

## BAB II

### TINJAUAN PUSTAKA DAN DASAR TEORI

#### 2.1 Tinjauan Pustaka

Dalam penelitian ini, penulis mengambil referensi tinjauan pustaka yang ditunjukkan pada table di bawah ini :

**Table 2.1 Perbandingan Penelitian**

Penulis	Judul	Bahasa Pemrograman	Hasil
Norhikmah, Azizah Rahma Safitri dan Laili Annas Sholikhhan (2016)	Penggunaan QR Code Dalam Presensi Berbasis Android	SQLite	Aplikasi presensi yang dapat memberikan rekapan data presensi dan dapat <i>export</i> data kedalam <i>excel</i> .
Nandang Hermanto, Nurfaizah dan Nur Rahmat Dwi Riyanto (2019)	Aplikasi Sistem Presensi Mahasiswa Berbasis Android	Java	Aplikasi presensi mahasiswa berbasis android yang dapat merekam data kehadiran mahasiswa dengan cepat dan dosen dapat mengetahui rekap kehadiran mahasiswa setelah selesai perkuliahan. Sistem dapat memberikan rekap kehadiran mahasiswa secara keseluruhan.
Erwin Susanto, Doan Perdana, Arif Indra Irawan, dan Rahmat Yasirandi (2019)	Pengembangan Sistem Presensi Menggunakan Quick Response Code Dinamis untuk Madrasah Aliyah Al Mukhlisin Bandung	Java dan Firebase	Aplikasi presensi yang dapat memberikan hasil rekapan data kehadiran siswa Madrasah Aliyah Al Mukhlisin Bandung dan <i>QR code</i> yang ditampilkan secara dinamik dan acak sesuai dengan nama dan nisn siswa.

Didi Juardi (2019)	Presensi Dan Reminder Menggunakan QR Code (Studi Kasus: SMA XXX)	Php dan MySQL	Sistem absensi dengan basis <i>web based</i> yang dapat mencatat presensi yang dilengkapi dengan SMS <i>gateway</i> yang akan dilaporkan kepada orang tua dan <i>reminder</i> pembayaran yang akan dikirimkan kepada orang tua siswa.
Mastari (2020)	Sistem presensi mahasiswa menggunakan <i>QR code</i> berbasis android pada STMIK Akakom Yogyakarta	Php dan Java	Memberikan informasi presensi mahasiswa untuk dosen dan mahasiswa serta laporan presensi yang dapat dikelola oleh bagian adak untuk proses rekapitulasi data presensi mahasiswa.
Alva Gani (2021)	Implementasi Qr Code Menggunakan Flutter Untuk Melakukan Presensi Mahasiswa Berbasis Mobile	Kotlin dan Dart	Aplikasi yang bisa digunakan untuk melakukan presensi, melihat rekap kehadiran dan melihat pemberitahuan kelas yang sudah dibuat oleh dosen.

Pada tahun 2016 Norhikmah, Azizah Rahma Safitri dan Laili Annas Sholikhah melakukan penelitian dengan judul “Penggunaan QR Code Dalam Presensi Berbasis Android”. Tujuan penelitian adalah untuk mempermudah dan mempercepat presensi maupun rekapitulasi presensi. Sistem yang digunakan sebelumnya masih secara manual, sehingga membutuhkan waktu yang cukup lama untuk melakukan presensi. Pada aplikasi ini tutor diwajibkan untuk login terlebih dahulu. Setelah itu, tutor memasukkan judul materi dan deskripsi materi pelatihan. Selanjutnya anggota AMCC bisa melakukan presensi. Hasil presensi akan tersimpan dalam SQLite, selain itu rekapan hasil presensi bisa diexport kedalam bentuk *excel*. Hasil penelitian ini adalah aplikasi dapat menerapkan *QR code* dalam

presensi pelatihan anggota AMCC sehingga lebih cepat dan efisien hanya dalam waktu 15 detik/*id card* dibandingkan dengan presensi manual (30 detik/*id card*).

Selanjutnya penelitian yang dilakukan oleh Nandang Hermanto, Nurfaizah dan Nur Rahmat Dwi Riyanto (2019) adalah membuat “Aplikasi Sistem Presensi Mahasiswa Berbasis Android”. Penelitian ini bertujuan untuk membuat rancangan sistem presensi mahasiswa menggunakan sistem operasi android dan pemanfaatan teknologi komunikasi yang sedang berkembang pesat yaitu *smartphone* dengan sistem operasi android. Aplikasi ini mengkombinasikan *QR code* dengan *International Mobile Station Equipment Identity* (IMEI). IMEI digunakan sebagai nomor unik dari handphone sehingga satu mahasiswa hanya memiliki satu akun yang digunakan untuk presensi. Dengan pengkombinasian *QR code* dan IMEI diharapkan dapat mempermudah dan mempercepat proses presensi pada saat perkuliahan serta dapat memberikan kemudahan kepada dosen dalam merekap kehadiran mahasiswa. Selain itu dengan pengkombinasian *QR code* dan IMEI diharapkan bisa untuk menghindari kecurangan pada proses presensi. Proses penggunaan sistem presensi ini adalah dosen menampilkan *QR code* dan mahasiswa akan *scanning QR code*. Mahasiswa bisa memilih menu lanjutan sesi presensi terakhir atau menu buka presensi baru, jika mahasiswa memilih lanjutan sesi presensi terakhir maka nomor pertemuan pada rekap kehadiran tidak berubah tetapi jika mahasiswa memilih buka sesi presensi baru maka nomor pertemuan pada rekap kehadiran akan bertambah satu dari sesi sebelumnya. Mahasiswa yang berhasil melakukan *scanning* tersebut akan ditampilkan pada halaman web. Rekap kehadiran mahasiswa berisi daftar hadir mahasiswa pada sesi terakhir dan jam

mahasiswa pada saat mahasiswa tersebut melakukan presensi. Hasil penelitian ini adalah sistem dapat memberikan informasi presensi mahasiswa untuk dosen dan mahasiswa serta rekap kehadiran mahasiswa.

Kemudian penelitian yang dilakukan oleh Erwin Susanto, Doan Perdana, Arif Indra Irawan, dan Rahmat Yasirandi (2019) dengan judul “Pengembangan Sistem Presensi Menggunakan *Quick Response Code Dinamis* untuk Madrasah Aliyah Al Mukhlisin Bandung”. Tujuan penelitian ini untuk mengembangkan sistem presensi otomatis dengan memanfaatkan teknologi *QR code*, sehingga proses presensi lebih efektif jika dibandingkan dengan presensi manual karena memanfaatkan teknologi digital. Sistem yang dibangun berbentuk aplikasi yang di-*install* pada *smartphone* guru dan siswa. Siswa yang hadir dapat melakukan presensi sebelum pelajaran dimulai. Pada aplikasi ini menggunakan citra *QR code* yang dibangkitkan secara dinamis (berubah-ubah) oleh sistem setiap kali pengguna login, hal ini dapat menghindari kecurangan siswa yang titip presensi. Kualitas layanan *QoS* pada sistem ini sudah sesuai dengan standar yang direkomendasikan oleh *International Telecommunication Union (ITU)*. Sistem ini juga menggunakan *Raspberry Pi* yang digunakan sebagai *gateway* ke *database server Firebase* secara *cloud*. Hasil penelitian ini adalah sistem dapat memberikan informasi presensi kepada mahasiswa dengan *QR code* yang dinamis dan guru dapat melihat hasil rekap kehadiran siswa yang hadir maupun tidak.

Tahun 2019, Didi Juardi melakukan penelitian dengan judul “Presensi Dan Reminder Menggunakan QR Code (Studi Kasus: SMA XXX)”, sistem yang dibuat berbasis *web based* yang menggunakan bahasa pemrograman PHP dengan

pengolahan *database* MySQL dan menggunakan metode SDLC dengan proses model *waterfall*. Tujuan penelitian ini adalah membangun sistem presensi menggunakan *QR code* sebagai media *input* presensi serta *SMS Gateway* sebagai media penyampaian informasi kepada orang tua siswa dan juga dalam penyusunan laporan presensi harian siswa serta staf tata usaha dimudahkan dengan adanya sistem presensi yang dibangun. Hasil penelitian ini adalah sistem absensi dengan basis *web based* yang dapat mencatat presensi yang dilengkapi dengan *SMS gateway* yang akan dikirimkan kepada orang tua dan *reminder* pembayaran yang akan dikirimkan kepada orang tua siswa.

Pada tahun 2020 Mastari melakukan penelitian dengan judul “Sistem presensi mahasiswa menggunakan *QR code* berbasis android pada STMIK Akakom Yogyakarta”. Penelitian bertujuan untuk membangun sistem presensi mahasiswa dengan menggunakan teknologi *QR code* dan *smartphone* yang digunakan sebagai media dalam pengambilan data sehingga proses presensi mahasiswa menjadi lebih efektif dan efisien. Mahasiswa dapat melakukan presensi dengan *scanning* kode *QR code* pada pertemuan mata kuliah, setelah dosen membuka sesi perkuliahan sesuai dengan kelas yang diambil. Jika mahasiswa terdaftar di kelas tersebut maka hasil *scan* tadi akan tersimpan ke database namun apabila mahasiswa tidak terdaftar pada kelas tersebut maka mahasiswa tidak bisa melakukan presensi. Hasil penelitian ini adalah sistem dapat memberikan informasi presensi untuk dosen dan mahasiswa serta laporan presensi yang dapat dikelola oleh bagian adak untuk proses rekapitulasi data presensi mahasiswa.

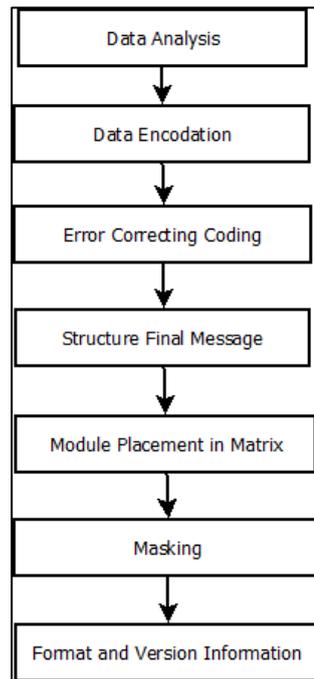
Kemudian, penelitian yang akan dilakukan oleh penulis adalah membuat

sebuah aplikasi presensi yang menggunakan teknologi *QR code*. Penelitian yang akan dilakukan dengan judul “Implementasi *QR Code* Menggunakan Flutter Untuk Melakukan Presensi Mahasiswa Berbasis Mobile (Studi Kasus : STMIK Akakom Yogyakarta)” ini merupakan pengembangan dari penelitian yang sudah dilakukan oleh Mastari sebelumnya dengan menggunakan teknologi yang berbeda. Pada penelitian yang akan dilakukan, terdapat penambahan fitur pemberitahuan kelas dan bahasa pemrograman yang akan digunakan adalah dart dan kotlin. Aplikasi yang akan dibangun merupakan aplikasi yang akan di-*install* pada *smartphone* mahasiswa. Mahasiswa dapat melakukan presensi dengan melakukan *scan* pada *QR code* yang ditampilkan oleh dosen. Untuk pemberitahuan kelas, dosen bisa membuat dan membagikannya melalui firebase console. Hasil penelitian ini adalah aplikasi dapat digunakan untuk presensi, melihat rekap kehadiran dan melihat pemberitahuan kelas.

## **2.2 Dasar Teori**

### **2.2.1 QR Code**

*QR code* merupakan barcode matriks dua dimensi yang dapat dibaca oleh *QR code reader* khusus dan ponsel kamera. *QR code* memungkinkan penggunaanya untuk berbagi informasi. (Price, 2013) Fungsi utama *QR code* yaitu kode yang dapat dengan mudah dibaca oleh *QR code reader* dan memiliki *respons* yang cepat. *QR code* mampu menyimpan informasi secara *horizontal* dan *vertical* atau dua arah. Oleh karena itu, *QR code* secara otomatis dapat menampung informasi yang lebih banyak daripada kode batang. (Roni Habibi, 2000)



**Gambar 2.1 Flowchart Pembangkitan QR Code**

Langkah-langkah pembuatan *QR code* meliputi penentuan kapasitas data yang akan dikodekan dan *encode* data. Langkah-langkahnya yaitu :

1. Menentukan Kapasitas. Kapasitas dari *QR code* ini ditentukan oleh versinya, berdasarkan tingkat *error correction* dan tipe data yang akan dikodekan (misalnya numerik dan alfanumerik). Langkah pertama yang perlu diperhatikan dan dilakukan yaitu mempertimbangkan tingkat koreksi kesalahan, kemudian tentukan versi yang akan digunakannya nanti.
2. Encode (pengkodean) data. Pengkodean data menjadi sebuah *QR code* dilakukan melalui serangkaian perhitungan sebagai berikut :
  - a. Menentukan tipe data.

Data akan dibaca tipe datanya terlebih dahulu. Masing-masing tipe data akan disimpan ke dalam representasi bilangan biner 4 bit dan mempunyai panjang karakter penyimpanan tertentu sehingga lebih

spesifik.

- b. Mengubah data ke dalam bentuk biner.

Data yang sudah diketahui tipe datanya nanti akan diubah ke dalam biner. Misalnya 'buku saya' memiliki tipe data alfanumerik sehingga akan dikonversi ke dalam 9 karakter alfanumerik.

- c. Mengubah biner ke dalam bentuk desimal.

Data yang sudah diubah ke dalam bentuk biner, akan diubah ke dalam bentuk desimal berdasarkan kapasitas dari masing-masing versi *QR code* yang telah ditentukan. Data biner dari 'buku saya' jika diubah ke dalam desimal menjadi 32 74 13 118 89 192 242 20 236 17 236 17 236. Data tersebut adalah data hasil representasi biner sebelum dilakukan perhitungan *error correction*.

- d. *Error Correction*.

*Coding message polynomial* ini juga yang telah diperoleh pada proses konversi data ke dalam bentuk biner kemudian akan dilakukan perhitungan *error correction* setelahnya.

- e. Alokasi data.

Data hasil *encode* ini juga berupa *final message* sehingga dialokasikan ke dalam bentuk gambar *QR code*. Data yang nantinya akan dialokasikan adalah data hasil representasi biner dan juga data hasil perhitungan koreksi kesalahan sehingga lebih efektif.

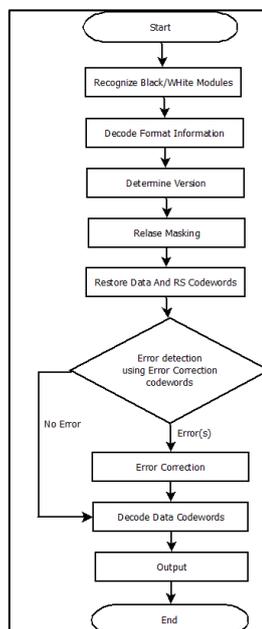
- f. Penentuan pola data.

Pola data juga akan menentukan banyaknya pinalti pada *QR code*.

Pinalti adalah banyaknya modul *error* yang dapat dikembalikan dan berupa kesalahan. Pinalti yang sedikit akan membuat pembacaan *QR code* menjadi lebih mudah dan *correct*. Penentuan pola data dilakukan dengan cara mencoba semua pola data yang ada, kemudian nantinya dipilih pola data yang menghasilkan pinalti paling minimum.

g. Penentuan format informasi.

Data format informasi terdiri juga dari tingkat *error correction* dan indikator pembentuk pola sebanyak 15 bit. Format informasi terdiri dari 2 bit untuk *error correction*, 3 bit untuk pembentukan pola dan 10 bit untuk format informasi data sehingga data dapat dipercaya keasliannya. (Roni Habibi D. A., 2000)



**Gambar 2.2 Flowchart Pembacaan QR Code**

Langkah-langkah untuk membaca *QR Code* menjadi teks aslinya merupakan kebalikan dari langkah-langkah pada pembangkitan *QR Code*. Secara umum

prosedur pembacaan *QR Code* dapat dijelaskan dengan flowchart pada gambar 2.2.

### **2.2.2 Push Notification**

*Push Notification* adalah mekanisme untuk mengirimkan informasi dari *server* ke *client* berdasarkan suatu event tertentu. (Sonya, 2016) *Push notification* akan meningkatkan *usability* dari aplikasi *mobile*. Kehadiran *push notification* ini efektif untuk memberikan informasi kepada pengguna aplikasi *mobile*. (Y, 2016)

Terdapat dua kategori yang membedakan *push notification*, yang pertama *In-app push notification* yang merupakan notifikasi yang akan diterima pada perangkat *mobile*. Pemberitahuan ini biasanya dikirimkan dari aplikasi yang terpasang pada perangkat *mobile*. Kemudian untuk yang kedua merupakan *web push notification* yaitu notifikasi yang diterima dari *website* yang dikunjungi. (Team, 2021)

### **2.2.3 Flutter**

Flutter merupakan platform pengembangan aplikasi *mobile* yang dikembangkan oleh Google untuk membuat aplikasi di *ios*, *Android*, *web app* dan juga *desktop app*. (Kurosaki, 2020) Beberapa kemampuan flutter yaitu memiliki *hot reload*, widget banyak dan memiliki tampilan yang menarik. (Giordano, 2019) Flutter memiliki tujuan untuk menjadi pengubah dalam pengembangan aplikasi seluler. Flutter menyediakan semua alat yang dibutuhkan oleh pengembang untuk membuat aplikasi yang mengagumkan tanpa kekurangan dalam kinerja dan skalabilitas. Flutter dalam struktur intinya memiliki beberapa konsep yang fokus pada kinerja aplikasi dan antarmuka pengguna. Flutter menggunakan dart untuk menyediakan alat yang berkontribusi pada produktivitas pengembang dalam fase pengembangan dan untuk membangun aplikasi yang dipotimalkan untuk publikasi.

(Biessek, 2019)

Flutter merupakan projek *open-source* yang dihosting di github dengan kontribusi dari google dan komunitas. Flutter menggunakan dart, bahasa berorientasi objek modern yang mengompilasi ke kode *native ARM* dan kode javascript. Flutter menggunakan mesin *rendering Skia 2D* yang bekerja dengan berbagai jenis platform perangkat keras dan perangkat lunak dan juga digunakan oleh google chrome, chrome os, android, mozilla firefox, firefox os dan lain-lain.

Flutter menggunakan dart untuk membuat UI (*user interface*) pengguna. Flutter bersifat deklaratif, dengan kata lain flutter membangun UI untuk mencerminkan *state* aplikasi. Ketika *state* (data) berubah, UI akan berubah dan flutter membuat instance baru dengan widget. (Napoli, 2019)

Dart adalah bahasa pemrograman utama yang dipakai untuk mengembangkan *framework* flutter dan dikembangkan langsung oleh google. (Kurosaki, 2020) Flutter membutuhkan bahasa modern tingkat tinggi agar mampu memberikan pengalaman terbaik kepada pengembang dan memungkinkan pembuatan aplikasi seluler yang mengagumkan. Dart bisa digunakan untuk membuat aplikasi web, android, ios dan juga menjalankan server. (Biessek, 2019) Dart dapat digunakan untuk membuat satu aplikasi yang kodenya dapat digunakan di berbagai platform. Aplikasi yang dibuat juga bisa digunakan pada android maupun ios. Dart juga bisa digunakan untuk *web development*. (Wiraganda, 2019)

Dart diluncurkan pada tahun 2011 dan telah berkembang sejak saat itu. Dart merilis versi stabilnya pada tahun 2013 dengan perubahan besar termasuk rilis Dart 2.0 menjelang akhir 2018 :

- Dart fokus pada konsep pengembangan web dengan tujuan utama mengganti JavaScript : namun sekarang dart fokus kepada pengembangan *mobile* pada flutter.
- Dart mencoba memecahkan masalah JavaScript : JavaScript tidak menyediakan ketahanan dari gabungan beberapa bahasa. Jadi, dart diluncurkan sebagai penerus JavaScript.
- Dart menawarkan kinerja terbaik dengan alat yang lebih baik untuk proyek skala besar: Dart memiliki *tools* yang modern dan stabil yang disediakan oleh plugin IDE, dart telah dirancang untuk mendapatkan kinerja terbaik.
- Dart menambahkan fitur OOP dan menyeimbangkan antara fleksibilitas dan ketahanan.

Dart adalah bahasa lintas platform modern yang terus meningkatkan fitur-fiturnya, membuatnya lebih baik dan fleksibel. Itulah mengapa flutter memilih dart sebagai bahasa yang digunakan. (Biessek, 2019)

#### **2.2.4 Kotlin**

Kotlin dibuat oleh tim programmer dari perusahaan JetBrains yang berpusat di Saint Petersburg Rusia. Kotlin adalah bahasa pemrograman bertipe data statis. Kotlin secara sintaks tidak kompatibel dengan bahasa Java, namun kotlin dirancang dapat untuk bisa bekerja sama dengan pustaka yang berasal dari bahasa Java. Kotlin tidak memerlukan titik koma sebagai pembatas statement, cukup ganti baris. (S.Kom, 2018)

### **2.2.5 Presensi**

Presensi adalah suatu pendaftaran kehadiran bagian dari pelaporan aktivitas suatu institusi, atau komponen institusi itu sendiri yang berisi data kehadiran yang disusun dan diatur sedemikian rupa sehingga mudah untuk dicari dan dipergunakan apabila sewaktu-waktu diperlukan oleh pihak yang berkepentingan. (Purwanto, 2009)

### **2.2.6 Smartphone**

*Smartphone* merupakan *platform* yang menjalankan aplikasi-aplikasi *mobile*. (Prasetya, 2013) Sedangkan menurut Ali Zaki, *smartphone* yang berarti telepon pintar, merupakan telepon seluler yang memiliki kemampuan seperti PC walaupun terbatas. Selain itu, *smartphone* juga mendukung email dan organizer. Fitur lainnya adalah kemampuannya untuk ditambah aplikasi-aplikasi baru. (Zaki, 1999)

### **2.2.7 Android Studio**

Android Studio adalah *Integrated Development Environment* (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna. Android Studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas para pengembang untuk membuat aplikasi Android. (Suryana, 2018)

### **2.2.8 Firebase**

Firebase adalah sebuah platform yang menyediakan beberapa layanan untuk memudahkan *developer* membangun aplikasi web, android dan ios maupun unity. (Akbar, 2018)

a. Cloud Firestore

*Cloud firestore* adalah database yang fleksibel dan skalabel untuk pengembangan seluler, web dan server dari firebase dan *google cloud platform*. *Cloud firestore* membuat data tetap terhubung di aplikasi klien melalui pemroses *realtime* dan menawarkan dukungan secara *offline* untuk seluler dan web. Dengan begitu, *developer* dapat mem-*build* aplikasi yang responsif dan mampu bekerja tanpa harus bergantung pada koneksi jaringan.

b. Firebase Cloud Messaging

*Firebase cloud messaging* adalah solusi pertukaran pesan lintas platform yang dapat digunakan untuk mengirim pesan secara terpercaya tanpa biaya. Pengguna dapat mengirim pesan notifikasi untuk mendorong interaksi kembali dan retensi pengguna.