

## BAB 2

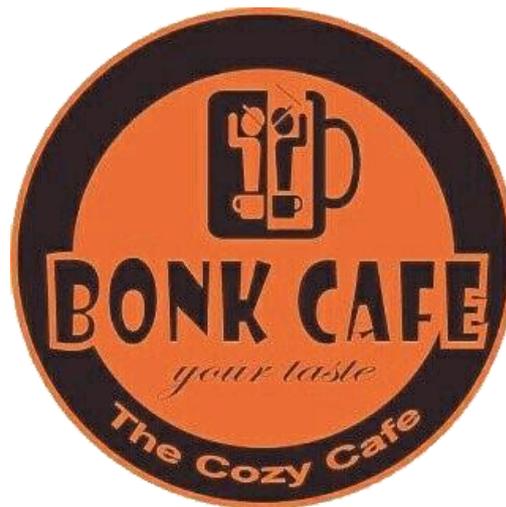
### TINJAUAN PUSTAKA

#### 2.1 Profil Bonk Cafe

Bonk *cafe* merupakan salah satu *cafe* yang baru berdiri pada bulan desember 2017 di daerah Kota Sidoarjo tepatnya di Jl. setia budi no. 28 Katerungan, Kria. Bonk *cafe* menyediakan aneka makanan bejenis *snack*, *steak* dan aneka minuman berbahan dasar kopi dan teh.

##### 2.1.1 Logo

Logo merupakan suatu identitas ataupun ciri bagi pembuka usaha, berikut ini logo yang dimiliki oleh Bonk *Cafe*.



**Gambar 2.1 Logo Bonk Cafe.**

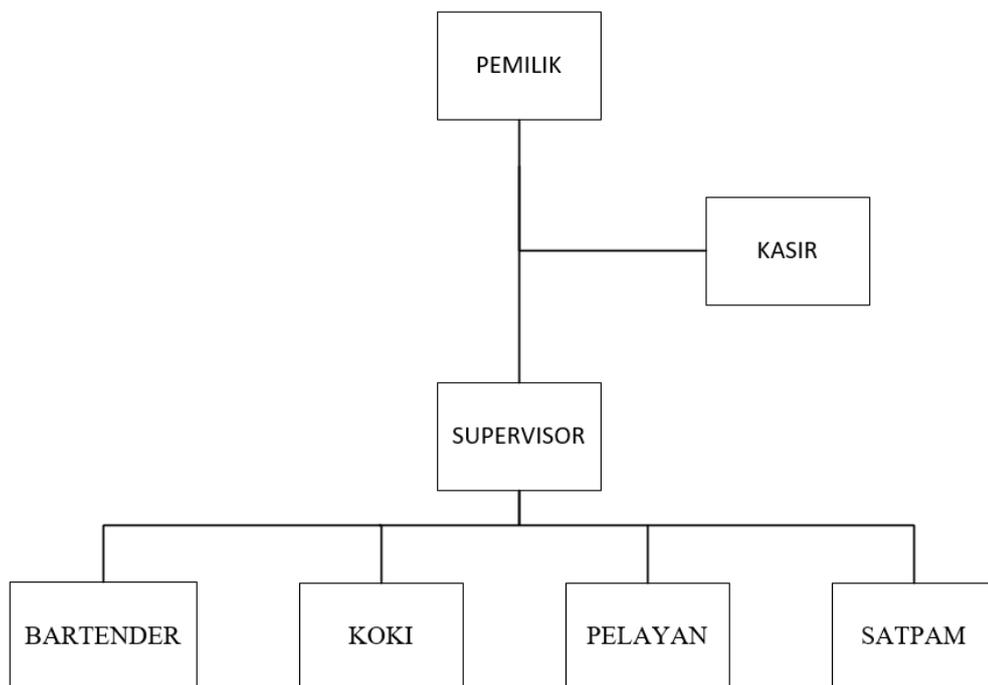
Adapun keterangan pada logo Bonk *Cafe* adalah sebagai berikut:

1. Lingkaran yang mengelilingi semua area dari logo dapat di ilustrasikan seperti cangkir kopi yang tertuang dilihat dari atas, melambangkan bonk *cafe* adalah *cafe* yang menyajikan kopi-kopi terbaik.
2. Gambar gelas dan 2 orang didalamnya dapat mengartikan bonk *cafe* adalah tempat yang asik bagi semua pengunjung untuk saling berkumpul sambil menikmati aneka makanan dan minuman yang ada di bonk *cafe*.

3. Tulisan *your taste* dan *The Cozy Cafe* menegaskan bahwa *bonk cafe* adalah *cafe* memiliki cita rasa yang disukai semua orang dan dapat dinikmati oleh semua pengunjung.

### 2.1.2 Struktur Organisasi

Tingkatan tanggung jawab dan kewajiban di *Bonk Cafe* dapat dilihat pada struktur organisasi berikut ini.



**Gambar 2.1** Struktur Organisasi *Bonk Cafe*.

Penjelasan fungsi tanggung jawab dan kewajiban dari masing-masing jabatan yang tertera di struktur organisasi diatas adalah sebagai berikut.

1. Pemilik

Bertanggung jawab atas seluruh kegiatan operasional *Cafe* dan memiliki kewajiban untuk menjaga arus keuangan *Cafe* agar terus dapat beroperasi.

2. Kasir

Bertanggung jawab atas penerimaan uang hasil penjualan di *cafe* dan pembayaran tagihan-tagihan operasional di *cafe*. Kasir berkewajiban memberikan laporan hasil penerimaan dan pengeluaran uang kepada pemilik. Personel Kasir di *Cafe Bonk* hanya 1 orang.

3. Supervisor

Bertanggung jawab atas kualitas makanan dan pelayanan di *cafe*. Kewajiban dari supervisor adalah melakukan pengarahan dan memperhatikan kinerja kepada staff koki, staff *bartender*, staff pelayan, satpam dan juru parkir. Supervisor di *Cafe Bonk* hanya 1 orang.

4. Koki

Bertanggung jawab atas kualitas makanan yang disajikan kepada pelanggan. Kewajiban koki adalah cek persediaan bahan baku makanan, memasak pesanan pelanggan. Total koki di *Cafe Bonk* ada 2 orang.

5. *Bartender*

Bertanggung jawab atas kualitas minuman yang disajikan kepada pelanggan. Kewajiban *bartender* adalah cek persediaan bahan baku minuman dan memproses minuman pelanggan. Total *bartender* di *Cafe Bonk* 1 orang.

6. Pelayan

Bertanggung jawab untuk mencatat dan mengantar pesanan pelanggan. Kewajiban dari pelayan adalah berpenampilan rapi dan memiliki tingkah laku yang baik didepan pelanggan. Total jumlah pelayan di *Cafe Bonk* 6 orang.

7. Satpam

Bertanggung jawab atas keamanan *cafe*. Kewajiban satpam adalah memberikan rasa nyaman kepada pelanggan dan melakukan penindakan atas tindakan kejahatan di lingkungan *cafe*. Jumlah Satpam di *Cafe Bonk* 1 orang.

### 2.1.3 Daftar Menu

Bonk *cafe* memiliki aneka jenis makanan dan minuman, berikut adalah daftar menu yang ada pada bonk *cafe*.

**Tabel 2.1 Daftar Menu.**

No.	Kategori	Nama Menu	Harga
1.	<i>Food</i>	Nasi Goreng	Rp. 10.000
2.	<i>Food</i>	Mie Goreng	Rp. 10.000
3.	<i>Food</i>	Paket Bento <i>Box</i>	Rp. 15.000
4.	<i>Snack</i>	Siomay	Rp. 7.000
5.	<i>Snack</i>	Tahu <i>Crispy</i>	Rp. 6.000
6.	<i>Snack</i>	<i>Cilok Steak</i>	Rp. 9.000
7.	<i>Snack</i>	<i>Fried Chicken</i>	Rp. 10.000
8.	<i>Snack</i>	<i>Chicken Katsu</i>	Rp. 10.000
9.	<i>Snack</i>	<i>Pizza Mini</i>	Rp. 9.000
10.	<i>Snack</i>	Roti Bakar	Rp. 7.000
11.	<i>Snack</i>	Kentang Goreng	Rp. 7.000
12.	<i>Snack</i>	<i>Cassadilas</i>	Rp. 9.000
13.	<i>Snack</i>	<i>Monay Bag</i>	Rp. 8.000
14.	<i>Ice cream</i>	<i>Hard Ice Cream</i>	Rp. 9.000
15.	<i>Ice cream</i>	<i>Ice Cream Cone</i>	Rp. 5.000
16.	<i>Ice cream</i>	<i>Rujak Ice Cream</i>	Rp. 9.000
17.	<i>Ice cream</i>	<i>Pancake Ice Cream</i>	Rp. 9.000
18.	<i>Drink</i>	<i>Tea (Ice)</i>	Rp. 3.000
19.	<i>Drink</i>	<i>Lime Tea (Hot)</i>	Rp. 5.000
20.	<i>Drink</i>	<i>Tea (Hot)</i>	Rp. 3.000
21.	<i>Drink</i>	<i>Lime Tea (Ice)</i>	Rp. 5.000
22.	<i>Drink</i>	<i>Jeruk (Hot)</i>	Rp. 4.000
23.	<i>Drink</i>	<i>Jeruk (Ice)</i>	Rp. 4.000
24.	<i>Drink</i>	<i>Chocolate (Hot)</i>	Rp. 6.000
25.	<i>Drink</i>	<i>Chocolate (Ice)</i>	Rp. 4.000

No.	Kategori	Nama Menu	Harga
26.	<i>Drink</i>	Susu Soda ( <i>Ice</i> )	Rp. 9.000
27.	<i>Drink</i>	Tea Tarik ( <i>Hot</i> )	Rp. 6.000
28.	<i>Drink</i>	Tea Tarik ( <i>Ice</i> )	Rp. 6.000
29.	<i>Drink</i>	Sop Buah ( <i>Ice</i> )	Rp. 9.000
30.	<i>Coffee</i>	Exspresso Vietnam Drip	Rp. 6.000
31.	<i>Coffee</i>	Exspresso Plunger	Rp. 8.000
32.	<i>Coffee</i>	Exspresso Bellety Exspress	Rp. 5.000
33.	<i>Coffee</i>	Americano	Rp. 3.000
34.	<i>Coffee</i>	Kopi Susu	Rp. 4.000
35.	<i>Coffee</i>	Cappuchino	Rp. 7.000
36.	<i>Coffee</i>	Cafe Mocca	Rp. 7.000
37.	<i>Coffee</i>	English Toffee	Rp. 7.000
38.	<i>Coffee</i>	Machiato	Rp. 7.000
39.	<i>Coffee</i>	Conpanna	Rp. 7.000
40.	<i>Milk shake</i>	Chocolachino	Rp. 6.000
41.	<i>Milk shake</i>	Frappe Volcadochino	Rp. 6.000
42.	<i>Milk shake</i>	Milo Frapp	Rp. 6.000
43.	<i>Milk shake</i>	Strobery Milkshake	Rp. 6.000
44.	<i>Milk shake</i>	Sirsak Milkshake	Rp. 6.000
45.	<i>Milk shake</i>	Grape Milkshake	Rp. 6.000
46.	<i>Milk shake</i>	Green Tea	Rp. 7.000
47.	<i>Milk shake</i>	Oreo Frappe	Rp. 7.000
48.	<i>Squash</i>	Blue Fire	Rp. 6.000
49.	<i>Squash</i>	Blue Ocean	Rp. 6.000
50.	<i>Squash</i>	Orange Squash	Rp. 6.000
51.	<i>Squash</i>	Lime Squash	Rp. 6.000
52.	<i>Squash</i>	Laccistroberry	Rp. 6.000
53.	<i>Squash</i>	Sun Flower	Rp. 6.000

## 2.2 Landasan Teori

Landasan teori merupakan penjelasan dari berbagai konsep dan teori-teori yang berkaitan dengan pembangunan aplikasi di Bonk *Cafe*.

### 2.2.1 Sistem

Secara umum sistem dapat didefinisikan sebagai sekumpulan objek-objek yang saling berinteraksi serta hubungan antar objek bisa dilihat sebagai satu kesatuan yang dirancang untuk mencapai satu tujuan [10]. Sistem mempunyai karakteristik atau sifat-sifat tertentu, yaitu mempunyai komponen-komponen (*components*), batas (*boundary*), lingkungan luar sistem (*environments*), penghubung (*interface*), masukan (*input*), keluaran (*output*), pengolah (*process*), dan sasaran (*objectives*) atau tujuan (*Goal*).

#### 1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerjasama membentuk suatu kesatuan. Komponen-komponen sistem atau elemen-elemen sistem dapat berupa subsistem atau bagian-bagian dari sistem. Setiap sistem tidak peduli betapapun kecilnya, selalu mengandung komponen-komponen atau subsistem-subsistem. Setiap subsistem mempunyai sifat-sifat dari subsistem untuk menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai suatu sistem yang lebih besar disebut dengan supra sistem, misalnya suatu perusahaan dapat disebut sebagai suatu sistem sedang industri yang merupakan sistem yang lebih besar dapat disebut dengan supra sistem. Kalau dipandang industri sebagai suatu sistem, maka perusahaan dapat disebut sebagai subsistem. Demikian juga bila perusahaan dipandang sebagai suatu sistem, maka sistem akuntansi adalah subsistemnya. Kalau sistem akuntansi dipandang sebagai suatu sistem, maka perusahaan adalah supra sistem dan industri adalah supra dari supra sistem.

#### 2. Batas Sistem

Batas sistem (*boundary*) merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lainnya atau dengan lingkungan luarnya. Batas sistem ini memungkinkan suatu sistem dipasang sebagai suatu

kesatuan. Batas suatu sistem menunjukkan ruang lingkup (scope) dari sistem tersebut.

### 3. Lingkungan Sistem

Lingkungan luar (*environment*) dari suatu sistem adalah apapun diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan dapat bersifat merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi dari sistem dan dengan demikian harus tetap dijaga dan dipelihara. Sedang lingkungan luar yang merugikan harus ditahan dan dikendalikan, kalau tidak maka akan mengganggu kelangsungan hidup dari sistem.

### 4. Penghubung Sistem

Penghubung (*interface*) merupakan media penghubung antara satu subsistem dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem ke subsistem yang lain. Keluaran (*output*) dari satu subsistem akan menjadi satu masukan (*input*) bagi subsistem yang lain dan akan melalui penghubung. Dengan penghubung satu subsistem dapat berintegrasi dengan subsistem yang lainnya membentuk satu kesatuan.

### 5. Masukan Sistem

Masukan (*input*) adalah energi yang dimasukkan kedalam sistem. Masukan dapat berupa masukan peralatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance input* adalah energi yang diproses agar didapatkan keluaran. Sebagai contoh didalam sistem komputer, program adalah *Maintenance input* yang digunakan untuk mengoperasikan komputernya sedangkan data adalah signal input untuk diolah menjadi informasi.

### 6. Keluaran Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan-bahan yang lain menjadi keluaran

berupa barang jadi. Sistem akuntansi akan mengolah transaksi menjadi laporan keuangan dan laporan-laporan lain yang dibutuhkan oleh manajemen.

#### 7. Sasaran Sistem

Suatu sistem pasti mempunyai tujuan (*Goal*) atau sasaran (objektif). Kalau sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan sekali, masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem.

### 2.2.2 Unified Modelling Language (UML)

*Unified Modeling Language* (UML) merupakan sistem arsitektur yang bekerja dalam OOAD (*Object-Oriented Analysis/Design*) dengan satu bahasa yang konsisten untuk menentukan, visualisasi, mengkonstruksi, dan mendokumentasikan artifact (sepotong informasi yang digunakan atau dihasilkan dalam suatu proses [11]). Tujuan UML diantaranya adalah :

1. Memberikan model yang siap pakai, pemodelan visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.
2. Memberikan contoh bahasa pemodelan yang bebas dari berbagai Bahasa pemrograman dan proses rekayasa.
3. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan.

UML terdiri dari beberapa diagram pemodelan berikut ini.

#### A. *Use Case* Diagram

*Use Case* Diagram merupakan model diagram UML (*Unified Modeling Language*) yang digunakan untuk menggambarkan kesepakatan fungsional yang diharapkan dari sebuah system [11]. Diagram ini menjelaskan manfaat suatu sistem jika dilihat menurut pandangan orang yang berada diluar sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar.

#### B. *Activity* Diagram

*Activity Diagram* memodelkan alur proses dan hubungan antar proses dalam suatu sistem informasi dan juga berisikan tentang scenario yang ada dalam sistem tersebut [11].

#### C. *Sequence Diagram*

*Sequence Diagram* adalah diagram yang menjelaskan interaksi antar objek yang disusun berdasarkan waktu proses berlangsung. Diagram ini digunakan untuk menggambarkan tahap demi tahap yang harus dilakukan oleh pengguna sistem untuk menghasilkan sesuatu dari *use case diagram* yang sudah dibuat [11].

#### D. *Class Diagram*

*Class Diagram* adalah diagram yang menunjukkan *Class-Class* yang ada dari sebuah sistem dan saling berhubungan secara logika [11]. Diagram ini menggambarkan struktur statis dari sebuah sistem.

### 2.2.3 **Android**

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup system operasi, middleware dan aplikasi [12]. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc. membeli Android Inc. yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/smartphone. Kemudian untuk mengembangkan android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Pada saat perilisannya Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan *open source* pada perangkat *mobile*. Di lain pihak, Google merilis kode-kode Android dibawah lisensi Apache, sebuah lisensi perangkat lunak dan *open platform* perangkat seluler. Di dunia ini terdapat dua jenis distributor system operasi Android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Services* (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD).

Sekitar September 2007 *Google* mengenalkan Nexus One, salah satu jenis smartphone yang menggunakan Android sebagai system operasinya. Telepon seluler ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 Januari 2010. Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM *Holdings*, *Atheros Communications*, diproduksi oleh asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan Open Handset Alliance, OHA mengumumkan produk perdana mereka, Android, perangkat *mobile* yang merupakan modifikasi kernel Linux 2.6. Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru.

Pada masa saat ini kebanyakan *vendor-vendor* smartphone sudah memproduksi smartphone berbasis Android, *vendor-vendor* itu antara lain HTC, Motorola, Samsung, LG, Acer, Philips, T-Mobile, Nexian, IMO, Asus dan masih banyak lagi *vendor* smartphone didunia yang memproduksi Android, hal ini karena Android itu adalah system operasi yang *Open Source* sehingga bebas didistribusikan dan dipakai oleh *vendor* manapun.

Android juga tidak hanya menjadi sistem operasi di smartphone, saat ini Android menjadi pesaing utama dari Apple pada system operasi Tablet PC. Pesatnya pertumbuhan Android selain faktor yang disebutkan diatas adalah karena Android itu sendiri adalah *platform* yang sangat lengkap baik itu system operasinya, aplikasi dan Tool Pengembangan, Market aplikasi Android serta dukungan yang sangat tinggi dari komunitas *Open Source* di dunia, sehingga Android terus berkembang pesat baik dari segi teknologi maupun dari segi jumlah device yang ada di dunia.

Android sebagai *platform mobile* pertama yang lengkap, terbuka, dan bebas karena beberapa faktor berikut ini [12]:

- a. Lengkap (*Complete Platform*): Pendekatan yang komprehensif sering digunakan para desainer ketika mereka sedang mengembangkan *platform* Android. Android merupakan sistem operasi yang aman dan banyak menyediakan tools dalam membangun software dan memungkinkan untuk peluang pengembangan aplikasi.

- b. Terbuka (*Open Source Platform*): *platform* Android disediakan melalui lisensi *Open Source*. Penhembang dapat dengan bebas untuk mengembangkan aplikasi. Android sendiri menggunakan Linux Kernel 2.6.
- c. *Free* (*Free Platform*): Android adalah *platform* / aplikasi yang bebas untuk develop. Tidak ada lisensi atau biaya royalty untuk dikembangkan pada *platform* Android. Tidak ada biaya keanggotaan diperlukan. Aplikasi untuk Android dapat didistribusikan dan diperdagangkan dalam bentuk apapun.

Android merupakan generasi baru *platform mobile*, *platform* yang memberikan pengembangan untuk melakukan pengembangan sesuai dengan yang diharapkannya. GNU, General Public Lisensi Versi 2 (GPLv2) merupakan lisensi dari sistem operasi yang mendasari Android yang sering dikenal dengan istilah “copyleft” lisensi dimana setiap perbaikan pihak ketiga harus terus jatuh di bawah terms. Android didistribusikan di bawah lisensi Apache Software (ASL/Apache2), yang memungkinkan untuk distribusi kedua dan seterusnya. Komersialisasi pengembang (*produsen* handset khususnya) dapat memilih untuk meningkatkan *platform* tanpa harus memberikan perbaikan mereka kepada masyarakat Open Source. Sebaliknya, pengembang dapat keuntungan dari perangkat tambahan seperti perbaikan dan mendistribusikan ulang pekerjaan mereka dibawah lisensi apapun yang mereka inginkan.

Pengembang memiliki beberapa pilihan ketika membuat aplikasi yang berbasis Android. Salah satunya adalah Eclipse yang paling sering digunakan banyak pengembang karena tersedia secara bebas untuk merancang dan mengembangkan aplikasi Android. Eclipse adalah IDE yang paling populer untuk pengembangan Android, karena memiliki Android plug-in yang tersedia untuk memfasilitasi pengembang Android. Selain itu Eclipse juga mendapatkan dukungan langsung dari Google untuk menjadi IDE pengembangan aplikasi Android, ini terbukti dengan adanya penambahan plugins untuk Eclipse untuk membuat project Android di mana source software langsung dari situs resminya Google. Tetapi hal diatas tidak menutup kemungkinan untuk menggunakan IDE

yang lain seperti Netbeans untuk melakukan pengembangan Android. Aplikasi Android dapat dikembangkan pada system operasi berikut:

- a. Windows XP Vista/7 dan 8.
- b. Mac OS X (Mac OS X 10.4.8 atau lebih baru).
- c. Linux.

### 2.2.3.1 The *Dalvik Virtual Machine* (DVM)

Salah satu elemen kunci dari Android adalah *Dalvik Virtual Machine* (DVM) [12]. Android berjalan di dalam *Dalvik Virtual Machine* (DVM) bukan di Java Virtual Machine (JVM), sebenarnya banyak persamaan dengan Java Virtual Machine (JVM) seperti Java ME (Java *Mobile* Edition), tetapi Android menggunakan *Virtual Machine* sendiri yang menurut saya dikustomisasi dan dirancang untuk memastikan bahwa beberapa feature-feature berjalan lebih efisien pada perangkat *mobile*.

*Dalvik Virtual Machine* (DVM) adalah “*register bases*” sementara Java Virtual Machine (JVM) adalah “*stack based*”, DVM didesain dan ditulis oleh Dan *Bornsten* dan beberapa *engineers Google* lainnya. Jadi bisa kita katakana “*Dalvik equals(Java) == False*”. *Dalvik Virtual Machine* menggunakan kernel Linux untuk menangani fungsionalitas tingkat rendah termasuk keamanan, threading, dan proses serta manajemen memori. Ini memungkinkan kita untuk menulis Aplikasi C/C+ sama halnya seperti OS Linux kebanyakan. Meskipun dalam kenyataannya kita harus banyak memahami Arsitektur dalam proses system dari kernel linux yang digunakan dalam Android tersebut.

Semua hardware yang berbasis Android dijalankan menggunakan *Virtual Machine* untuk eksekusi aplikasi, pengembang tidak perlu khawatir tentang implementasi perangkat keras tertentu. *Dalvik Virtual Machine* mengeksekusi *executablefile*, sebuah format yang dioptimalkan untuk memastikan memori yang digunakan sangat kecil. *The Executable file* diciptakan dengan mengubah kelas bahasa java dan dikompilasi menggunakan tools yang disediakan dalam SDK Android.

### 2.2.3.2 Android SDK (*Software Development Kit*)

Android SDK *tools* API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java [12]. Android merupakan subset perangkat lunak untuk ponsel yang meliputi system operasi, *middleware* dan aplikasi kunci yang di rilis oleh Google. Saat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada *platform* menggunakan Bahasa pemrograman Java.

Android memberi kita kesempatan untuk membuat aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan *Handphone/Smartphone*. Beberapa fitur-fitur Android yang paling penting adalah:

1. *Framework* Aplikasi yang mendukung penggantian komponen dan reusable.
2. Mesin *Virtual Dalvik* dioptimalkan untuk perangkat *mobile*.
3. *Integrated Browser* berdasarkan engine *open source* WebKit.
4. Grafis yang dioptimalkan dan didukung oleh libraries grafis 2D, grafis 3D berdasarkan spesifikasi opengLES 1,0(Opsional akselerasi hardware)
5. SQLite untuk penyimpanan data.
6. Media Support yang mendukung audio, video, dan gambar (MPEG4, H.164, MP3, AAC, AMR, JPG, PNG, GIF), GSM Telephony (tergantung hardware).
7. Bluetooth, EDGE, 3G, dan WiFi (tergantung hardware).
8. Kamera, GPS, Kompas, dan accelerometer (tergantung hardware).
9. Lingkungan Development yang lengkap dan kaya termasuk perangkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin untuk IDE Eclipse.

### 2.2.3.3 ADT (*Android Development Tools*)

*Android Development Tools* (ADT) adalah plugin yang didesain untuk IDE Eclipse yang memberikan kita kemudahan dalam mengembangkan aplikasi android dengan menggunakan IDE Eclipse [12]. Dengan menggunakan ADT untuk Eclipse

akan memudahkan kita dalam membuat aplikasi project android, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya, selain itu juga dapat melakukan running aplikasi menggunakan Android SDK melalui Eclipse. Dengan ADT juga kita dapat melakukan pembuatan *package* Android (.apk) yang digunakan untuk distribusi aplikasi android yang kita rancang.

Mengembangkan aplikasi Android dengan menggunakan ADT di eclipse sangat dianjurkan dan sangat mudah untuk memulai pengembangan aplikasi Android. Berikut adalah versi ADT untuk Eclipse yang sudah dirilis:

1. ADT 12.0.0 (July 2011).
2. ADT 11.0.0 (June 2011).
3. ADT 10.0.1 (March 2011).
4. ADT 10.0.0 (February 2011).
5. ADT 9.0.0 (January 2011).
6. ADT 8.0.1 (December 2010).
7. ADT 8.0.0 (December 2010).
8. ADT 0.9.9 (September 2010).
9. ADT 0.9.8 (September 2010).
10. ADT 0.9.7 (May 2010).
11. ADT 0.9.6 (March 2010).
12. ADT 0.9.5 (December 2009).
13. ADT 0.9.4 (October 2009).

Semakin tinggi *platform* android yang kita gunakan, semakin lebih terbaru pula ADT yang digunakan, Karena biasanya munculnya *platform* baru diikuti oleh munculnya versi ADT yang terbaru.

#### **2.2.4 Framework**

*Framework* secara sederhana dapat diartikan kumpulan dari fungsi-fungsi/prosedur-prosedur dan *Class-Class* untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang programmer, tanpa harus membuat fungsi atau *Class* dari awal [13]. Ada beberapa alasan mengapa menggunakan *Framework*:

1. Mempercepat dan mempermudah pembangunan sebuah aplikasi *web*.
2. Relatif memudahkan dalam proses maintenance karena sudah ada pola tertentu dalam sebuah *framework* (dengan syarat programmer mengikuti pola standar yang ada).
3. Umumnya *framework* menyediakan fasilitas-fasilitas yang umum dipakai sehingga kita tidak perlu membangun dari awal (misalnya validasi, ORM, pagination, multiple database, scaffolding, pengaturan session, error handling, dll).
4. Lebih bebas dalam pengembangan jika dibandingkan CMS.

*Framework* menganut konsep *Model View Controller* (MVC) merupakan suatu konsep yang cukup populer dalam pembangunan aplikasi *web* (Mulyanto, 2012). Berawal pada bahasa pemrograman Small Talk, MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, *user* interface, dan bagian yang menjadi kontrol aplikasi. Terdapat 3 jenis komponen yang membangun suatu MVC pattern dalam suatu aplikasi yaitu :

1. *View*, merupakan bagian yang menangani presentation logic. Pada suatu aplikasi *web* bagian ini biasanya berupa file *template* HTML, yang diatur oleh *controller*. *View* berfungsi untuk menerima dan merepresentasikan data kepada *user*. Bagian ini tidak memiliki akses langsung terhadap bagian model.
2. *Model*, biasanya berhubungan langsung dengan database untuk memanipulasi data (*insert, update, delete, search*), menangani validasi dari bagian controller, namun tidak dapat berhubungan langsung dengan bagian view.
3. *Controller*, merupakan bagian yang mengatur hubungan antara bagian model dan bagian *view*, *controller* berfungsi untuk menerima request dan data dari *user* kemudian menentukan apa yang akan diproses oleh aplikasi.

Dengan menggunakan prinsip MVC suatu aplikasi dapat dikembangkan sesuai dengan kemampuan developernya, yaitu programmer yang menangani bagian *model* dan *controller*, sedangkan *designer* yang menangani bagian *view*,

sehingga penggunaan arsitektur MVC dapat meningkatkan *maintanability* dan organisasi kode. Walaupun demikian dibutuhkan komunikasi yang baik antara programmer dan designer dalam menangani variabel-variabel yang akan ditampilkan.

### 2.2.5 Codeigniter

Codeigniter adalah sebuah framework PHP yang dapat membantu mempercepat *developer* dalam pengembangan aplikasi *web* berbasis PHP dibandingkan jika menulis semua kode program dari awal. Codeigniter menyediakan banyak *library* untuk mengerjakan tugas-tugas yang umumnya ada pada sebuah aplikasi berbasis *web* [14]. Selain itu, struktur dan susunan logis dari codeigniter membuat aplikasi yang dibuat menjadi semakin teratur dan rapi. Dengan demikian, dapat fokus pada fitur-fitur apa yang dibutuhkan aplikasi dengan membuat kode program seminimal mungkin.

Ada beberapa kelebihan CodeIgniter (CI) dibandingkan dengan Framework PHP lain yaitu [14]:

1. Performa cepat.

Salah satu alasan tidak menggunakan framework adalah karena eksekusinya yang lebih lambat daripada PHP from the scratch, tapi Codeigniter sangat cepat bahkan mungkin bisa dibilang codeigniter merupakan framework yang paling cepat dibanding framework yang lain.

2. Konfigurasi yang sangat minim (nearly zero configuration).

Tentu saja untuk menyesuaikan dengan database dan keleluasaan routing tetap diizinkan melakukan konfigurasi dengan mengubah beberapa file konfigurasi seperti database, PHP atau Autoload PHP, namun untuk menggunakan codeigniter dengan *setting* standard, anda hanya perlu mengubah sedikit saja file pada folder config.

3. Banyak komunitas.

Dengan banyaknya komunitas CI ini, memudahkan kita untuk berinteraksi dengan yang lain, baik itu bertanya atau teknologi terbaru.

4. Dokumentasi yang sangat lengkap.

Setiap paket instalasi codeigniter sudah disertai *user guide* yang sangat bagus dan lengkap untuk dijadikan permulaan, bahasanya pun mudah dipahami.

#### 5. Gratis.

Codeigniter berlisensi dibawah Apache/BSD open source, jadi Anda bisa menggunakannya secara bebas. Untuk informasi lebih lanjut, anda bisa membaca license agreement yang dapat dibaca saat instalasi codeigniter.

#### 6. Menggunakan PHP 4.

Meskipun codeigniter dapat berjalan pada PHP 5, namun sampai saat ini kode program codeigniter masih dibuat dengan menggunakan PHP 4. Hal ini dilakukan agar codeigniter dapat tersebar lebih luas di komunitas PHP. Karena hingga saat ini, sebagian besar *web* hosting masih menggunakan PHP 4. Jika codeigniter dibuat dengan PHP 5, tentu saja hasilnya akan jauh lebih canggih, karena bisa memanfaatkan teknologi PHP 5 yang saat ini masih belum dapat dilakukan oleh PHP 4, misalnya untuk menerapkan konsep OOP Multiple Inheritance.

#### 7. Berukuran kecil.

Ukuran codeigniter yang kecil merupakan keunggulan tersendiri. Dibanding framework lainnya yang berukuran besar, serta membutuhkan resource yang besar pula untuk berjalan. Pada codeigniter, bisa diatur agar sistem meload library yang dibutuhkan saja, sehingga dapat berjalan ringan dan cepat.

#### 8. Menggunakan konsep M-V-C.

Codeigniter menggunakan konsep M-V-C (*Model-View-Controller*) yang memungkinkan pemisahan antara layer application-logic dan presentation.

#### 9. URL yang sederhana.

Secara default, URL yang dihasilkan Codeigniter sangat bersih (*Clean*) dan *Search Engine friendly* (SEF).

#### 10. Memiliki paket library yang lengkap.

Codeigniter memiliki library yang lengkap untuk mengerjakan operasi-operasi yang umum dibutuhkan oleh sebuah aplikasi berbasis *web*, misalnya

mengakses database, mengirim email, memvalidasi form, menangani session, dan sebagainya.

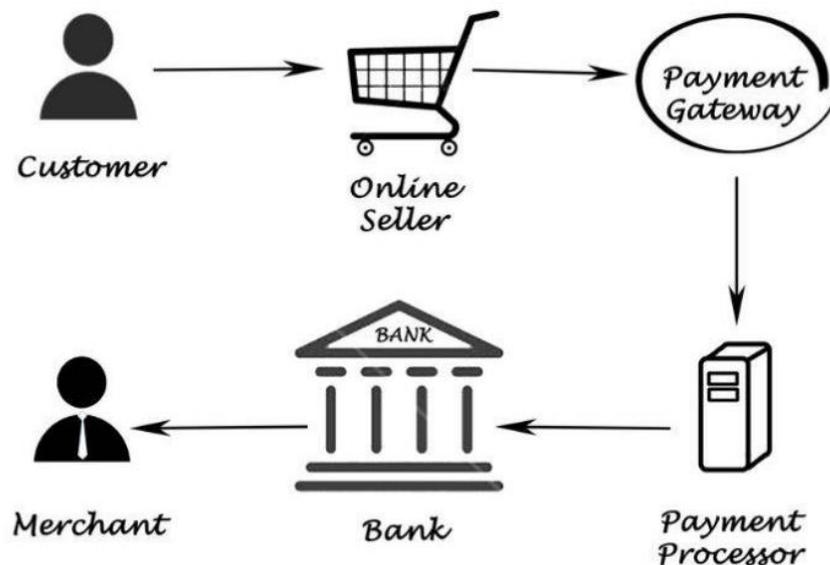
11. Extensible.

Sistem dapat dikembangkan dengan mudah dengan menggunakan plugin dan helper, atau dengan menggunakan hooks.

12. Tidak memerlukan Template Engine.

### 2.2.6 Payment Gateway

*Payment gateway* adalah salah satu cara untuk memproses transaksi elektronik. *Paymentgateway* menyediakan alat – alat untuk memproses pembayaran antara *customer*, *businesses* dan *banks*. *Payment gateway* merupakan bagian terpenting dari suatu transaksi antar *customer*, *business*, dan lembaga – lembaga perbankan yang keduanya digunakan. *Payment Gateway* digunakan untuk memfasilitasi transaksi elektronik. Beberapa fitur – fitur utama dari *payment gateway* meliputi enkripsi pembayaran dan data pribadi, komunikasi antara lembaga keuangan yang terkait, *business* dan *customer*, otorisasi pembayaran. Beberapa tools mempunyai fitur *payment gateway* yang dapat membantu pelanggan mengetahui biaya pengiriman dan penanganan, serta pajak penjualan. Ada juga pendeteksian untuk penipuan dan fitur – fitur lainnya yang dapat digunakan dengan *payment gateway* [4].



Sumber gambar : Mekanisme Payment Gateway via shutterstock.com

**Gambar 2.2 Mekanisme Payment Gateway.**

Penjelasan dari gambar diatas :

1. Pelanggan toko online melihat dan memutuskan untuk membeli suatu produk berdasarkan informasi yang disampaikan pada koneksi sumber *payment gateway*.

2. *Payment gateway* akan meneruskan informasi tersebut ke prosesor pembayaran bank pelanggan.
3. Prosesor pembayaran akan meneruskan informasi transaksi ke asosiasi penerbit kartu yang digunakan, misalnya Mastercard atau Visa
4. Tahap selanjutnya, bank terkait akan menerima permintaan ini dan mengirimkan balasan ke prosesor dengan kode khusus serta memberikan konfirmasi apakah transaksi tersebut berhasil atau gagal.
5. Prosesor pembayaran akan mengirimkan pesan itu ke *payment gateway* kemudian akan diteruskan ke *website* atau aplikasi penjual serta pemegang kartu hingga transaksi dinyatakan berhasil.

### 2.2.7 PHP

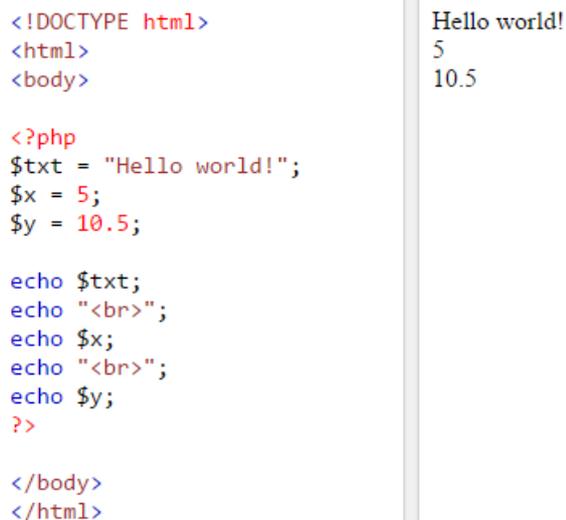
PHP adalah pemrograman interpreter yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan [14]. PHP disebut juga pemrograman Server Side Programming, hal ini dikarenakan seluruh prosesnya dijalankan pada server. PHP adalah suatu bahasa dengan hak cipta terbuka atau yang juga dikenal dengan open source yaitu pengguna data mengembangkan kode-kode fungsi sesuai kebutuhannya.

PHP (*Perl Hypertext Preprocessor*) adalah bahasa server-side-scripting yang menyatu dengan HTML untuk membuat halaman *web* yang dinamis [15]. Dengan menggunakan program PHP, sebuah *website* akan lebih interaktif dan dinamis. Kelebihan-kelebihan dari PHP yaitu:

1. PHP merupakan sebuah bahasa script yang tidak melakukan sebuah kompilasi dalam penggunaannya. Tidak seperti halnya bahasa pemrograman aplikasi yang lainnya.
2. PHP dapat berjalan pada *web* server yang dirilis oleh Microsoft, seperti IIS atau PWS juga pada apache yang bersifat open source.
3. Karena sifatnya yang open source, maka perubahan dan perkembangan interpreter pada PHP lebih cepat dan mudah, karena banyak milis-milis dan developer yang siap membantu pengembangannya.

4. Jika dilihat dari segi pemahaman, PHP memiliki referensi yang begitu banyak sehingga sangat mudah untuk dipahami.
5. PHP dapat berjalan pada 3 operating sistem, yaitu: Linux, unix, dan windows, dan juga dapat dijalankan secara runtime pada suatu console.

Contoh penerapan PHP pada pembangunan *web*:



```

<!DOCTYPE html>
<html>
<body>

<?php
$txt = "Hello world!";
$x = 5;
$y = 10.5;

echo $txt;
echo "<br>";
echo $x;
echo "<br>";
echo $y;
?>

</body>
</html>

```

Hello world!  
5  
10.5

**Gambar 2.3 Contoh Sintaks PHP.**

### 2.2.8 MYSQL

MySQL adalah Relational *Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat closed source atau komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian database, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis [14]. Keandalan suatu sistem *database* (DBMS) dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh *user* maupun program-program aplikasinya. MySQL biasanya digunakan atau

diinstall bersamaan dengan XAMPP sehingga untuk melihat isi tabel bisa menggunakan *PHPmyAdmin*.

Contoh Penerapan SQL dalam database:

SQL Statement:	Hasil:
<code>SELECT Country FROM Customers;</code>	<b>Country</b>
	Germany
	Mexico
	Mexico
	UK
	Sweden
	Germany
	France

**Gambar 2.4 Contoh Sintaks SQL.**

### 2.2.9 HTML

*Hypertext Markup Language* (HTML) adalah bahasa yang digunakan untuk menulis halaman *web*. HTML merupakan pengembangan dari standar pemformatan dokumen teks yaitu *Standard Generalized Markup Language* (SGML) [14]. HTML sebenarnya adalah dokumen ASCII atau teks biasa, yang dirancang untuk tidak tergantung pada suatu sistem operasi tertentu.

Mendesain HTML berarti melakukan suatu tindakan pemrograman. Namun HTML bukanlah sebuah bahasa pemrograman. Namun HTML hanyalah berisi perintah-perintah yang telah terstruktur berupa tag-tag penyusun. Menuliskan tag-tag HTML tidaklah sebatas hanya memasukkan perintah-perintah tertentu agar HTML kita dapat di akses oleh browser. Mendesain HTML adalah sebuah seni tersendiri. Homepage yang merupakan implementasi dari HTML adalah refleksi dari orang yang membuatnya. Untuk itu kita perlu mendesainnya dengan baik agar para pengunjung homepage yang kita buat merasa senang dan bermanfaat.

Mendesain HTML dapat dilakukan dengan cara menggunakan HTML Editor, seperti notepad++, adobe dreamweaver dan lain-lain.

Contoh Penerapan Sintaks HTML:



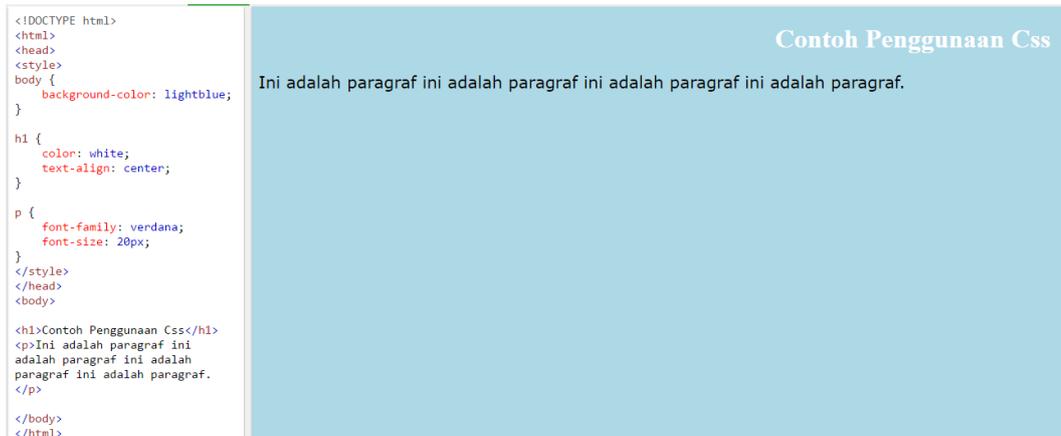
**Gambar 2.5 Contoh Sintaks HTML.**

### 2.2.10 Cascading Style Sheets (CSS)

*Cascading Style Sheets* (CSS) adalah suatu bahasa stylesheet yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa markup [14]. Penggunaan yang paling umum dari CSS adalah untuk memformat halaman *web* yang ditulis dengan HTML dan XHTML. Walaupun demikian, bahasanya sendiri dapat dipergunakan untuk semua jenis dokumen XML termasuk SVG dan XUL. Spesifikasi CSS diatur oleh World Wide *Web* Consortium (W3C).

CSS digunakan oleh penulis maupun pembaca halaman *web* untuk menentukan warna, jenis huruf, tata letak, dan berbagai aspek tampilan dokumen. CSS digunakan terutama untuk memisahkan antara isi dokumen (yang ditulis dengan HTML atau bahasa markup lainnya) dengan presentasi dokumen (yang ditulis dengan CSS). Pemisahan ini dapat meningkatkan aksesibilitas isi, memberikan lebih banyak keleluasaan dan kontrol terhadap tampilan, dan mengurangi kompleksitas serta pengulangan pada stuktur isi.

Beriku Contoh Sintaks Sederhana CSS:



**Gambar 2.6 Contoh Sintaks CSS.**

### 2.2.11 *Black Box Testing*

Tester menggunakan *behavioral test* (disebut juga *Black Box Test*), sering digunakan untuk menemukan *bug* dalam *high level operations*, pada tingkatan fitur, profil operasional dan skenario customer [16]. Tester dapat membuat pengujian fungsional *blackbox* berdasarkan pada apa yang harus sistem lakukan. *Behavioral testing* melibatkan pemahaman rinci mengenai domain aplikasi, masalah bisnis yang dipecahkan oleh sistem dan misi yang dilakukan sistem. *Behavioral test* paling baik dilakukan oleh penguji yang memahami desain sistem, setidaknya pada tingkat yang tinggi sehingga mereka dapat secara efektif menemukan bug umum untuk jenis desain. *Blackbox testing* juga disebut *functional testing*, sebuah teknik pengujian fungsional yang merancang test *case* berdasarkan informasi dari spesifikasi.

