi Desktop.

b) *Intent*

*Intent* adalah istilah yang digunakan dalam pemrograman *Android* untuk mengacu pada mekanisme berbagai pesan pemberitahuan atau bertukar data *Activity* atau untuk menjalankan aplikasi lain.

c) *Service*

*Service* adalah komponen aplikasi yang berjalan di belakang layar user interface untuk menyelesaikan akan layanan tertentu seperti mengecek RSS feed secara kontinu atau memainkan music. *Service* tetap berjalan meski *Activity* yang mengendalikannya telah berhenti. Media player adalah sebuah contoh aplikasi yang menggunakan *Service*.

d) *Content Provider*

*Content provider* membuat suatu aplikasi dapat berbagi sejumlah data tertentu kepada aplikasi lain. Jika membutuhkan data nama-nama kontak, aplikasi Anda tinggal meminta data tersebut.

e) *Broadcast Receiver*

Komponen yang memantau, menerima, dan bereaksi terhadap pesan yang diebarkan baik oleh system maupun aplikasi lain. Misalnya, ketika baterai lemah, Android akan mengirim pesan “baterai lemah” kepada semua *broadcast receiver* yang ingin diberitahu pesan ini.

(Juhara, P. Zamrony. 2016, Panduan Lengkap Android, CV ANDI OFFSET)

## **2.6 Aplikasi Web**

### **2.6.1 World Wide Web**

*World Wide Web* (WWW) adalah suatu program yang ditemukan oleh Tim Berners-Lee pada tahun 1991. Awalnya Berners-Lee hanya ingin menemukan cara untuk menyusun arsip-arsip risetnya. Untuk itu, beliau mengembangkan suatu sistem untuk keperluan pribadi. Sistem itu adalah program peranti lunak yang diberi nama Enquire. Dengan program itu, Berners-Lee berhasil menciptakan jaringan yang menautkan berbagai arsip sehingga memudahkan pencarian yang dibutuhkan.

Bagaimana World Wide Web bekerja? World Wide Web bekerja pada tiga mekanisme berikut:

- a. Informasi disimpan di dalam dokumen yang sering kita sebut halaman *web*.
- b. Halaman *web* adalah *file-file* yang disimpan dalam *computer*. Komputer tersebut dikenal dengan istilah *web server*.
- c. Komputer yang mengakses isi dari halaman *web* disebut dengan *web clients*.
- d. *Web clients* menampilkan halaman *web* dengan program yang dikenal dengan nama *Web Browser* seperti Chrome, Firefox, Internet Explorer.

(Priyanto Hidayatullah, Jauhari Khairul Kawistara. 2017. Pemrograman Web)

### 2.6.2 Internet

*Internet* adalah jaringan global yang menghubungkan komputer-komputer di seluruh dunia. Dengan *Internet*, sebuah komputer dapat mengakses data yang terdapat pada komputer lain di benua yang berbeda.

Pada tahun 1982 Istilah "*Internet*" pertama kali digunakan, dan TCP/IP diadopsi sebagai protokol universal untuk jaringan tersebut. Pada tahun 1986 diperkenalkan sistem nama *domain*, yang sekarang dikenal dengan nama DNS (*Domain Name System*) yang berfungsi untuk menyeragamkan sistem perbaikan nama alamat di jaringan komputer. (Priyanto Hidayatullah, Jauhari Khairul Kawistara. 2017. Pemrograman Web)

### 2.6.3 Browser

Berikut adalah contoh *browser* populer:

a. Mozilla Firefox

Mozilla Firefox adalah penjelajah *web* antar-platform gratis yang dikembangkan oleh Yayasan Mozilla dan ratusan sukarelawan.

b. Google Chrome

Google Chrome adalah sebuah penjelajah *web* sumber terbuka yang dikembangkan oleh Google.

c. Internet Explorer

Internet Explorer disingkat IE atau MSIE, adalah sebuah browser web proprietier yang gratis dari Microsoft. (Bekti, Humaira Bintu, 2015, Mahir membuat Website dengan Dreamweaver CS6, CSS, dan JQUERY. Yogyakarta: CV. ANDI OFFSET.)

## 2.7 Pemograman Web

### 2.7.1 HTTP

*HyperText Transfer Protocol* adalah protokol agar *client* dan *server* bisa berkomunikasi dengan gaya *request-response*. HTTP menentukan bagaimana cara pengirimannya, serta bagaimana *web server* dan *browser* beraksi dan bereaksi terhadap berbagai perintah.

Cara Kerja HTTP:

1. Pertama-tama, komputer klien (HTTP klien) membuat sambungan, lalu mengirimkan permintaan dokumen ke web server.
2. HTTP server kemudian memproses permintaan klien, sementara itu, HTTP klien menunggu respon dari server tersebut.
3. Web server merespon permintaan dengan kode status data, lalu barulah menutup sambungan ketika telah selesai memproses permintaan.

Pada intinya, dalam kasus HTTP, client terlebih dahulu melakukan permintaan data kepada server, lalu kemudian server mengirimkan respon berupa file HTML yang ditampilkan dalam browser, ataupun data lainnya yang diminta oleh klien.

Sementara itu, HTTPS bekerja dengan menggunakan sublayer berupa SSL atau TLS (Transport Layer Security), tentu berbeda dengan HTTP yang memiliki aplikasi layer biasa. Oleh karena itulah HTTPS sering disebut juga sebagai kombinasi dari HTTP dan SSL/TLS.

(Priyanto Hidayatullah, Jauhari Khairul Kawistara. 2017. Pemrograman Web)

### **2.7.2 HTML**

HTML (*Hyper Text Markup Language*) adalah *format* standar yang digunakan untuk membuat halaman *web*. HTML merupakan subset dari bahasa yang lebih luas, yaitu SGML (*Standard Generalized Markup Language*). Versi terakhir saat ini adalah HTML5. (Mukhamad Masrur, 2016. Pemograman Web Dinamis menggunakan Java Server Pages dengan Database Relasional MySQL. Yogyakarta: CV. ANDI OFFSET).

Penanda perintah di dalam HTML di sebut TAG. TAG digunakan untuk menentukan tampilan dari dokumen HTML. Tag ada yang berpasangan dengan format seperti ini. Tag yang berpasangan dengan format seperti berikut:

<BEGIN TAG> Isi dokumen </END TAG>

Tag ada juga yang tidak berpasangan dengan *format* dengan *format* berikut ini:

<TAG> Isi dokumen

Namun perlu diperhatikan bahwa tag <p> lebih baik diakhiri dengan tag </p> karena untuk kedecoannya, HTML mengharuskan penggunaan end tag.

Yang bisa dilakukan dengan HTML, yaitu:

- a. Mengatur tampilan dari halaman web dan isinya. ★
- b. Membuat tabel dalam halaman web.
- c. Mempublikasikan halaman web secara online.
- d. Membuat form yang bisa digunakan untuk menangani registrasi dan transaksi via web.
- e. Menambahkan objek-objek seperti audio, video, animasi, java applet dalam halaman web.
- f. Menampilkan area gambar (*canvas*) di browser.

(Priyanto Hidayatullah, Jauhari Khairul Kawistara. 2017. Pemograman Web. Yogyakarta: CV. ANDI OFFSET)

### 2.7.3 PHP

*Hypertext Preprocessor* adalah bahasa script yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk membuat program situs *web dinamis*. PHP juga sering digunakan untuk membangun sebuah CMS.

PHP adalah bahasa pemrograman *script server-side* yang didesain untuk pengembangan *web*. Disebut *server side* dikarenakan PHP diproses pada komputer server. Hal ini berbeda jika dibandingkan dengan bahasa pemrograman *client-side* seperti JavaScript yang diproses pada *web browser* (client).

PHP memiliki beberapa kelebihan, yaitu:

- a. Dapat membuat *Web* menjadi *Dinamis*.
- b. PHP bersifat *Open Source* yang berarti dapat digunakan oleh siapa saja secara gratis.
- c. Program yang dibuat dengan PHP bisa dijalankan oleh semua Sistem Operasi (OS) karena PHP berjalan secara *Web Base* yang artinya semua Sistem Operasi (OS) karena PHP berjalan secara *Web Base* yang artinya semua Sistem Operasi termasuk *Handphone* yang mempunyai *Web Browser* dapat menggunakan program PHP.
- d. Aplikasi PHP lebih cepat dibandingkan dengan ASP maupun Java.
- e. Mendukung banyak paket Database seperti MySQL, Oracle, PostgreSQL, dan lain-lain.
- f. Bahasa pemrograman PHP tidak memerlukan kompilasi pada penggunaannya.

- g. Banyak *Web Server* yang mendukung PHP seperti Apache, Lighttpd, Iis, dan lain-lain,
- h. Pengembangan aplikasi PHP mudah karena banyak Dokumentasi, *Referensi*, dan *Developer* yang membantu dalam pengembangannya.
- i. Banyak berebaran Aplikasi dan Program PHP yang gratis dan siap pakai seperti Wordpress, PrestaShop, dan lain-lain. (Priyanto Hidayatullah, Jauhari Khairul Kawistara. 2017. Pemograman Web. Yogyakarta: CV. ANDI OFFSET)

#### 2.7.4 Bootstrap

*Bootstrap* merupakan salahsatu *framework* HTML, CSS, dan JS yang cukup populer, serta banyak digunakan oleh para pengembang *web*. *Framework* ini banyak digunakan untuk membuat *website* yang bersifat *responsif*. Artinya bisa menyesuaikan tampilan layoutnya berdasarkan ukuran *viewport* dari *device* pengaksesnya, mulai dari *smarthpone*, tablet, atau layar PC.

*Bootstrap* telah menyediakan beberapa fitur yang berguna untuk mendesain sebuah halaman *web* yang responsif, yaitu:

- a. *Scaffolding*, yaitu bagian yang mengatur *background*, *link styles*, *grid system*, serta *layout*. Fitur ini akan membantu para *developer* untuk mambuat *layout* yang rapih.
- b. *Base CSS*, yaitu bagian yang mengatur tampilan elemen HTML seperti *typography*, *code*, *table*, *form*, *button*, serta *glyphicons* (kumpulan ikon-ikon kecil).
- c. *Components*, yaitu bagian yang mengatur tampilan komponen, seperti *tab*, *navbar*, *alerts*, *page header*.

d. *JavaScript Plugins*, yaitu yang menyediakan beberapa komponen interaktif seperti *tooltips*, *popovers*, *modals*, dan lain sebagainya.

(Utomo, Priyo Eko, 2016. Bikin Sendiri Toko Online Dinamis dengan Bootstrap dan PHP. Yogyakarta: MediaKom)

Fitur-fitur yang digunakan pada *Bootstrap* dapat membuat sang *developer Bootstrap* kaya jika dia berorientasi dengan uang. *Framework* ini seharusnya layak untuk menjadi *framework* berbayar, mengingat fitur yang ditawarkan sangat lengkap. Kenyataannya, *framework* ini justru tersedia secara gratis, yaitu dengan lisensi MIT. MIT adalah lisenensi *open source*, di mana semua kalangan bebas menggunakan *framework* ini secara gratis tanpa takut terkena masalah legalitas. (WAHANA KOMPUTER dan PENERBIT ANDI. 2016. Responsive Web Design with Bootstrap. Yogyakarta: CV. ANDI OFFSET.)

### 2.7.5 CSS

*Cascading Style Sheet* (CSS) adalah lembar kode yang digunakan untuk mengatur penampilan elemen-elemen *HyperText Markup Language* (HTML). Hal ini dibuat dengan tujuan agar kode HTML hanya memusatkan perhatian pada konten atau isi halaman *web*, sedangkan urusan penampilan/data informasi menjadi urusan CSS. Walaupun CSS sering kali diletakkan di dokumen HTML, kode CSS bukanlah bagian dari dokumen HTML. CSS merupakan bahasa tersendiri yang dirancang untuk memperkaya tampilan informasi di halaman web.

Secara umum, CSS memiliki propert-properti yang dapat dipakai untuk mengatur tampilan elemen-elemen HTML. Gambaran tentang fungsi properti CSS dapat dilihat dari tabel berikut

Tabel 2.2 Fungsi CSS

Properti	Keterangan
Warna	Hal ini memungkinkan warna latar belakang elemen yang diatur dan begitu juga warna teks di elemen.
Bingkai	Hal ini memungkinkan elemen seperti div atau canvas diberi bingkai dan dimensinya pun bisa ditentukan.
Font	Berbagai pengaturan font dapat dilakukan, antara lain terkait dengan jenis font, ukuran font, dan berbagai dekorasi untuk tulisan.
Tabel	Tabel dan seluruh elemen pendukungnya bisa diformat dengan mudah. Sebagai contoh, sangat mudah untuk membuat tabel yang baris bernomor genap dan ganjil dibedakan dalam hal warna latar belakangnya.
Transformasi	Elemen mudah sekali untuk ditransformasikan, misalnya diputar dan diskalakan.
Antarmuka	Sebagai contoh, sangat mudah untuk mengganti bentuk kursor ketika penunjuk mouse menyorot ke suatu elemen tertentu.
Posisi	Posisi elemen dapat diformat dengan berbagai cara.
List	Jika Anda melibatkan list (tag<ul>), properti-properti CSS memungkinkan mengatur sifat-sifat di list.

Keuntungan menggunakan CSS:

- a. Memisahkan presentation sebuah dokumen dari content document itu sendiri.
- b. Mempermudah dan mempersingkat pembuatan dari pemeliharaan dokumen *web*.
- c. Mempercepat proses rendering/pembacaan HTML.
- d. Ukuran file lebih kecil.
- e. Dapat berkolaborasi dengan JavaScript.
- f. Pasangan serta XHTML.
- g. Menghemat pekerjaan tentunya, dimana hanya membuat satu halaman CSS.
- h. Mudah mengganti-ganti tampilan dengan hanya mengubah file CSS-nya.

Aturan penulisan CSS (Deklarasi):

Selector (property : value;)

Keterangan:

- Selector: tag HTML yang akan didefinisikan (misal body, table, td dan lain-lain).
- Property: atribut yang akan diubah.
- Value: nilai dari atribut.

Contoh:

```
body {font-family : sans-serif; font-size : 18px;}
```

Contoh penulisan secara kelompok: (menggunakan tanda koma “,” atau tanda dan “&”)

```
h1, h2 {color : pink;};
```

```
table, tr & td {color : white;};
```

Terdapat tiga cara penempatan kode CSS, yaitu:

a. Internal CSS

Yaitu menuliskan kode langsung *script* CSS di file HTML-nya. Penulisan CSS dengan mendefinisikan dulu didalam sebuah tag `<style type="text/CSS">` yang ditulis sebelum tag `</head>`.

b. External CSS

Yaitu memanggil file CSS dari tempat/folder/lokasi lain, dengan kata lain file CSS terpisah dengan file HTML. *Browser* akan membaca definisi style dari file CSS, dan akan mengikuti format tersebut. *External style* ditulis di *text editor* lain dengan ekstensi.CSS. Contoh:

```
<link type="text/CSS" href=http://externalCSS/coba.css rel="stylesheet" media = "all"></link>
```

c. Inline CSS

Yaitu penulisan kode CSS dalam tag HTML. Contoh:

```
<div align="center" style"background:pink;">
```

Ini adalah bagian yang diatur style nya

```
</div> (Kadir, Abdul, 2015. Belajar Sendiri Pasti Bisa CSS3. Yogyakarta: CV.
```

```
ANDI OFFSET.)
```

## 2.8 Database

*Database* adalah sekumpulan *file* data yang satu sama lainnya saling berhubungan yang diorganisasi sedemikian rupa sehingga memudahkan untuk mendapatkan dan memproses data tersebut. Data adalah kumpulan fakta dasar (mentah) yang terpisah.

Sebuah *database* harus dibuat dengan rapi agar setiap data yang dimasukkan sesuai dengan tempatnya. Sebagai contoh, di sebuah Perpustakaan penyimpanan buku dilakukan berdasarkan atas jenis atau kategori-kategori tertentu dari buku-buku tersebut, misalnya kategori buku komputer, buku pertanian, dan lain-lain. Kemudian dikelompokkan lagi berdasarkan abjad dari judul-judul buku. Ini dilakukan agar setiap pengunjung dapat dengan mudah mencari dan mendapatkan buku yang dimaksud.

Untuk itu, sebuah *database* yang baik harus memiliki kategori-kategori yang digunakan sebagai pengelompokan data-data. Penggunaan *database* pada komputer dilakukan dengan menggunakan tabel-tabel. Pada tabel-tabel tersebut masih dikelompokkan lagi menjadi beberapa bagian untuk membedakan data yang satu dengan data yang lain. Pada sebuah tabel *database* harus memiliki satu kategori data yang digunakan sebagai kunci untuk membedakan data-data yang ada di dalam suatu tabel. Data kunci tersebut tidak boleh sama antara satu dengan data yang lain. Data kunci disebut dengan **Primary Key**. (Mukhamad Masrur, 2016. Pemrograman Web Dinamis menggunakan Java Server Pages dengan Database Relasional MySQL. Yogyakarta: CV. ANDI OFFSET).

### 2.8.1 DBMS

*Database Management System* adalah aplikasi yang dipakai untuk mengelola basis data. DBMS biasanya menawarkan beberapa kemampuan yang terintegrasi seperti:

- a. Membuat, menghapus, menambah, dan memodifikasi basis data.
- b. Tidak semua orang bisa mengakses basis data yang ada sehingga memberikan keamanan bagi data.
- c. Kemampuan berkomunikasi dengan program aplikasi yang lain.  
Misalnya dimungkinkan untuk mengakses basis data MySQL menggunakan aplikasi yang dibuat menggunakan PHP.
- d. Kemampuan pengaksesan melalui komunikasi antarkomputer (*client server*). (Hidayatullah, Priyanto dan Jauhari Khairul Kawistara. 2014. *Pemograman Web*. Bandung: Informatika Bandung).

### 2.8.2 Web Server

*Web server* adalah suatu program komputer yang mempunyai tanggung jawab atau tugas menerima permintaan HTTP dari komputer klien, yang dikenal dengan nama *web browser* dan melayani mereka dengan menyediakan respon HTTP berupa konten data, biasanya berupa halaman *web* yang terdiri dari dokumen HTML dan objek seperti gambar dan lain-lain.

Untuk menjalankan PHP yang disertai *database* MySQL dapat digunakan dua jenis *web server* yaitu Online mode dan Offline mode. Pada *Online mode*, selain komputer harus mempersiapkan *domain* dan *hosting* serta koneksi internet yang memadai untuk mengelolanya sehingga harus keluar biaya ekstra terlebih

dahulu. Sedangkan *pada Offline mode*, yang anda persiapkan cukup komputer dan beberapa software untuk membuat *web server* lokal. Pada cara kedua ini komputer PC akan dibuat menjadi *web server* lokal atau localhost sebelum benar-benar meng-uploadnya ke *web server* Internet.

Ada beberapa jenis *software* untuk membangun *web server* lokal atau localhost yang support akan sistem operasi Windows diantaranya adalah WampServer, Appserv, XAMPP, PHP Triad, atau Vertigo. (Kadir, Abdul. 2008, Tuntunan Praktis: Belajar Database Menggunakan MySQL. Yogyakarta: CV. ANDI OFFSET)

### 2.8.3 MySQL

MySQL adalah salah satu aplikasi DBMS yang sudah sangat banyak digunakan oleh para pemograman aplikasi *web*. Contoh DBMS lainnya adalah: PostgreSQL (*freeware*). SQL Server, MS Access dari Microsoft, DB2 dari IBM, Oracle, dan Oracle Corp, Dbase, FoxPro, dsb.

Kelebihan MySQL adalah gratis, handal selalu di-*update* dan banyak forum yang memfasilitasi para pengguna jika memiliki kendala. MySQL hyga menjadi DBMS yang sering dibundling dengan *web server* sehingga proses instalasinya jadi lebih mudah. (Priyanto Hidayatullah, Jauhari Khairul Kawistara. 2017. Pemograman Web. Yogyakarta: CV. ANDI OFFSET).

## 2.9 Metodologi Pengembangan Sistem

Dalam perancangan sistem pembayaran SPP dan tabungan siswa berbasis *web* pada SMK Jayakarta ini penulis menggunakan metodologi *waterfall*. Metodologi *waterfall* adalah suatu proses pengembangan perangkat lunak berurutan, di mana kemajuan dipandang sebagai terus mengalir ke bawah (seperti air terjun). (Meinarini Catur Utami , Yoga Tri Hutomo. 2014. Penerapan Waterfall Dalam Analisis Dan Perancangan Sistem Informasi Manajemen Dokumen Surat Menyurat Pada Bank BJB Kantor Cabang BSD Tangerang).

### 2.9.1 Requirement (Analisis Kebutuhan)

Dalam langkah ini merupakan analisa terhadap kebutuhan sistem. Pengumpulan data dalam tahap ini bisa melakukan sebuah penelitian, wawancara atau *study literature*. Seseorang sistem analisis akan menggali informasi sebanyak-banyaknya dari user sehingga akan tercipta sebuah sistem komputer yang bisa melakukan tugas-tugas yang diinginkan oleh user tersebut. Tahapan ini akan menghasilkan dokumen user requirement atau bisa dikatakan sebagai data yang berhubungan dengan keinginan user dalam pembuatan sistem. Dokumen inilah yang akan menjadi acuan sistem analisis untuk menterjemahkan kedalam bahasa pemrograman.

#### 2.9.1.1 Wawancara

Wawancara adalah situasi berhadap-hadapan antara pewawancara dan responden yang dimaksudkan untuk menggali informasi yang diharapkan, dan bertujuan mendapatkan data tentang responden dengan minimum bias dan maksimum efisiensi (Singh, 2002). Sementara Steward & Cash (1982)

mendefinisikan wawancara sebagai sebuah proses komunikasi dyad (interpersonal), dengan tujuan yang telah ditentukan sebelumnya, bersifat serius, yang dirancang agar tercipta interaksi yang melibatkan aktivitas bertanya dan menjawab pertanyaan. 168 | Aspirasi Vol. 4No. 2, Desember 2013 Singh (2002) menuliskan bahwa terdapat dua macam wawancara yaitu wawancara formal dan informal. Wawancara formal atau disebut juga wawancara terstruktur adalah sebuah prosedur sistematis untuk menggali informasi mengenai responden dengan kondisi dimana satu set pertanyaan ditanyakan dengan urutan yang telah disiapkan oleh pewawancara dan jawabannya direkam dalam bentuk yang terstandardisasi. Wawancara informal adalah sebuah wawancara dimana tidak dipersiapkan terlebih dahulu pertanyaan-pertanyaan, tidak ada persiapan urutan pertanyaan, dan pewawancara yang berkuasa penuh untuk menentukan pertanyaan sesuai dengan poin-poin utama (Singh, 2002). Dikarenakan hampir segala sesuatunya tergantung pewawancara maka proses wawancara menjadi tidak terstruktur, dan karenanya wawancara semacam ini disebut juga wawancara tidak terstruktur.

### **2.9.2 Design (Desain)**

Proses *design* akan menterjemahkan syarat kebutuhan sebuah perancangan perangkat lunak yang dapat diperkirakan sebelum dibuat koding. Proses ini berfokus pada struktur data, arsitektur perangkat lunak, representasi *interface*, dan *detail* (algoritma) prosedural. Tahapan ini akan menghasilkan dokumen yang disebut *software requirement*. Dokumen inilah yang akan digunakan *programmer* untuk melakukan aktivitas pembuatan sistemnya.

### 2.9.2.1 Unified Modelling Language (UML)

*Unified Modelling Language (UML)* adalah keluarga notasi grafis yang didukung oleh model-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO). Definisi ini merupakan definisi yang sederhana.

UML 2 terdiri dari 13 jenis diagram resmi, yaitu:

Tabel 2.4 Diagram UML

Diagram	Kegunaan	Turunan
Activity	Behavior prosedural dan paralel	Di UML 1
Class	Class, fitur dan hubungan-hubungan	Di UML 1
Communication	Interaksi antar objek; penekanan pada jalur	Diagram kolaborasi UML 1
Component	Struktur dan koneksi komponen	Di UML 1
Composite Structure	Dekomposisi runtime sebuah class	Di UML 2
Deployment	Pemindahan artifak ke node	Di UML 1
Interaction Overview	Campuran sequence dan	Baru di UML 2

	activity diagram	
Object	Contoh konfigurasi dari contoh-contoh	Tidak resmi di UML1
Package	Struktur hirarki compile-time	Tidak resmi di UML1
Sequence	Interaksi antar objek; penekanan pada sequence	Di UML 1
State Machine	Bagaimana even mengubah objek selama aktif	Di UML 1
Timing	Interaksi antar objek; penekanan pada timing	Baru di UML 2
Use Case	Bagaimana pengguna berinteraksi dengan sebuah sistem	DI UML 1

(Fowler, Martin, 2014. UML Distilled Edisi 3 Panduan Singkat Bahasa Permodelan Objek Standar).

### 2.9.2.1.1 Use Case Diagram

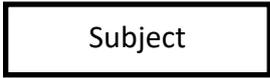
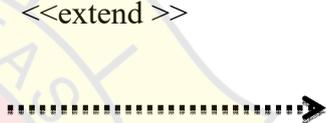
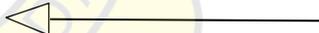
*Use Case* adalah teknik untuk merekam persyaratan fungsional sebuah sistem. *Use case* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan memberi sebuah narasi tentang bagaimana sistem tersebut digunakan.

Tidak ada cara standar penulisan isi sebuah *use case* dan format yang berbeda dapat bekerja dengan baik dalam kasus yang berbeda. Setiap *use case* memiliki aktor utama yang meminta sistem untuk memberi sebuah layanan. Aktor utama adalah aktor dengan tujuan yang akan dipenuhi *use case* dan biasanya, tetappi tidak selalu, merupakan inisiator *use case*. Selain itu terdapat banyak aktor lain yang berkomunikasi dengan sistem pada saat menjalankan *use case*. Mereka dikenal dengan aktor sekunder.

Setiap langkah dalam *use case* adalah sebuah elemen dalam interaksi antara aktor dan sistem. Setiap langkah harus berupa pernyataan sederhana dan dengan jelas menunjukan siapa yang menjalankan langkah tersebut, Langkah tersebut menunjukkan tujuan aktor, bukan mekanisme yang harus dilakukan aktor.

Tabel 2.4 Simbol *Use Case Diagram*

Nama elemen dan fungsi	Notasi
<p>Actor menggambarkan tokoh atau sistem yang memperoleh keuntungan dan berada di luar dari sistem. Actor dapat berasosiasi dengan actor lainnya dengan menggunakan <i>specialization</i> atau <i>superclass association</i>. Actor ditempatkan di luar <i>subject boundary</i>.</p>	 <p>Actor / Role</p>
<p><i>Use Case</i>, mewakili sebuah bagian dari fungsionalitas sistem dan ditempatkan dalam <i>system boundary</i>.</p>	 <p>Use Case</p>

<i>Subject Boundary</i> , menyatakan lingkup dari subjek.	
<i>Association Relationship</i> , menghubungkan actor untuk berinteraksi dengan <i>use case</i> .	
<i>Include Relationship</i> , menunjukkan <i>inclusion fungsionalitas</i> dari sebuah <i>use case</i> dengan <i>use case</i> lainnya. Arah panah dari <i>base use case</i> ke <i>included use case</i>	
<i>Extend Relationship</i> , menunjukkan <i>extension</i> dari sebuah <i>use case</i> untuk menambahkan <i>optional behavior</i> . Arah panah dari <i>extension use case</i> ke <i>base use case</i>	
<i>Generalization Relationship</i> Menunjukkan generalisasi dari <i>use case</i> khusus ke yang lebih umum	

(Fowler, Martin, 2014. UML Distilled Edisi 3 Panduan Singkat Bahasa Permodelan Objek Standar)

### 2. 9.2.1.2 Activity Diagram

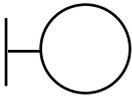
*Activity diagram* adalah teknik untuk menggambarkan logika prosedural, proses bisnis, dan jalur kerja. Dalam beberapa hal, diagram ini memainkan peran mirip sebuah diagram alir, tetapi perbedaan prinsip antara diagram ini dan notasi diagram alir adalah diagram ini mendukung *behavior paralel*.

*Activity diagram* memungkinkan siapapun yang melakukan proses untuk memilih urutan dalam melakukannya. Dengan kata lain, diagram hanya menyebutkan aturan-aturan rangkaian dasar yang kita ikuti. Hal ini penting untuk permodelan bisnis karena proses-proses sering muncul secara paralel. Inijuga berguna pada algoritma yang bersamaan, dimana urutan-urutan independen dapat melakukan hal-hal secara paralel.

### 2.9.2.1.3 Sequence Diagram

Sebuah *sequence diagram*, secara khusus, menjabarkan *behavior* sebuah skenario tunggal. Diagram tersebut menunjukkan sejumlah objek contoh dan pesan-pesan yang melewati objek-objek ini di dalam *use case*.

Tabel 2.5 Simbol *Sequence Diagram*

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk gambaran awal sistem dan menjadi landasan untuk menyusun basis data
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi interface atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan form entry dan form cetak
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas,

	contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , simbol mengirim pesan antar class.
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , <i>activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi.
	<i>Lifeline</i> , garis titik-titik yang terhubung dengan objek, sepanjang lifeline terdapat activation

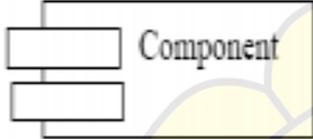
Setiap garis alir memiliki sebuah balok aktivasi yang menunjukkan saat partisipan aktif di dalam interaksi. Hal ini berhubungan dengan salah satu metode partisipan yang terdapat dalam susunan tersebut. Balok aktivasi merupakan pilihan dalam UML. Satu-satunya pengecualian adalah saat memeriksa sebuah desain dalam sesi desain, karena akan terlihat janggal untuk menggambar balok-balok di papan tulis.

(Fowler, Martin, 2014. UML Distilled Edisi 3 Panduan Singkat Bahasa Permodelan Objek Standar)

### 2. 9.2.1.4 Deployment Diagram

Deployment Diagram digunakan untuk menggambarkan detail bagaimana komponen disusun di infrastruktur sistem.

Tabel 2.6 Simbol *Deployment Diagram*

Gambar	Keterangan
	<p>Pada deployment diagram, komponen komponen yang ada diletakkan didalam node untuk memastikan keberadaan posisi mereka.</p>
	<p>Node menggambarkan bagian-bagian hardware dalam sebuah sistem. Notasi untuk node digambarkan sebagai sebuah kubus 3 dimensi.</p>
	<p>Sebuah association digambarkan sebagai sebuah garis yang menghubungkan dua node yang mengindikasikan jalur komunikasi antara elementelemen hardware.</p>

(Fowler, Martin, 2014. UML Distilled Edisi 3 Panduan Singkat Bahasa Permodelan Objek Standar).

### **2.9.3 *Implementation & Testing (Implementasi dan Uji Coba Program )***

Tahapan inilah yang merupakan tahapan secara nyata dalam mengerjakan suatu sistem. Dalam artian penggunaan computer akan dimaksimalkan dalam tahapan ini. Setelah pengkodean selesai maka akan dilakukan testing terhadap sistem yang telah dibuat tadi. Tujuan testing adalah menemukan kesalahan-kesalahan terhadap sistem tersebut dan kemudian bisa diperbaiki

### **2.9.4 *Integration & Testing (Penerapan / Pengujian Program)***

Tahapan ini bisa dikatakan final dalam pembuatan sebuah sistem. Setelah melakukan analisa, design dan pengkodean maka sistem yang sudah jadi akan digunakan oleh *user*.

### **2.9.5 *Operation & Maintenance (Pemeliharaan)***

Perangkat lunak yang sudah disampaikan kepada pelanggan pasti akan mengalami perubahan. Perubahan tersebut bisa karena mengalami kesalahan karena perangkat lunak harus menyesuaikan dengan lingkungan (peripheral atau sistem operasi baru) baru, atau karena pelanggan membutuhkan perkembangan fungsional.