

BAB 2

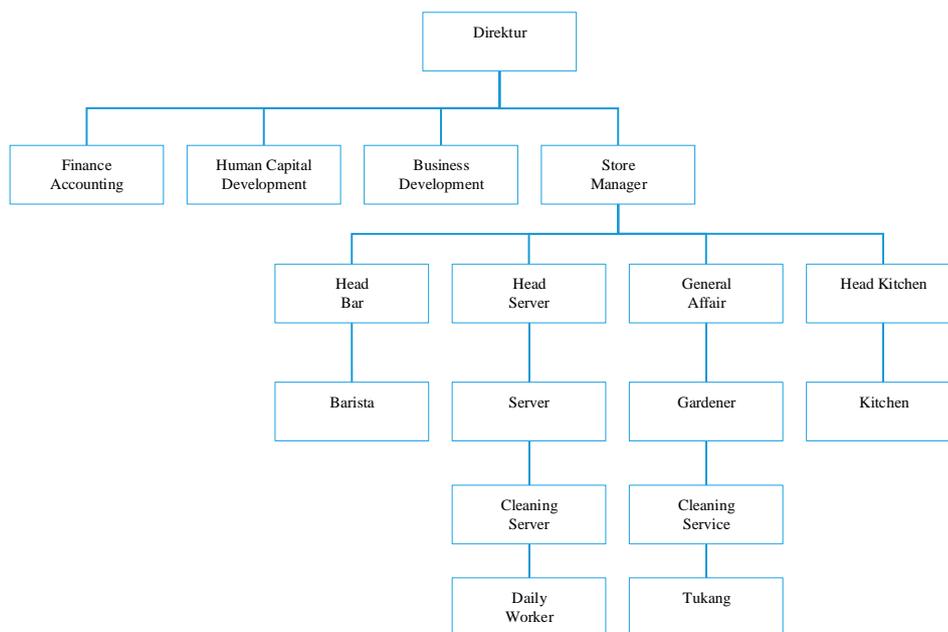
TINJAUAN PUSTAKA

2.1 Profil Tempat Penelitian

Kedai XYZ kopi merupakan usaha yang bergerak di bidang kuliner yang bertempat di Jl. Ir. H. Juanda, Dago Atas, Bandung. Dengan luas tempat 2400 meter persegi, kedai XYZ kopi menyediakan dua pilihan tempat, yaitu didalam ruangan dan diluar ruangan dengan tema alam yang dikelilingi oleh pohon pinus, ada 10 meja yang disediakan untuk pelanggan didalam ruangan, sedangkan diluar ruangan terdapat kurang lebih ada 100 meja yang disediakan. Jarak antara tempat dalam ruangan dan luar ruangan kurang lebih 30 meter. Kedai XYZ Kopi merupakan salah satu kedai yang cukup populer dan ramai pengunjung di Kota Bandung, tidak kurang dari 350 orang yang datang setiap hari ke kedai, bahkan pada akhir pekan bias mencapai 500 orang yang datang.

2.1.1 Struktur Organisasi

Struktur Organisasi Kedai XYZ Kopi dapat dilihat pada gambar



Gambar 2.1 Struktur Organisasi Kedai XYZ Kopi

Adapun deskripsi kerja pada gambar struktur organisasi adalah sebagai berikut :

1. Direktur.
Bertanggung jawab atas segala kegiatan yang berjalan di Kedai XYZ Kopi.
2. Finance Accounting.
Bertanggung jawab terhadap pencarian, pengelolaan, pengalokasian dana, dan melakukan pembayaran.
3. Human Capital Development.
Bertanggung jawab terhadap peningkatan kinerja karyawan melalui pelatihan dalam peningkatan mutu organisasi perusahaan.
4. Business Development.
Bertugas dalam melakukan riset terhadap pasar, mencari peluang pelanggan baru, dan menjaga hubungan dengan pelanggan.
5. Store Manager.
Bertanggung jawab atas kondisi seluruh kedai dan para staff yang berada di Kedai 32 Kopi.
6. Head Bar.
Bertanggung jawab atas seluruh pekerjaan bar, pengadaan bahan, dan pengolahan bahan yang sesuai dengan standar yang telah ditetapkan.
7. Head Server.
Bertanggung jawab dalam mengatur strategi agar semua permasalahan service kepada pelanggan bisa ditangani dengan baik.
8. General Affair.
Bertugas mendukung kegiatan operasional di Kedai XYZ Kopi.
9. Head Kitchen.
Bertanggung jawab atas seluruh pekerjaan dapur, pengadaan bahan, dan pengolahan bahan di dapur yang sesuai dengan standar yang telah ditetapkan.
10. Barista.
Bertugas dalam menyediakan minuman untuk pelanggan, dan memastikan alat yang digunakan tetap higienis.

11. Server.

Bertugas memberikan pelayanan yang baik kepada pelanggan.

12. Gardener.

Bertugas memelihara area taman dan sekitar Kedai XYZ Kopi.

13. Kitchen.

Bertugas dalam menyediakan makanan untuk pelanggan, dan memastikan alat yang digunakan tetap higienis.

14. Cleaning Server.

Bertugas merapikan tempat/meja untuk pelanggan agar pelanggan merasa nyaman.

15. Cleaning Service.

Bertugas merapikan area Kedai XYZ Kopi agar tetap terasa nyaman untuk pelanggan.

16. Daily Worker.

Membantu pekerjaan Cleaning Server/Cleaning Service.

17. Tukang.

Bertugas mengurus semua pekerjaan yang berhubungan dengan bangunan yang ada di Kedai XYZ Kopi.

2.2 Landasan Teori

Landasan Teori adalah seperangkat definisi, konsep serta proposisi yang telah disusun rapi serta sistematis tentang variabel-variabel dalam sebuah penelitian. Landasan teori ini akan menjadi dasar dalam pembangunan aplikasi transaksi menu di Kedai XYZ Kopi menggunakan *qr-code* dan *one time password* berbasis *e-wallet*, adapun teori yang terkait dengan pembanguna sistem ini yang didalamnya berhubungan dengan perangkat keras, perangkat lunak, serta bahasa pemrograman yang digunakan.

2.2.1 Transaksi Digital

Menurut Indra Bastian, Transaksi adalah suatu pertemuan antara dua belah pihak (penjual dan pembeli) yang saling menguntungkan dengan adanya data, bukti atau dokumen pendukung [18]. Sedangkan menurut Slamet Wiyono, Transaksi dapat diartikan sebagai kejadian ekonomi atau keuangan yang melibatkan paling tidak dua pihak (seseorang dengan seseorang atau beberapa orang lainnya) yang saling melakukan pertukaran, melibatkan diri dalam perserikatan usaha, pinjam meminjam dan lainnya atas dasar suatu ketentuan hukum yang berlaku [19].

Berdasarkan definisi transaksi menurut Indra Bastian dan Slamet Wiyono, dapat diambil kesimpulan bahwa Transaksi adalah suatu pertemuan yang melibatkan paling tidak dua pihak yang melakukan pertukaran, melibatkan diri dalam perserikatan usaha, pinjam meminjam yang saling menguntungkan satu sama lain atas dasar suatu ketentuan dalam hukum yang berlaku.

Salah satu inovasi dalam transaksi digital adalah Blockchain. Blockchain merupakan database yang terdistribusi berupa catatan atau buku besar dari semua transaksi digital yang tersebar pada semua pengguna sistem [20]. Blockchain juga menerapkan teknologi *signature* berbasis kriptografi yang memungkinkan setiap kali terjadi transaksi, blockchain akan mengautentikasi data pengguna secara *realtime* sebelum transaksi disahkan.

2.2.2 Uang Elektronik

Teknologi pembayaran secara elektronik tidak hanya menggantikan cek tetapi juga tunai dalam bentuk *electronic money* atau *e-money* (adanya uang hanya dalam bentuk elektronik) [21]. Menurut Frederic S Mishkin dalam buku yang berjudul Ekonomi Uang, Perbankan, dan Pasar Keuangan, ada tiga bentuk dari *e-money* [21], diantaranya adalah :

1. Kartu Debit

Kartu Debit yang bentuknya seperti kartu kredit, memungkinkan konsumen membeli barang dan jasa dengan secara langsung dapat memindahkan dana secara elektronik dari rekening pihak pembeli ke rekening pihak penjual melalui alat pembaca kartu.

2. *Stored-value Card*

Stored-value Card dikenal sebagai *smart card*, berisi chip komputer yang dapat mengakses tunai secara digital dari pemilik rekening kapanpun dibutuhkan pada mesin ATM. Bentuk yang paling sederhana dari *stored-value card* adalah dibeli dengan dollar tertentu yang dibayar di muka, mirip seperti kartu telepon Prabayar.

3. *E-Cash*

E-Cash dapat digunakan untuk membeli barang atau jasa melalui internet. Pelanggan dapat mendapatkan *E-Cash* dengan membuat rekening di bank yang mempunyai jaringan internet dan kemudian bias mempunyai *E-Cash* yang dipindahkan dari komputer pribadinya.

Uang elektronik merupakan pengembangan lebih lanjut kartu kredit. Berbeda dengan kartu kredit yang terlebih dahulu harus menandatangani struk pembelian atau menekan nomor sandi pada mesin, dengan uang elektronik pelanggan hanya menorehkan uang elektroniknya pada alat pembaca dan secara otomatis jumlah pembelian barang atau jasa akan berpindah kedalam komputer kasir [22].

Beberapa Bank Indonesia telah menawarkan uang elektronik, baik berbasis *chip* misalnya *e-Money* oleh Bank Mandiri, *TapCash* oleh Bank BNI, *Brizzi* oleh Bank BRI, *Flazz* oleh Bank BCA, *MegaCash* oleh Bank Mega, dan *JakCard* oleh Bank DKI. Adapun uang elektronik berbasis *server* yang dikenal juga sebagai dompet elektronik (*e-wallet*) yang berbeda dengan kartu Prabayar, layanan *e-wallet* tidak hanya disediakan oleh bank melainkan juga disediakan oleh lembaga selain bank seperti operator telekomunikasi misalnya Saku-ku milik Bank BCA, *Mandiri eCash* milik Bank Mandiri, Rekening Ponsel bank

CIMB Niaga T-Cash milik Telkomsel, Go-Pay milik Go-Jek, OVO milik Grab, dan lainnya.

Menurut Prof. Dr. Damsar dalam buku yang berjudul Pengantar Sosiologi Pasar ada tiga kekurangan dari layanan *e-wallet* [22], diantaranya adalah :

1. Layanan *e-wallet* sangat terbatas, seiring berkembangnya *e-wallet* pengguna seringkali kesulitan karena tidak ada satupun *e-wallet* yang multiguna atau bisa digunakan untuk beragam transaksi. Jika ingin berbelanja di *merchant* yang tidak ada ikatan kerjasama dengan penyedia layanan *e-wallet*, pengguna mau tidak mau harus melakukan pembayaran secara tunai.
2. Saldo yang ada pada *e-wallet* mengendap tidak berkembang, sampai saat ini belum ada *e-wallet* yang menawarkan bunga atau imbas hasil saldo yang tersisa.
3. Uang yang ada pada *e-wallet* tidak bisa ditarik tunai dan hanya bisa untuk transaksi *payment*. Seiring perkembangan ada beberapa *e-wallet* dengan fasilitas penarikan tunai atau transfer, tetapi jumlah yang dapat ditarik tunai terbatas dan fasilitas transfer pun hanya dapat dilakukan ke sesama pemilik aplikasi.

1. Manfaat

Penggunaan Uang Elektronik sebagai alat pembayaran dapat memberikan manfaat sebagai berikut [23]:

1. Memberikan kemudahan dan kecepatan dalam melakukan transaksi transaksi pembayaran tanpa perlu membawa uang tunai.
2. Tidak lagi menerima uang kembalian dalam bentuk barang (seperti permen) akibat pedagang tidak mempunyai uang kembalian bernilai kecil (receh).
3. Sangat applicable untuk transaksi massal yang nilainya kecil namun frekuensinya tinggi, seperti: transportasi, parkir, tol, fast food, dll.

2. Dasar Hukum

Penyelenggaraan Uang Elektronik telah diatur dalam [23]:

1. Peraturan Bank Indonesia Nomor 11/12/PBI/2009 tanggal 13 April 2009 tentang Uang Elektronik (Electronic Money).
2. Surat Edaran Bank Indonesia No.11/11/DASP tanggal 13 April 2009 perihal Uang Elektronik (Electronic Money).

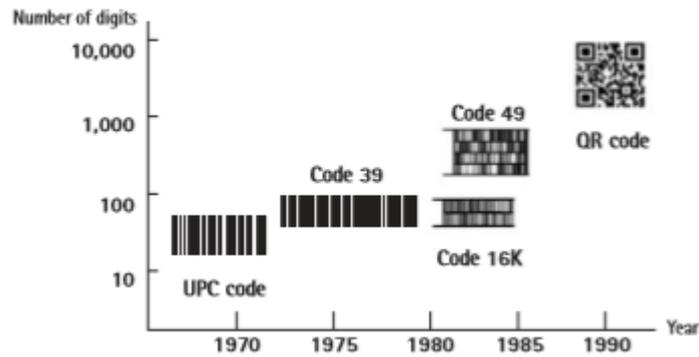
2.2.3 QR-Code

Qr code merupakan teknik yang mengubah data tertulis menjadi kode-kode 2 dimensi yang tercetak kedalam suatu media yang lebih ringkas. Qr code adalah barcode 2 dimensi yang diperkenalkan pertama kali oleh perusahaan Jepang Denso Wave pada tahun 1994. Barcode ini pertama kali digunakan untuk pendataan inventaris produksi suku cadang kendaraan dan sekarang sudah digunakan dalam berbagai bidang. Qr adalah singkatan dari Quick Response karena ditujukan untuk diterjemahkan isinya dengan cepat. QR-Code salah satu tipe dari barcode yang dapat dibaca dengan kamera handphone [24].

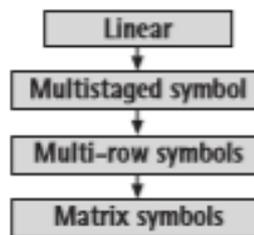


Gambar 2.2 Gambar Kode QR

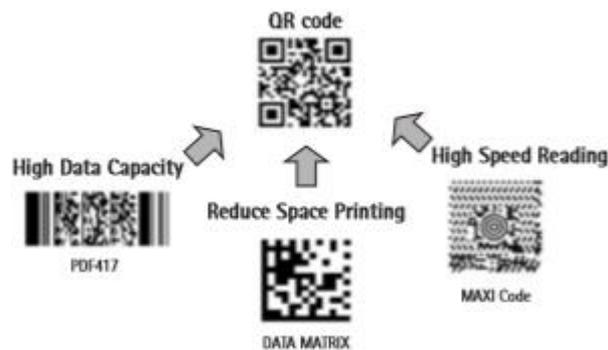
Simbol dua dimensi umumnya mengandung jumlah data yang jauh lebih banyak bila dibandingkan dengan simbol linier (sekitar 100 kali lebih banyak), dan karenanya memerlukan waktu pemrosesan data yang lebih lama dan proses yang lebih kompleks. Oleh karena itu, QR-Code telah banyak mempertimbangkan pola findernya untuk memungkinkan pembacaan kecepatan tinggi [25]. Adapun sejarah dari simbol QR-Code dapat dilihat pada gambar 2.2.



Gambar 2.3 Sejarah Simbol QR-Code



Gambar 2.4 Tahapan Pengembangan Simbol QR-Code



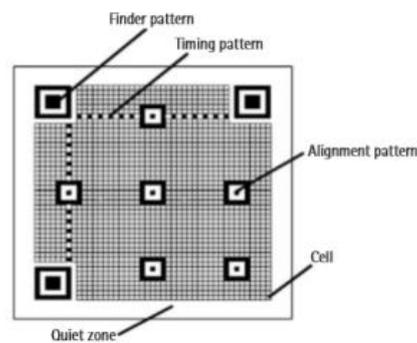
Gambar 2.5 Tahap Pengembangan QR-Code

Qr-code mampu menyimpan semua jenis data, seperti data angka/numerik, alphanumeric, biner, kanji/kana. Selain itu Qr-Code mampu menampung data secara horizontal dan vertikal, jadi secara otomatis ukuran dari tampilan gambar QRCode bisa hanya persepuluh dari ukuran sebuah barcode. Tiga tanda berbentuk persegi di tiga sudut memiliki fungsi agar simbol dapat dibaca dengan hasil yang sama dari sudut manapun [26].

Adapun beberapa karakteristik dari QR-Code adalah sebagai berikut [25]:

- a. Dapat dibaca dari semua arah (360°) dengan kecepatan tinggi.
- b. Tahan terhadap distorsi simbol, Simbol sering terdistorsi saat menempel pada permukaan melengkung atau oleh pembaca yang dimiringkan (miring antara wajah sensor CCD dan wajah simbol).
- c. Fungsi pemulihan data (Simbol tahan terhadap luntur atau rusak).
- d. Pengkodean karakter kanji dan kana yang efisien.

Adapun struktur dari QR-Code dapat dilihat pada gambar 2.6 [25].



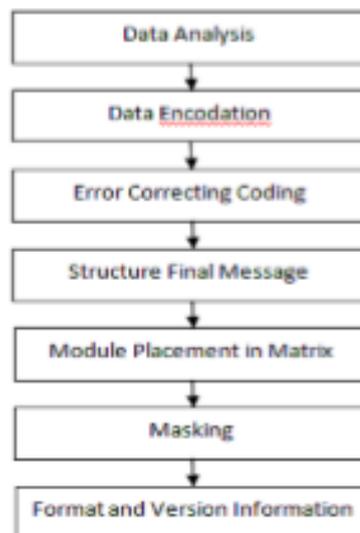
Gambar 2.6 Struktur QR-Code

Adapun penjelasan dari gambar 2.5 adalah sebagai berikut :

- a. Pola Finder. Pola untuk mendeteksi posisi QR Code. Dengan mengatur pola ini di tiga sudut simbol, posisi, ukuran, dan sudut simbol dapat dideteksi. Pola pencari ini terdiri dari struktur yang dapat dideteksi di semua arah (360°).
- b. Alignment Pattern Pola untuk memperbaiki distorsi QR Code. Ini sangat efektif untuk memperbaiki distorsi nonlinier. Koordinat pusat dari pola pelurusan akan diidentifikasi untuk memperbaiki distorsi simbol. Untuk tujuan ini, sel yang terisolasi hitam ditempatkan dalam pola pelurusan untuk membuatnya lebih mudah untuk mendeteksi koordinat pusat dari pola pelurusan.

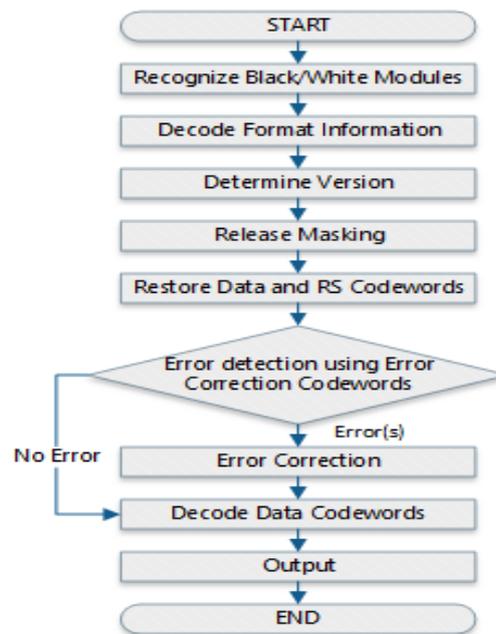
- c. Pola Waktu. Pola untuk mengidentifikasi koordinat pusat dari setiap sel dalam QR Code dengan pola hitam dan putih yang disusun secara bergantian. Ini digunakan untuk mengoreksi koordinat pusat sel data ketika simbol terdistorsi atau ketika ada kesalahan untuk pitch sel. Ini diatur dalam arah vertikal dan horizontal.
- d. Zona Hening. Merupakan ruang atau margin yang diperlukan untuk membaca QR Code. Zona hening ini memudahkan simbol dideteksi dari antara gambar yang dibaca oleh sensor CCD. Diperlukan empat atau lebih sel untuk zona hening.
- e. Area Data. Data QR Code akan disimpan (disandikan) ke dalam area data. Bagian abu-abu pada Gambar 2.5 mewakili area data. Data akan dikodekan ke dalam angka biner dari '0' dan '1' berdasarkan aturan pengkodean. Angka-angka biner dari '0' dan '1' akan dikonversi menjadi sel-sel hitam dan putih dan kemudian akan diatur.

Prosedur pembangkitan QR Code dari sebuah teks dapat dilihat pada gambar 2.7 [27].



Gambar 2.7 Alur Pembangkitan QR-Code

Langkah-langkah untuk untuk membaca QR Code menjadi teks aslinya merupakan reverse atau kebalikan dari langkah-langkah pada pembangkitan QR Code. Secara umum prosedur pembacaan QR Code dapat dilihat pada gambar 2.8 [27].



Gambar 2.8 Alur Pembacaan QR-Code

Adapun spesifikasi dari QR-Code dapat dilihat pada gambar 2.8 [25].

Symbol Size	Min. 21x21 cell – Max 177x177 cell (with 4 cell interval)	
Information Type and Volume	Numerical Character	7,089 characters at maximum
	Alphabeth sign	4,296 characters at maximum
	Binary (8 bit)	2,953 characters at maximum
	Kanji character	1,817 characters at maximum
Conversoin efficiency	Numerical character mode	3,3 cells/characters
	Alphanumerical/signs mode	5,5 cells/characters
	Binary (8 bit) mode	8 cells/characters
	Kanji character mode (13 bit)	13 cells/characters
Error correction funtionality	Level L	Approx 7% of the symbols area restored at maximum
	Level M	Approx 15% of the symbols area restored at maximum
	Level Q	Approx 25% of the symbols area restored at maximum
	Level H	Approx 30% of the symbols area restored at maximum
Linking funtionality	Possible to be divided into 16 symbols at maximum	

Gambar 2.9 Spesifikasi QR-Code

Berikut penjelasan dari spesifikasi QR-Code pada gambar 2.8 adalah sebagai berikut :

- a. *Symbol Size*, Kode QR dapat dipilih ukurannya secara bebas sesuai dengan volume data yang akan disimpan dan metode membaca. Ukuran simbol bertambah oleh empat sel dalam arah vertikal dan horizontal - 21x21 sel, 25x25 sel, 29x29 sel, dan ada 40 jenis ukuran dengan ukuran maksimum yang diatur ke 177x177 sel. Misalnya, dalam kasus untuk 45x45 sel, jika satu sel persegi berukuran 0,25 mm, satu sisi simbol akan menjadi $45 \times 0,25 \text{ mm} = 11,25 \text{ mm}$. Zona hening perlu ditambahkan pada kedua sisi simbol yang ukuran minimumnya adalah empat sel, dan oleh karena itu, ruang yang diperlukan untuk mencetak simbol ini adalah kuadrat $(4 + 45 + 4) \times 0,25 \text{ mm}$ yaitu 13,25 mm.
- b. *Information Type and Volume* Kode QR dapat menangani berbagai jenis data seperti karakter numerik, huruf, tanda, karakter Kanji, Hiragana, Katakana, tanda kontrol, dan gambar. Pada dasarnya dapat memiliki set karakter yang didukung oleh ISO / IEC 646 dan ISO / IEC 10646.
- c. *Data Conversion Efficiency* QR Code memiliki empat jenis mode konversi - karakter numerik, alfanumerik / tanda, biner, dan karakter Kanji - untuk penyandian data. Setiap mode memiliki pertimbangan untuk meningkatkan efisiensi konversinya.
- d. *Error Correction Functionality* QR Code memiliki fungsi koreksi kesalahan untuk mengembalikan data. Ada empat level restorasi berbeda sehingga Anda dapat memilih level yang cocok dengan setiap lingkungan penggunaan.

2.2.4 One-Time Password

One-time password merupakan teknik autentikasi yang hanya berlaku untuk satu kali penggunaan[8]. OTP menjamin keamanan terhadap serangan replay, dimana penyerang yang berhasil mendapatkan nilai OTP yang sudah digunakan tidak mungkin dapat digunakan kembali.

OTP menggunakan algoritma pembangkitan dengan sifat semu acak yang telah diuji secara kriptografi[8]. Berikut merupakan metode-metode yang digunakan pada pembangkitan OTP :

1. Berdasarkan sinkronisasi waktu, metode seperti ini digunakan untuk otentikasi antara client dan server karena hanya berlaku untuk waktu yang singkat.
2. Berdasarkan suatu algoritma matematika yang menggunakan masukkan nilai OTP sebelumnya untuk mendapatkan nilai OTP baru.
3. Berdasarkan suatu nilai challenge dimana OTP dihasilkan dari suatu algoritma matematika yang mengkombinasikan pengetahuan terhadap nilai rahasia challenge (contoh metode seperti ini diterapkan pada otentikasi untuk suatu transaksi atau counter).

Kombinasi password OTP hanya dapat digunakan sekali. Ada dua cara utama untuk memperoleh One-Time Password, diantaranya adalah :

1. Token hardware: misalnya perangkat token, yang dapat Anda plug ke port USB Anda dan akan secara otomatis ketik kode OTP untuk Anda.
2. Token perangkat lunak: seperti Google Authenticator, dalam hal ini aplikasi Android sederhana menampilkan Anda kode OTP yang dapat Anda masukkan pada form login Anda.

Password OTP dibuat secara acak atau semi-acak. Hal tersebut penyerang sulit untuk menebak nilai yang dihasilkan. Selain itu OTP juga menggunakan fungsi hash yang dapat digunakan untuk menghasilkan nilai tetapi sulit untuk ditebak dan dibalikkan.

1. HOTP (*Hmac-based One-Time Password algorithm*)

Algoritma HOTP (*Hmac-based One-Time Password algorithm*) bergantung kepada dua hal: kunci rahasia bersama dan faktor bergerak (*moving factor*). Kunci rahasia bersama adalah sebuah string yang diketahui oleh kedua belah pihak, sedangkan faktor bergerak adalah sebuah bilangan penghitung yang selalu bertambah apabila sebuah transaksi dilakukan [28].

Algoritma ini bersifat event-based, artinya setiap kali OTP baru dihasilkan, faktor bergerak akan bertambah, maka password kemudian dihasilkan harus berbeda setiap kali.

Algoritma ini menggunakan algoritma HMAC-SHA-1. HMAC adalah salah satu tipe dari Message Authentication Code (MAC) yang berbasis fungsi kriptografi hash satu arah. MAC sendiri adalah fungsi satu-arah yang menggunakan kunci privat dalam pembangkitan nilai hash. Seperti fungsi MAC yang lain, HMAC ini digunakan untuk memeriksa integritas data dan autentifikasi dari sebuah pesan yang dikirimkan, tapi di lain pihak tidak mendukung prinsip non-repudiation. Hal ini dikarenakan pemilikan kunci privat yang bisa diketahui oleh beberapa orang, sehingga tidak memungkinkan untuk mengetahui orang yang membangkitkan nilai hash yang sebenarnya.

Adapun bentuk umum dari HOTP adalah :

$$\text{HOTP}(K, C) = \text{Truncate}(\text{HMAC-SHA-1}(K, C))$$

Dimana Truncate merupakan fungsi untuk mengkonversi nilai HMAC-SHA-1 menjadi nilai HOTP dan K adalah Kunci rahasia bersama dan C adalah faktor bergerak. Nilai HOTP akan diperlakukan sebagai big-endian. Berikut ini adalah contoh ekstraksi dari the HMAC-SHA-1 berupa array byte.

```
int offset = hmac_result[19] & 0xf ;
int bin_code = (hmac_result[offset] & 0x7f) << 24
               | (hmac_result[offset+1] & 0xff) << 16
               | (hmac_result[offset+2] & 0xff) << 8
               | (hmac_result[offset+3] & 0xff) ;
```

Byte Number
00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19
Byte Value
1f 86 98 69 0e 02 ca 16 61 85 50 ef 7f 19 da 8e 94 5b 55 5a

2. *TOTP (Time-based One-Time Password algorithm)*

Algoritma TOTP mirip dengan HOTP. Algoritma ini juga bergantung kepada kunci rahasia bersama dan faktor bergerak (moving factor), tetapi proses generate moving factor berbeda [11]. Pada TOTP moving factor akan terus berganti tergantung pada waktu generate. Proses penghitungannya sama dengan HOTP. Adapun bentuk umum dari TOTP adalah :

$$\text{TOTP} = \text{HOTP} (K, T)$$

Dimana K adalah Kunci rahasia bersama dan T adalah sebuah nilai integer yang merepresentasikan waktu. T dihitung menggunakan fungsi:

$$T = \frac{T_{\text{current}} - T_0}{x}$$

T current adalah waktu saat ini dalam detik sedangkan T0 disepakati pada saat inisialisasi awal, biasanya nilainya diset nol (T0=0). Parameter X menentukan berapa lama sebuah TOTP valid. Standar yang digunakan adalah 30. Jadi dengan mengisi X=30, kita mengizinkan sebuah OTP hanya valid selama 30 detik

2.2.5 *Firebase*

Firebase adalah sebuah layanan *Cloud Service Provider* dan BaaS (*Backend as a Service*) yang saat ini dimiliki Google. *Firebase* berevolusi dari *Envolve* yang merupakan sebuah startup yang didirikan oleh James Tamplin dan Andrew Lee pada 2011. *Envolve* menyediakan pengembang sebuah API yang memungkinkan integrasi fungsi obrolan online ke situs web mereka. Pengembang menggunakan *Envolve* untuk menyinkronkan data aplikasi seperti keadaan game secara real time di seluruh penggunanya. Tamplin dan Lee memutuskan untuk memisahkan sistem obrolan dan arsitektur *Realtime* yang mendukungnya. Mereka mendirikan *Firebase* sebagai perusahaan terpisah pada April 2012. Pada Oktober 2014 *Firebase* diakuisi oleh Google, lalu di tahun berikutnya Google mengakuisi *Divshot* untuk menggabungkannya dengan tim *Firebase*. Setelah diakuisi oleh

Google, Firebase memperluas layanan untuk menjadi platform terpadu bagi para pengembang seluler.

Pada Januari 2017, Google mengakuisisi Fabric dan Crashlytics dari Twitter untuk bergabung dengan layanan tersebut ke tim Firebase. Adapun beberapa fitur yang ada pada Firebase diantaranya adalah [29]:

1. Remote Config.

Merupakan fitur yang memungkinkan untuk melakukan perubahan konfigurasi pada aplikasi Android / iOS tanpa harus melakukan pembaruan aplikasi di Play Store / App Store.

2. Cloud Messaging.

Fitur yang memungkinkan dapat mengirim dan menerima pesan lintas platform seperti Android, iOS dan Web.

3. Authentication.

Merupakan layanan sistem otentikasi yang menerapkan kode client-side yang memungkinkan pengguna dapat mendaftar dan login ke aplikasi Facebook ataupun Twitter. Pengembang juga dapat mengaktifkan otentikasi pengguna dengan login email dan kata sandi yang disimpan dengan Firebase.

4. Realtime Database.

Firebase menyediakan realtime database dan backend sebagai layanan. Layanan ini menyediakan pengembang antarmuka pemrograman aplikasi yang memungkinkan data aplikasi disinkronkan pada seluruh klien dan disimpan di Firebase cloud.

5. Storage.

Fitur yang dapat menyimpan dan menerima konten dari pengguna seperti gambar secara langsung dari klien firebase.

6. Hosting.

Hosting pada firebase berisi file static seperti CSS, HTML, JavaScript dan file lain yang tidak berubah secara dinamis (Seperti PHP). Layanan tersebut mengirim file lewat Content Delivery Network (CDN) melalui HTTPS dan SSL.

2.2.6 Java

Java merupakan bahasa pemrograman tingkat tinggi (high level) yang mudah dipahami oleh manusia mengingat kata-kata atau statement-nya menyerupai bahasa manusia (Bahasa Inggris), namun dalam penulisannya memerlukan aturan (syntax) yang ketat [30]. Java dibuat oleh perusahaan Sun Microsystems, oleh James Gosling, Patrick Naughton, dan Mike Sheridan pada 1991. Dalam perkembangannya Java diakuisisi oleh perusahaan Oracle.

Spesifikasi Bahasa Java dan Java API mendefinisikan standar baku Bahasa pemrograman Java. Spesifikasi Bahasa Java merupakan suatu definisi teknis yang membuat sintaks dan semantik Bahasa pemrograman Java. API (Application Program Interface) memuat kelas dan antarmuka yang telah didefinisikan untuk mengembangkan program Java. Spesifikasi Bahasa Java stabil tetapi API masih terus berkembang [31].

API adalah Application Programming Interface yaitu sebuah layer yang berisi kelas-kelas yang sudah didefinisikan dan antarmuka pemrograman yang membantu para pengembang aplikasi dalam perancangan membangun sebuah aplikasi [32]. API membuat para pengembang mampu untuk mengakses fungsi-fungsi sistem operasi yang diizinkan melalui Bahasa Java. Pada saat ini ada 3 buah API dari Java, yaitu :

1. J2SE (Java 2 Standar Edition).

Java 2 SE dapat digunakan untuk mengembangkan aplikasi-aplikasi yang bersifat client-side standalone atau *applet*.

2. J2EE (Java 2 Enterprise Edition).

Java 2 EE dapat digunakan untuk mengembangkan aplikasi-aplikasi pada sisi server, seperti Java *servlets* dan JavaServer Pages.

3. J2ME (Java 2 Micro Edition).

Java 2 ME adalah API yang menggunakan subset dari J2SE tetapi memiliki kegunaan untuk mengembangkan aplikasi pada handled device seperti pada smartphone atau PDA yang didalamnya telah ditanam interpreter Java.

Java mempunyai konsep OOP (*Object Oriented Programming*) sehingga dalam mengembangkan sebuah aplikasi menjadi lebih mudah serta dapat menjaga sistem tetap modular, fleksibel, dan ekstensibel. Java memiliki kemampuan untuk menangani kesalahan dalam penulisan kode atau *syntax*, khususnya pada *error checking* atau *error runtime* seperti pada *Exception Handling*.

Bahasa pemrograman Java mempunyai karakteristik yang diantaranya adalah [33] :

1. Sederhana

Bahasa pemrograman Java hampir sama seperti Bahasa pemrograman C++ tetapi ada beberapa fitur yang dihilangkan.

2. Berorientasi Objek

Pemrograman berorientasi objek berdasarkan pemodelan dunia nyata yang disebut objek yang terdiri dari data dan perilaku yang dikemas dalam objek.

3. Terdistribusi

Bahasa pemrograman Java memiliki banyak pustaka kelas untuk berkomunikasi menggunakan protokol internet TCP/IP, termasuk protokol seperti HTTP dan FTP.

4. Interpreter

Kompiler Java menerjemahkan file sumber Java menjadi byte kode yang dapat dijalankan pada semua komputer yang menjalankan interpreter Java atau browser.

5. Kokoh

Pemrograman pada umumnya mempunyai Batasan termasuk tipe data dan penggunaan pointer. Java tidak mendukung aritmatika pointer melainkan array.

6. Aman

Java membatasi pada pointer sehingga developer tidak dapat mengakses memori jika tidak diizinkan.

7. Arsitektur Netral

Kompiler Java membuat *bytecode* yang dikirim ke browser dan diterjemahkan oleh browser yang memiliki interpreter Java.

8. *Performance* Tinggi

Kompilasi Java termasuk kebebasan menerjemahkan bytecode kode mesin pada platform yang ada. Hal ini memberikan efisiensi yang sama seperti kompilasi dan pemrosesan secara tradisional.

9. *Multithreaded*

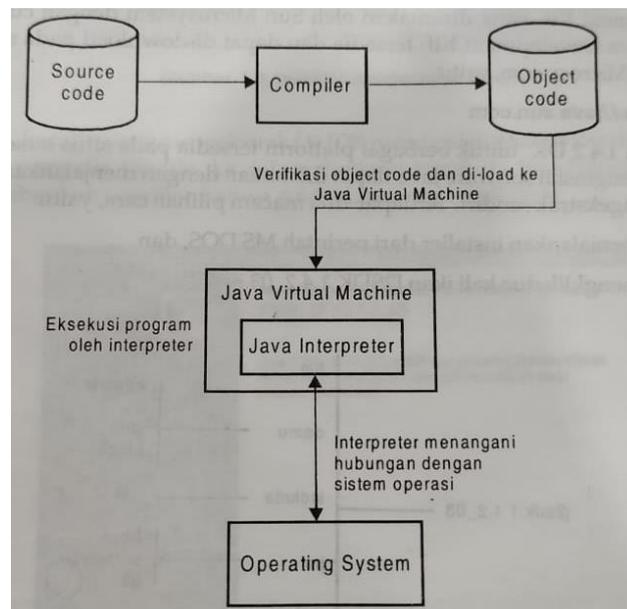
Java adalah Bahasa pemrograman yang dapat digunakan untuk membuat aplikasi dimana beberapa event berlangsung pada saat yang sama.

10. Dinamis

Java menggunakan metode interface agar tidak memerlukan kompilasi kembali jika suatu kelas diubah. Metode dan variabel dari instance baru dapat digunakan tanpa mempengaruhi objek pada *client*.

1. Lingkungan Java

Program Java dapat dijalankan pada berbagai macam sistem operasi yang berbeda-beda seperti Windows 95/98/NT/2000, Linux dan Sun Solaris. Kemampuan tersebut dikarenakan eksekusi program Java dilakukan tidak pada komputer yang digunakan programmer. Kompiler Java mengkonversi kode sumber menjadi program biner yang terdiri dari *bytecode*. *Bytecode* adalah intruksi mesin untuk Java *Virtual Machine*. Berikut adalah alur pemrograman Java dapat dilihat pada gambar 2.9 [33].



Gambar 2.10 Alur Perogramman Java

2.2.7 *Object Oriented Programming*

Object Oriented Programming atau metode berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek dan pengujian berorientasi objek [34].

Ada tiga konsep pemrograman berbasis objek, yaitu diantaranya [35] :

1. Enkapsulasi

Enkapsulasi adalah suatu mekanisme untuk menyembunyikan atau memproteksi suatu proses dari kemungkinan penyalahgunaan dari luar sistem sekaligus menyederhanakan penggunaan sistem itu sendiri.

2. Pewarisan

Pewarisan merupakan penurunan adalah konsep pemrograman dimana sebuah kelas dapat menurunkan properti dan method yang dimilikinya kepada kelas lain.

3. Polymorphism

Polymorphism berasal dari Bahasa Yunani yang berarti “banyak bentuk”. Dalam OOP, konsep ini memungkinkan digunakannya suatu interface yang sama untuk memerintah suatu objek agar melakukan suatu aksi atau tindakan yang mungkin akan memberikan hasil akhir yang serupa, tetapi melalui proses yang berbeda.

2.2.8 Android

Android merupakan Sistem Operasi berbasis Linux, yang bersifat terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri dan digunakan oleh berbagai macam perangkat mobile. Pada saat perilisannya pada tanggal 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Google merilis kode-kode Android dibawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler.

1. Arsitektur Android

Adapun arsitektur Android secara garis besar adalah sebagai berikut [36] :

1. *Applications dan Widgets*, merupakan layer dimana kita berhubungan dengan aplikasi. Pada layer ini terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak dan lainnya. Semua aplikasi ditulis menggunakan Bahasa pemrograman Java.
2. *Applications Framework*, merupakan layer dimana para pembuat aplikasi melakukan pengembangan aplikasi yang akan dijalankan pada sistem operasi android, karena karena pada layer inilah aplikasi dapat dirancang dan dibuat, seperti *content – providers* yang berupa sms dan panggilan telepon. Komponen – komponen yang termasuk di dalam

layer ini adalah *Views*, *Content Provider*, *Resource Manager*, *Notification Manager*, dan *Activity Manager*.

3. *Libraries*, merupakan layer di mana fitur – fitur android berada, biasanya para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan diatas kernel, layer ini meliputi berbagai library C / C++ seperti Libe dan SSL, serta:
 - a. *Libraries* media untuk pemutaran media audio dan video.
 - b. *Libraries* untuk manajemen tampilan.
 - c. *Libraries Graphics* mencakup SGL dan OpenGL untuk grafis 2D dan 3D.
 - d. *Libraries SQLite* untuk dukungan *database*.
 - e. *Libraries SSL* dan *WebKit* terintegrasi dengan *web browser* dan *security*.
 - f. *Libraries LiveWebcore* mencakup modern *web browser* dengan *engine embeded web view*.
 - g. *Libraries 3D* yang mencakup implementasi OpenGL ES 1.0 API's.
4. *Android Runtime*, merupakan layer yang membuat aplikasi Android dapat dijalankan di mana dalam prosesnya menggunakan implementasi linux. *Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi Android. Dalam *Android Runtime*, dibagi menjadi dua bagian yaitu :
 - a. *Core Libraries*, aplikasi android dibangun dalam Bahasa java, sementara Dalvik sebagai virtual mesinnya bukan virtual machine java, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menterjemahkan Bahasa java / c yang ditangani oleh *Core Libraries*.
 - b. *Dalvik Virtual Machine*, Virtual Mesin berbasis regiter yang dioptimalkan untuk menjalankan fungsi – fungsi secara efisien, di mana merupakan pengembangan yang mampu membuat linux kernel untuk melakukan *threading* dan manajemen tingkat rendah.

5. *Linux Kernel*, merupakan layer di mana inti dari operating system dari android itu berada. Berisi *file-file system* yang mengatur sistem *processing, memory, resource, drivers*, dan sistem – sistem operasi android lainnya. Linux Kernel yang digunakan android adalah linux kernel release 2.6.

2.2.9 Library ZXing (Zebra Crossing)

QR Code merupakan kode 2 dimensi yang digunakan untuk menyimpan informasi. Pembuatan dan pemindaian QR Code pada aplikasi android di dalam sistem ini dilakukan menggunakan library ZXing (Zebra Crossing). ZXing merupakan sebuah library open-source pemrosesan gambar barcode multi-format 1D/2D yang diimplementasikan menggunakan bahasa pemrograman java.

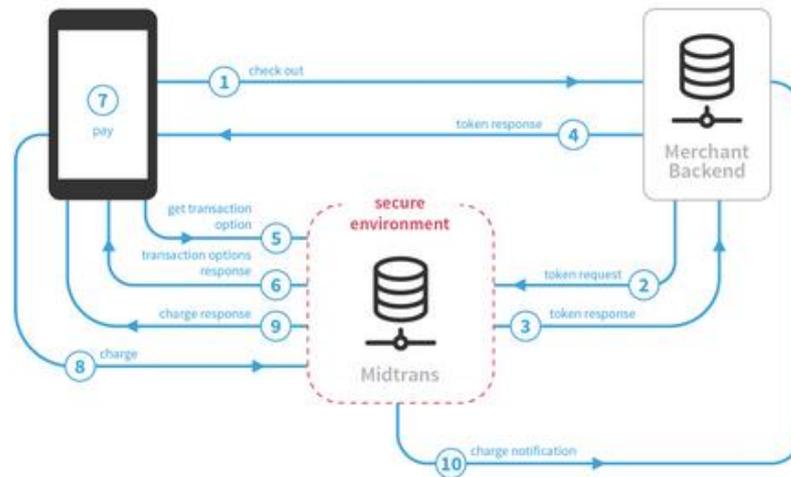
Untuk mengintegrasikan library ZXing dalam suatu project aplikasi android adalah dengan cara menambahkan library kedalam gradle. Cara untuk menambahkan library ZXing kedalam gradle dapat dilihat pada gambar 2.10 berikut.

```
dependencies {
    implementation 'com.journeyapps:zxing-android-embedded:3.6.0' //Library QR-Code
}
```

Gambar 2.11 Integrasi Library ZXing

2.2.10 Midtrans Payment Gateway

Midtrans berdiri pada 2012 dengan nama Veritrans untuk memfasilitasi bisnis online di Indonesia dengan sistem pembayaran yang terpercaya dan mudah untuk digunakan. Midtrans menyediakan 16 metode pembayaran dalam satu langkah integrasi mudah agar dapat menerima pembayaran dengan mudah dan lancar[37]. Termasuk metode pembayaran Go-Pay didalamnya, dimana ketika proses pembayaran dilakukan maka aplikasi android akan di-direct untuk pindah ke aplikasi Go-Jek. Alur dari proses transaksi yang terjadi pada midtrans dapat dilihat pada gambar 2.11 [38].



Gambar 2.12 Alur Transaksi Midtrans

Berikut penjelasan dari gambar alur transaksi midtrans :

1. Checkout: Pelanggan mengklik tombol Checkout pada aplikasi Host dan aplikasi membuat permintaan ke Merchant Server.
2. Token request: Merchant Server membuat permintaan ke server Midtrans dengan Informasi Pemesanan.
3. Token response: Midtrans merespons dengan token transaksi yang valid ke server Merchant.
4. Token response: Server pedagang meneruskan token ke Mobile SDK.
5. Get transAction options: SDK Seluler meminta pembayaran / informasi pedagang berdasarkan token.
6. Transaction options response: SDK Seluler membuat opsi pembayaran dan informasi pembayaran untuk melakukan pembayaran.
7. Pay: Pelanggan memilih metode pembayaran dan rincian pembayaran dan mengklik "Bayar".
8. Charge: SDK Seluler mengirimkan permintaan Biaya ke Midtrans Backend untuk Pemrosesan pembayaran.
9. Charge response: SDK Seluler menerima respons dari Midtrans Backend dan memicu pengendali pada Aplikasi Seluler dengan status berhasil / gagal / tertunda.
10. Charge notification : Midtrans Backend mengirimkan pemberitahuan ke Merchant backend yang mengkonfirmasi penyelesaian transaksi.

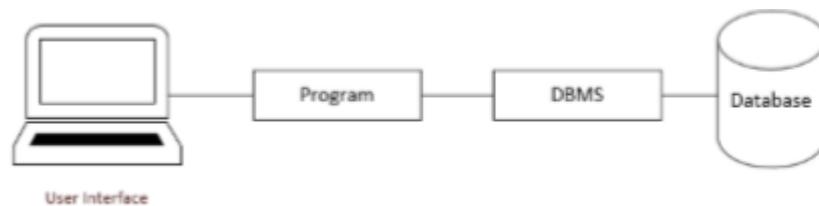
2.2.11 Basis Data

Basis Data terdiri dari dua kata, yaitu Basis dan Data. Basis dapat diartikan sebagai markas atau gudang, tempat bersarang/berkumpul. Sedangkan Data adalah representasi fakta dunia nyata yang mewakili sesuatu objek seperti manusia (pegawai, siswa, pembeli pelanggan), barang, hewan, peristiwa, konsep, keadaan, dan sebagainya, yang diwujudkan dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya [39].

Sebagai satu kesatuan istilah, basis data dapat diartikan didefinisikan sebagai kumpulan data yang terintegrasi dan diatur sedemikian rupa sehingga data tersebut dapat dimanipulasi, diambil, dan dicari secara cepat [40].

1. Database Management Sistem (DBMS)

Database berbeda dengan Database Management Sistem (DBMS). DBMS adalah kumpulan program yang digunakan untuk mendefinisikan, mengatur, dan memproses database; sedangkan database itu sendiri esensinya adalah sebuah struktur yang dibangun untuk keperluan penyimpanan data. DBMS alat yang berperan untuk membangun struktur tersebut [40]. Jadi dapat diartikan bahwa DBMS merupakan perantara antara user dengan database. Peranan DBMS pada suatu sistem dapat dilihat pada gambar 2.12.



Gambar 2.13 Peranan DMBS

2. Bahasa Basis Data

Cara berinteraksi antara pemakai dengan basis data diatur dalam suatu bahasa khusus yang ditetapkan oleh perusahaan pembuat DBMS. Bahasa itu dapat kita sebut sebagai Bahasa Basis Data yang terdiri atas

sejumlah perintah (statement) yang diformulasikan dan dapat diberikan user dan dikenali/diproses oleh DBMS untuk melakukan suatu aksi tertentu. Bahasa Basis Data dapat dibedakan kedalam dua bentuk yaitu Data Definition Language (DDL) dan Data Manipulation Language (DML).

a. Data Definition Language (DDL).

DDL merupakan struktur basis data yang menggambarkan skema basis data secara keseluruhan dan didesain dengan bahasa khusus.

b. Data Manipulation Language (DML).

DML merupakan bentuk Bahasa Basis Data yang berguna untuk melakukan manipulasi dan pengambilan data pada suatu basis data.

2.2.12 Unified Modeling Language (UML)

UML (*Unified Modeling Language*) merupakan sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak. UML mempunyai sebuah standar untuk merancang model sebuah sistem. Dengan memanfaatkan UML kita dapat membuat model untuk semua jenis perangkat lunak, dimana aplikasi tersebut dapat berjalan pada perangkat keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. UML lebih cocok digunakan untuk pengembangan perangkat lunak yang menggunakan object oriented, seperti: C++, Java, C#, atau VB.NET, karena UML merupakan bahasa yang menggunakan class dan operation dalam konsep dasarnya. Walaupun demikian, UML tetapi dapat digunakan untuk memodelkan aplikasi procedural dalam VB atau C.

UML mendefinisikan notasi dan syntax/semantic. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram perangkat lunak. Setiap bentuk memiliki makna tertentu, dan UML syntax mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari tiga notasi yang telah ada sebelumnya: Grady Booch OOD (Object-Oriented Design), Jim

Rumbaugh OMT (Object Modeling Technique), dan Ivar Jacobson (Object-Oriented Software Engineering) [40].

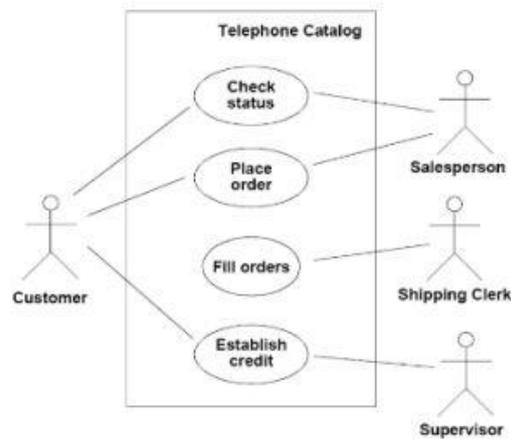
UML menyediakan 10 (sepuluh) macam diagram untuk memodelkan aplikasi berorientasi object, yaitu:

1. *Use Case Diagram* berfungsi untuk memodelkan proses bisnis.
2. *Conceptual Diagram* berfungsi untuk memodelkan konsep-konsep yang ada di dalam aplikasi.
3. *Sequence Diagram* untuk memodelkan pengiriman pesan (*message*) antar objek.
4. *Collaboration Diagram* untuk memodelkan interaksi antar objek.
5. *State Diagram* untuk memodelkan perilaku objek di dalam sistem.
6. *Activity Diagram* untuk memodelkan perilaku user dan objek di dalam sistem.
7. *Class Diagram* untuk memodelkan struktur kelas.
8. *Object Diagram* untuk memodelkan struktur objek.
9. *Component Diagram* untuk memodelkan komponen objek.
10. *Deployment Diagram* untuk memodelkan distribusi aplikasi.

Dari 10 (sepuluh) macam diagram yang disediakan UML, terdapat 3 (tiga) diagram yang paling sering digunakan dalam pembangunan aplikasi berorientasi objek, yaitu *use case diagram*, *activity diagram* dan *class diagram*.

1. Use Case Diagram

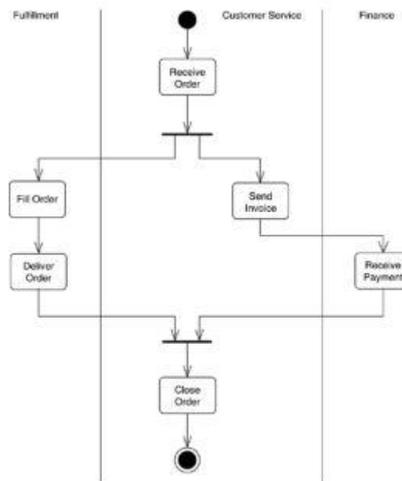
Use Case Diagram adalah titik awal yang sangat baik untuk hampir setiap aspek pembangunan sistem berorientasi objek, desain, pengujian, dan dokumentasi. *Use Case Diagram* menggambarkan sistem persyaratan ketat dari luar ke dalam, dan juga menentukan nilai yang sistem memberikan kepada pengguna[41]. Contoh *use case diagram* dapat dilihat pada gambar 2.13.



Gambar 2.14 Use Case Diagram

2. Activity Diagram

Activity Diagram sangat baik jika digunakan pada pemodelan proses bisnis. *Activity Diagram* sangat membantu untuk mengkoordinasi kita hingga mencapai tujuan bisnis [41]. Contoh *activity diagram* dapat dilihat pada gambar 2.14.



Gambar 2.15 Activity Diagram

3. Class Diagram

Class Diagram merupakan diagram yang selalu ada di permodelan berorientasi objek, karena class diagram menunjukkan hubungan antar

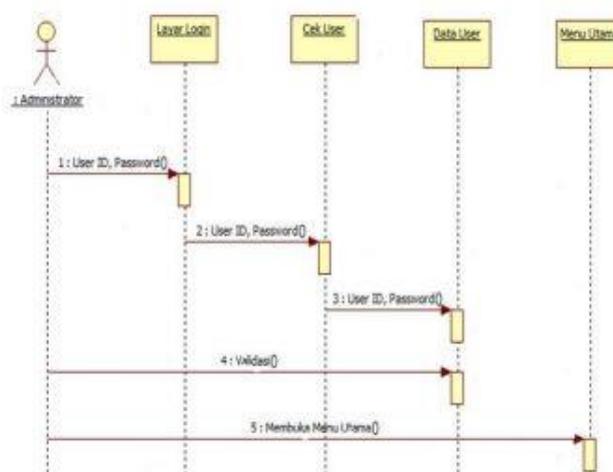
class dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan [41]. Contoh *class diagram* dapat dilihat pada gambar 2.15.



Gambar 2.16 Class Diagram

4. Sequence Diagram

Sequence Diagram adalah diagram yang menggambarkan interaksi antar objek. *Sequence Diagram* secara khusus menjabarkan *behavior* sebuah scenario tunggal. Diagram tersebut menunjukkan sejumlah objek contoh dan pesan – pesan yang melewati objek ini dalam sebuah *use case* [41]. Contoh *sequence diagram* dapat dilihat pada gambar 2.16.



Gambar 2.17 Sequence Diagram

2.2.13 Pengujian Alpha

Pengujian Alpha dilakukan untuk memastikan bahwa aplikasi dapat berjalan dengan benar sesuai dengan kebutuhan dan tujuan yang diharapkan. Pengujian yang digunakan pada aplikasi ini ada 2 (dua), yaitu teknik pengujian white box testing dan black box testing.

Pengujian black-box berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian black-box memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk program. Pengujian black-box berusaha menemukan kesalahan dalam kategori sebagai berikut.

1. Fungsi-fungsi yang tidak benar atau hilang,
2. Kesalahan interface,
3. Kesalahan dalam struktur data atau akses database eksternal,
4. Kesalahan kinerja,
5. Inisialisasi dan kesalahan terminasi.

Pengujian alpha berfokus pada persyaratan fungsional perangkat lunak. Fungsi-fungsi yang diuji meliputi tombol-tombol menu dengan kasus yang benar dan kasus yang salah.

2.2.14 Pengujian Beta

Pengujian Beta dilakukan secara langsung terhadap pengguna dengan menggunakan kuesioner mengenai kepuasan pengguna atas aplikasi yang telah dibangun. Adapun metode penilaian pengujian yang digunakan adalah metode kuantitatif berdasarkan data dari pengguna. Kuesioner disebarakan menggunakan teknik Simple Random Sampling dimana anggota sample dari populasi dilakukan secara acak tanpa memperhatikan strata yang ada dalam populasi.

2.2.15 Android Studio

Android Studio adalah Lingkungan Pengembangan Terpadu (*Integrated Development Environment/IDE*) resmi untuk pengembangan aplikasi Android, yang didasarkan pada IntelliJ IDEA yang dikembangkan oleh

Jetbrains. Selain sebagai editor kode dan fitur developer IntelliJ yang andal, Android Studio menawarkan banyak fitur yang meningkatkan produktivitas Anda dalam membuat aplikasi Android, diantaranya [42] :

1. Sistem build berbasis *Gradle* yang fleksibel.
2. Emulator yang cepat dan kaya fitur.
3. Lingkungan terpadu tempat Anda bisa mengembangkan aplikasi untuk semua perangkat Android.
4. Terapkan Perubahan untuk melakukan push pada perubahan kode dan resource ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi.
5. Template kode dan integrasi *GitHub* untuk membantu Anda membuat fitur aplikasi umum dan mengimpor kode sampel.
6. Framework dan fitur pengujian yang lengkap.
7. Fitur lint untuk merekam performa, kegunaan, kompatibilitas versi, dan masalah lainnya.
8. Dukungan C++ dan NDK.
9. Dukungan bawaan untuk *Google Cloud Platform*, yang memudahkan integrasi *Google Cloud Messaging* dan *App Engine*.

2.2.16 CodeIgniter

CodeIgniter adalah sebuah Application Development Framework (toolkit) berlisensikan dibawah lisensi MIT untuk membangun website menggunakan PHP secara gratis. Tujuannya adalah untuk memungkinkan pengembangan proyek-proyek lebih cepat daripada menulis kode dari awal, tersedia banyak library untuk tugas-tugas yang biasa diperlukan, serta antarmuka dan struktur logis yang sederhana untuk mengakses library ini. CodeIgniter memungkinkan pembangunan website dengan meminimalkan jumlah kode yang dibutuhkan untuk tugas yang diberikan [43]. CodeIgniter ditulis atau dibuat oleh Ellis Lab dan dirilis pertamakali pada 28 Februari 2006.

CodeIgniter menggunakan pendekatan *Model-View-Controller*, yang memungkinkan pemisahan antara logika dan presentasi. Berikut adalah penjelasan mengenai bagian dari *Model-View-Controller* [44].

- a. **Model** mewakili struktur data. Biasanya *class* model akan berisi fungsi yang membantu untuk mengambil, menyimpan, dan memperbarui informasi dalam database.
- b. **View** adalah informasi yang disajikan kepada pengguna. *View* yang biasanya akan menjadi halaman web, tetapi dalam CodeIgniter, *view* juga bisa menjadi bagian dari sebuah halaman seperti header atau footer. Hal ini juga dapat menjadi halaman *RSS*, atau jenis-jenis lain dari “halaman”.
- c. **Controller** berfungsi sebagai perantara antara *Model*, *View*, dan *resource* lain yang diperlukan untuk memproses *HTTP request* dan menghasilkan halaman web.

2.2.17 HTML

HTML (*HyperText Markup Language*) digunakan untuk membuat struktur atau membuat desain halaman website [45]. Disebut *HyperText* karena didalam script HTML dapat memungkinkan membuat sebuah teks menjadi *link* yang dapat berpindah dari satu halaman ke halaman lainnya dengan mengklik *link* tersebut [46].

Pada dasarnya struktur HTML terdiri dari dua bagian, yaitu bagian *head* dan *body*. Bagian *head* diawali dengan tag `<head>` dan diakhiri dengan tag `</head>`. Bagian *body* diawali dengan tag `<body>` dan diakhiri dengan tag `</body>`. Kedua bagian tersebut disatukan dengan tag `<html>` dan `</html>` [46].

```

<!DOCTYPE html>
<html>
  <head>
    <title>Mengenal Paragraf</title>
  </head>
  <body>
    Ayo belajar HTML (paragraf 1)
    <br />
    Bersama Jubilee (paragraf 2)
  </body>
</html>

```

Gambar 2.18 Struktur HTML

2.2.18 CSS

CSS (*Cascading Style Sheet*) adalah suatu Bahasa *stylesheet* yang digunakan untuk mengatur tampilan suatu website, baik tata letaknya, jenis huruf, warna dan semua yang berhubungan dengan tampilan. Pada umumnya CSS digunakan untuk membuat format halaman web yang ditulis dengan HTML atau XHTML [47].

Ada dua acara yang bias diterapkan untuk menggunakan CSS pada web. Yang pertama adalah dengan membuat CSS langsung didalam satu file HTML (*internal/inline style sheet*). Cara yang kedua dengan memanggil CSS tersebut dari file CSS sendiri (*external style sheet*) [47].

```

<head>
<link rel="stylesheet" type="text/css"
href="mystyle.css" />
</head>

```

Gambar 2.19 Contoh Eksternal CSS

```

<head>
<style type="text/css">
hr {color:sienna;}
p {margin-left:20px;}
body {background-image:url("images/back40.gif");}
</style>
</head>

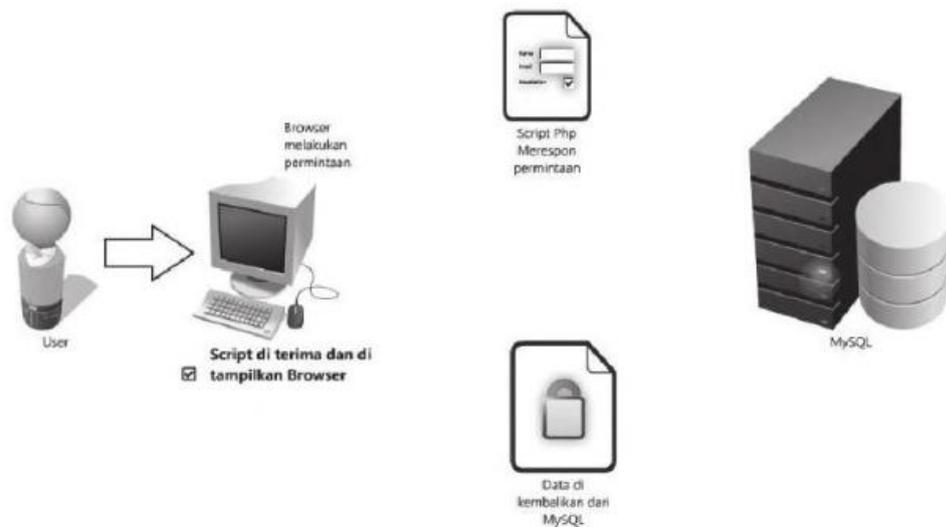
```

Gambar 2.20 Contoh Nama File Internal css: css1.html

2.2.19 PHP

PHP *Hypertext Preprocessor*, awalnya PHP merupakan singkatan dari *Personal Home Page* yang digunakan untuk membuat website pribadi yang dinamis dan interaktif. Dinamis artinya website tersebut bias berubah-ubah tampilan dan kontennya sesuai kondisi tertentu. Sedangkan interaktif berarti PHP dapat memberi *feedback* bagi penggunanya, misalnya menampilkan hasil pencarian produk [45].

Secara teori, cara kerja PHP berjalan dari database dalam server kemudian menyalurkannya kedalam *script* PHP untuk kemudian ditampilkan kedalam browser agar dapat dibaca oleh pengguna [48]. Berikut merupakan cara kerja dari PHP pada gambar 2.19.



Gambar 2.21 Cara Kerja PHP

2.2.20 MySQL

MySQL merupakan server yang melayani database. Untuk membuat dan mengolah database, dapat menggunakan Bahasa pemrograman yang disebut dengan query (perintah) SQL [45]. SQL (Structured Query Language) adalah sebuah Bahasa yang digunakan untuk mengakses data dalam basis data relasional, sedangkan MySQL query merupakan perintah atau instruksi yang dapat digunakan untuk mengelola database atau tabel dalam database MySQL.

MySQL adalah RDBMS yang cepat dan mudah digunakan, serta banyak digunakan untuk berbagai kebutuhan. MySQL dikembangkan oleh MySQL AB Swedia, adapun beberapa hal yang membuat MySQL banyak digunakan antara lain [49]:

- a. Berlisensi *open-source* dengan kata lain dapat digunakan secara gratis.
- b. Merupakan program yang *powerfull* dan menyediakan fitur yang lengkap.
- c. Menggunakan bentuk standar Bahasa data SQL.
- d. Dapat berkerja dengan banyak sistem operasi dan dengan banyak Bahasa pemrogramman.
- e. Bekerja dengan cepat dan baik, bahkan dengan banyak dataset.
- f. Sangat mudah digunakan dengan PHP untuk pengembanga website.
- g. Mendukung banyak database, sampai 50 juta baris atau lebih dalam satu tabel.
- h. Dapat dikustomisasi sesuai dengan keinginan pengguna.

2.2.21 JavaScript

JavScritpt adalah Bahasa script berdasar pada objek yang memperbolehkan pengguna untuk mengendalikan banyak aspek interaksi pemakai pada suatu dokumen HTML. Dimana objek tersebut dapat berupa suatu window, frame, URL, dokumen, form, button, atau item lainnya yang mempunyai properti yang saling berhubungan dengannya [47].

Kode program JavaScript dapat dituliskan langsung pada file HTML dengan menggunakan tag container yang diawali dengan tag `<SCRIPT>` dan diakhiri dengan tag `</SCRIPT>` dengan atribut yang harus diisi yaitu Language, hal ini dimaksudkan untuk memberitahu pada browser yang akan digunakan bahwa yang akan ditulis adalah JavaScript. Berikut adalah contoh penulisan script dapat dilihat pada gambar 2.15.

```
<SCRIPT LANGUAGE="JavaScript">  
    //program Anda masukkan di sini  
</SCRIPT>
```

Gambar 2.22 Penulisan JavaScript

2.2.22 Bootstrap

Bootstrap awalnya bernama Twitter Blueprint yang dikembangkan oleh tim desainer dan pengembang di Twitter yaitu Mark Otto dan Jacob Thornton pada tahun 2010, setelah digunakan sebagai framework internal pada situs Twitter, tahun berikutnya dirilis untuk public sebagai proyek open-source dengan menggunakan nama Bootstrap [50].

Bootstrap merupakan framework front-end yang berisi kumpulan tools gratis yang digunakan untuk membuat layout atau tampilan untuk aplikasi berbasis website yang fleksibel dan responsive serta memiliki komponen interface yang menarik. Bootstrap menggunakan HTML, CSS, dan JavaScript [51]. Adapun beberapa fitur dari bootstrap diantaranya adalah :

- a. Bootstrap menyediakan beberapa komponen yang siap pakai.
- b. Bootstrap kompatibel dengan beberapa browser seperti Mozilla Firefox, Google Chrome, dan yang lainnya.
- c. Telah mendukung metode *Responsive Web Design*, bahkan menggunakan paradigma *Mobile First* sebagai dasar pengembangan website yang *mobile-friendly*.
- d. Selain tersedia dalam bentuk CSS yang siap pakai, Bootstrap juga mendukung penggunaan *Sass* dan *Less* yang merupakan CSS *pre-processor* yang dipakai untuk mengkompilasi file CSS.
- e. Dilengkapi *library* JavaScript berbasis jQuery yang siap pakai.