

## BAB II TINJAUAN PUSTAKA

### 2.1 Penelitian yang terdahulu

Adapun hasil-hasil penelitian yang dijadikan perbandingan tidak terlepas dari topik penelitian mengenai Aplikasi Presensi Mata Kuliah Berbasis Teknologi *QR Code* ini. Berikut jurnal terdahulu dari beberapa jurnal terkait dengan penelitian yang dilakukan pemula yang dijelaskan pada Tabel 2. 1 berikut.

**Tabel 2. 1** Penelitian yang terdahulu

No	Judul Referensi	Penulis	Tahun	Metode Penelitian	Hasil Penelitian
1.	Aplikasi Presensi Mahasiswa dengan Menggunakan <i>QR Code</i> Berbasis <i>Android</i> pada Universitas Bina Darma	Al Satrio, Muhamat Abdillah, L. A. Syazili, Ahmad	2017	- Metode pengembangan sistem atau perangkat lunak. - Metode pengembangan sistem yang digunakan dalam penelitian ini adalah model <i>waterfall</i> .	Hasil penelitian dapat dilengkapi dengan tabel, grafik, atau gambar. Bagian pembahasan memaparkan hasil pengolahan data, interpretasi hasil penelitian yang diperoleh, mengaitkan dengan sumber rujukan yang relevan.

2	Penerapan Teknologi <i>QR Code</i> Berbasis <i>Web</i> untuk Absensi Pegawai pada BKPSDM Kabupaten Tanah Laut Herfia	Herfia Rhomadhona	2018	Metode pengembangan sistem ini menggunakan metode <i>waterfall</i>	Dalam penelitian ini menghasilkan aplikasi presensi pegawai pada sistem ini dapat memudahkan admin dalam mengelola data absensi pegawai, dapat mengelola data dan informasi pegawai secara akurat.
3.	Pemanfaatan Teknologi <i>QR Code</i> Pada Sistem Presensi Mahasiswa Berbasis <i>Mobile</i>	Yiyi Supendi, Irwin Supriadi, dan Agustinus A. W. Isto	2019	Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode prototipe sebagai model pengembang perangkat lunak.	Penelitian ini menghasilkan aplikasi presensi perkuliahan mahasiswa berbasis <i>website (back end)</i> dan juga <i>android (front end)</i> . Sistem ini sendiri memiliki 3 aktor yaitu admin, akademik, dosen, dan mahasiswa. Akademik memiliki akses untuk mengelola

					mata kuliah, dosen, mahasiswa, ruangan/kelas, jadwal kuliah, dan mengelola presensi mahasiswa melalui aplikasi <i>website</i> .
4.	Penerapan Aplikasi Presensi Siswa Menggunakan <i>QR Code</i> di SMAN 17 Surabaya	Endang Restuningsih	2019	Model penelitian yang digunakan dalam pengumpulan data menggunakan metode observasi dan wawancara, Pada analisis kebutuhan bagian ini akan dibagi menjadi dua bagian yaitu kebutuhan fungsional dan kebutuhan non fungsional.	Pada penelitian ini menghasilkan aplikasi presensi untuk siswa di SMAN 17 Surabaya. Terdapat 3 pilihan menu, Rekap Presensi, Presensi Hari Ini, dan Scan <i>QR Code</i> . Menu Rekap Presensi, digunakan untuk melihat rekap kehadiran, telat, sakit, izin, alpha per semester ganjil, genap dan tahun. Menu Presensi Hari Ini, digunakan

					<p>untuk mengetahui siswa yang hadir atau tidak pada hari tersebut. Menu Scan <i>QR Code</i>, digunakan untuk melakukan presensi dengan menscan <i>QR Code</i> pada kartu pelajar/presensi siswa.</p>
5.	<p>Pengembangan Aplikasi Mobile Presensi Mahasiswa Berbasis <i>QR-Code</i> Di Universitas Klabat</p>	<p>Adam, Stenly Ibrahim Lengkong, Oktoverano Pungus, Stenly</p>	2021	<p>Metode penelitian ini adalah <i>Software Development Life Cycle (SDLC)</i> dan model yang digunakan adalah <i>Waterfall</i>.</p>	<p>Hasil penelitian aplikasi yang dikembangkan dapat membuat proses presensi mahasiswa di Universitas Klabat menjadi lebih efisien dan mengurangi segala bentuk kecurangan yang sering terjadi</p>

## 2.2 Presensi

Presensi adalah suatu pendataan 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).

### **2.3 Sistem Presensi Konvensional**

Sistem Presensi Konvensional masih banyak digunakan oleh beberapa kampus dan institusi pendidikan saat ini. Namun, sistem ini memiliki beberapa kelemahan yang cukup signifikan. Penelitian yang dilakukan menunjukkan bahwa sistem presensi konvensional memiliki beberapa masalah seperti mudah untuk dipalsukan dan sulit untuk dilacak apabila terjadi kecurangan. Selain itu, sistem ini juga membutuhkan waktu dan tenaga yang cukup banyak untuk mengumpulkan dan memproses data kehadiran mahasiswa secara manual (Hanif, 2019).. Penelitian selanjutnya yang dilakukan juga menunjukkan bahwa sistem presensi konvensional rentan terhadap kecurangan. Mahasiswa dapat meminta teman mereka untuk melakukan presensi atas nama mereka atau memalsukan tanda tangan atau stempel pada daftar hadir (Pratama, 2017). Selain itu, proses pengumpulan dan pengolahan data kehadiran mahasiswa secara manual juga dapat memakan waktu yang cukup lama dan menimbulkan kesalahan dalam pengolahan data.

### **2.4 Quick Response Code**

*Quick Response Code (QR Code)* adalah model kode (2D) tipe baru yang membentuk pola hasil kombinasi garis horizontal dan vertikal, yang lebih fleksibel dan lebih cepat dari jenis stiker barkode (1D), sehingga saat memindai kode batang, pola disimpan dalam *QR Code* untuk aplikasi yang dirancang dapat dibaca dengan cepat. *Quick Response Code* sering di sebut *QR Code* adalah semacam simbol dua dimensi yang dikembangkan oleh Denso Wave yang merupakan anak perusahaan dari *Toyota* sebuah perusahaan Jepang pada tahun 1994 (Ching, 2010). Tujuan dari *QR Code* ini adalah untuk menyampaikan informasi secara cepat dan juga mendapat tanggapan secara cepat. Pada awalnya *QR Code* digunakan untuk pelacakan bagian kendaraan untuk manufacturing. Namun sekarang, telah digunakan untuk komersil yang ditujukan pada pengguna telepon seluler (Muharom, 2016).

### **2.5 Android**

*Android* adalah sistem operasi seluler yang dikembangkan oleh *Google* . Ini didasarkan pada versi modifikasi dari *kernel Linux* dan perangkat lunak open source

lainnya, dan dirancang terutama untuk perangkat seluler layar sentuh seperti smartphone dan tablet (Restuningsih, 2019). Selain itu, *Google* memiliki TV *Android* yang dikembangkan untuk televisi, *Android Auto* untuk mobil, dan *Wear OS* untuk jam tangan pintar yang dapat dikenakan, masing-masing dengan antarmuka pengguna khusus. *Android* versi juga digunakan di konsol game, kamera digital, komputer, dan perangkat elektronik lainnya. Awalnya dikembangkan oleh *Android Inc.*, diakuisisi oleh *Google* pada tahun 2005, *Android* diluncurkan pada tahun 2007, dengan perangkat *Android* komersial pertama dirilis pada September 2008. Untuk Versi stabil untuk saat ini adalah *Android 12*. *Android 12* dirilis secara publik pada 4 Oktober 2021 dan merupakan versi *Android* terbaru hingga saat ini (*Android 12*, 2023).

## **2.6 *Firestore***

*Firestore* adalah suatu layanan dari *Google* yang digunakan untuk mempermudah para pengembang aplikasi dalam mengembangkan aplikasi. *Firestore Realtime Database* memungkinkan pengguna membangun aplikasi yang kaya dan kolaboratif dengan mengakses database secara aman langsung dari kode sisi klien. Data disimpan ke drive lokal Anda. Peristiwa real-time terjadi terus menerus bahkan dalam mode offline, memberikan pengalaman yang responsif bagi pengguna akhir. Saat konektivitas perangkat dipulihkan, database real-time menyinkronkan perubahan data lokal dengan pembaruan jarak jauh yang dilakukan saat klien sedang offline, secara otomatis menggabungkan perbedaan apa pun. *Realtime Database* adalah database *NoSQL*, sehingga memiliki pengoptimalan dan fungsionalitas yang berbeda dengan database terkait (Kumar, 2018).

## **2.7 *Dart***

*Dart* merupakan bahasa pemrograman yang dibuat oleh *Google* dan didesain oleh Lars Bak dan Kasper Lund. Bahasa pemrograman *Dart* dapat digunakan untuk membangun aplikasi *server* atau dalam bentuk *command line interface*, *web*, ataupun *mobile* (*Android* dan *iOS*). *Dart* merupakan Bahasa pemrograman yang mendukung adanya pendefinisian fungsi di luar kelas atau sering disebut dengan *toplevel function*. Dalam *Dart*. Kode program utama disimpan di dalam fungsi `main()` sama halnya seperti *C/C++* (Muslim, 2022).

## 2.8 Flutter

*Flutter* adalah sebuah *framework open-source* atau SDK yang dikembangkan oleh Google untuk membangun antarmuka (*Customer interface/UI*) aplikasi yang memiliki kinerja tinggi serta dapat dipublikasi ke platform *Android* dan *IOS* dari *codebase* tunggal. *Flutter* menggunakan bahasa pemrograman *Dart* yang pastinya terasa familiar dengan bahasa pemrograman *Java* atau *Javascript* (Erzed, 2022). *Flutter* menggunakan bahasa pemrograman *dart* sehingga dapat dengan mudah untuk dipelajari. Bahasa pemrograman *dart* dianggap mudah apabila telah terbiasa dan familiar menggunakan Bahasa pemrograman *Java* atau *Javascript*. Selain itu, *flutter* juga menyediakan kerangka *reactive functional*, mesin *render 2D*, *widget* yang siap untuk digunakan, dan *tools* yang digunakan untuk membantu dalam melakukan pengembangan aplikasi (Muslim, 2022).

## 2.9 Visual Studio Code

*Visual Studio Code (VS Code)* adalah kode editor yang bersifat *open source* untuk pengembangan dan debugging aplikasi cloud dan situs web modern yang tersedia secara gratis di *Linux*, *OS X*, dan *Windows*. *VS Code* mendukung lebih dari 30 bahasa pemrograman, *markup* dan *database* yang berbeda, beberapa diantaranya adalah *PHP*, *HTML*, *CSS* dan *SQL* (Gamma, 2016).

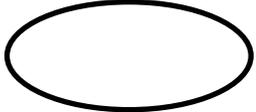
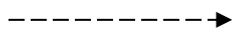
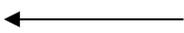
## 2.10 Use Case Diagram

*Use case diagram* adalah gambaran grafis dari beberapa atau semua *actor*, *use case*, dan interaksi diantaranya yang memperkenalkan suatu sistem. *Use case diagram* tidak menjelaskan secara detil tentang penggunaan *use case*, tetapi hanya memberi gambaran singkat hubungan antara *use case*, aktor, dan sistem (Pratama, 2019).

Berikut simbol – simbol *use case diagram* sebagaimana yang dijelaskan pada Tabel 2. 2.

**Tabel 2. 2** Simbol -simbol *Use Case Diagram*

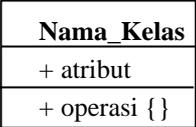
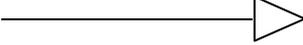
NO	SIMBOL	KETERANGAN
1	<p><i>Actor</i></p> 	<p>Merupakan Penggunaan dari sistem. Penamaan aktor menggunakan kata benda.</p>

2	<p style="text-align: center;"><i>Use case</i></p> 	Merupakan pekerjaan yang dilakukan oleh aktor. Penamaan <i>use case</i> dengan kata kerja.
3	<p style="text-align: center;">-End -End2</p> 	Hubungan antara aktor dengan <i>use case</i>
4	<p style="text-align: center;"><i>Association</i></p> 	<i>Association</i> adalah Abstraksi penghubung antara aktor dan use case
5	<p style="text-align: center;"><i>Generalitation</i></p> 	<i>Generalitation</i> Menunjukkan spesialisasi aktor untuk dapat berpartisipasi dengan <i>use case</i>
6	<p style="text-align: center;">&lt;&lt; include &gt;&gt;</p> 	Hubungan antara <i>use case</i> dengan <i>use case</i> , <i>include</i> menyatakan bahwa sebelum <i>Include</i> pekerjaan dilakukan harus mengerjakan pekerjaan lain terlebih dahulu.
7	<p style="text-align: center;">&lt;&lt; extend &gt;&gt;</p> 	Hubungan antara <i>use case</i> dengan <i>use case</i> , <i>extends</i> menyatakan bahwa jika pekerjaan yang dilakukan tidak sesuai atau terdapat kondisi khusus, melakukan pekerjaan itu.

### 2.11 Class Diagram

*Class Diagram* adalah jenis diagram struktur statis yang menjelaskan struktur dari sebuah sistem dengan menunjukkan kelas-kelas, *interface*, atribut-atribut dan hubungan antar kelas dari struktur sistem tersebut. *Class diagram* akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek (Wahyudi, 2018). Berikut pada Tabel 2. 3 simbol yang digunakan dalam *Class Diagram*.

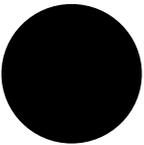
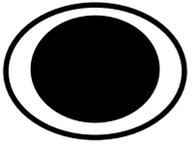
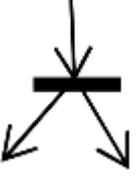
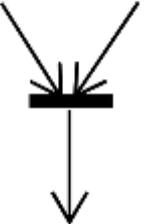
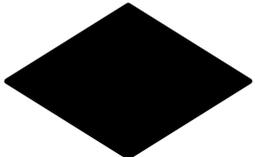
**Tabel 2. 3** Simbol- simbol Class Diagram

No.	Simbol	Deskripsi
1.	<p>Kelas</p> 	Kelas pada struktur sistem.
2.	<p>Antarmuka / <i>Interface</i></p> 	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek.
3.	<p>Asosiasi / Association</p> 	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai <i>multiplicity</i> .
4.	<p>Asosiasi berarah / <i>Directed Association</i></p> 	Relasi antarkelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
5.	<p>Generalisasi</p> 	Relasi antarkelas dengan makna generalisasi – spesialisasi (umum khusus).
6.	<p>Kebergantungan / <i>Dependency</i></p> 	Relasi antarkelas dengan makna kebergantungan antar kelas.
7.	<p>Agregasi / <i>Aggregation</i></p> 	Relasi antarkelas dengan makna semua-bagian ( <i>whole-part</i> ).

### 2.12 Activity Diagram

*Activity Diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis (Hendini, 2016). Simbol-simbol yang digunakan dalam *activity Diagram* sebagaimana yang dijelaskan pada Tabel 2. 4.

Tabel 2. 4 Simbol - simbol *Activity Diagram*

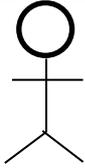
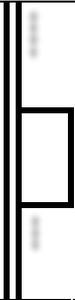
NO	SIMBOL	KETERANGAN
1		<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas.
2		<i>End Point</i> , akhir aktivitas
3		<i>Activities</i> , menggambar kan suatu proses/kegiatan bisnis
4		<i>Fork</i> /percabangan, digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.
5		<i>Join</i> (penggabungan) atau <i>rake</i> , digunakan untuk menunjukkan adanya dekomposisi.
6		<i>Decision Points</i> , menggambarkan pilihan untuk pengambilan keputusan, <i>True</i> dan <i>False</i> .
7		<i>Swimlane</i> , pembagian activity diagram untuk menunjukkan siapa melakukan apa.

### 2.13 Sequence Diagram

*Sequence Diagram* adalah salah satu jenis diagram yang terdapat dalam UML yang memvisualisasikan kolaborasi dinamis antara sejumlah objek (Pratama, 2019).

Berikut dibawah ini adalah simbol-simbol yang digunakan dalam *activity diagram* disertai dengan keterangan fungsinya sebagaimana dijelaskan pada Tabel 2.5.

**Tabel 2. 5** Simbol - simbol *Sequence Diagram*

Simbol	Nama	Keterangan
	<i>Actor</i>	Menggambarkan orang yang sedang berinteraksi dengan sistem.
	<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan.
	<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari form.
	<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel.
	<i>A focus of Control &amp; A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya message.

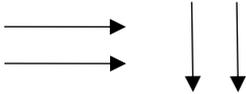
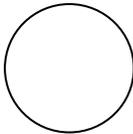
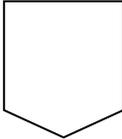
	<i>A message</i>	Menggambarkan pengiriman pesan.
---	------------------	---------------------------------

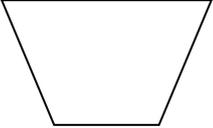
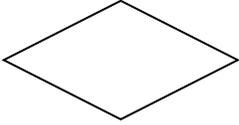
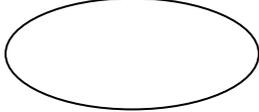
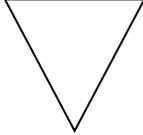
### 2.14 Flowchart

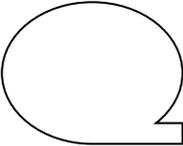
*Flowchart* atau diagram alir merupakan chart (bagan) yang menunjukkan hasil (flow) dalam program atau prosedur sistem secara logika. Digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi. Diagram ini bisa memberi solusi selangkah demi selangkah untuk penyelesaian masalah yang ada di dalam proses atau algoritma tersebut (Sitorus, 2016).

Pada Tabel 2.5 yang berisi simbol-simbol yang digunakan dalam flowchart disertai dengan keterangan fungsinya.

**Tabel 2. 6** Simbol-simbol *flowchart*

NO	SIMBOL	KETERANGAN
1	<p><i>Flow Direction</i></p> 	Simbol arus / <i>flow</i> , berfungsi untuk menyatakan jalannya arus suatu proses.
2	<p><i>Connector</i></p> 	Simbol <i>connector</i> , berfungsi untuk menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama.
3	<p><i>Offline Connector</i></p> 	Simbol <i>offline connector</i> , berfungsi untuk menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda.
4	<p><i>Process/Proses</i></p> 	Simbol <i>process</i> , berfungsi untuk menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.

5	<p><i>Manual Operation</i></p> 	<p>Simbol <i>manual</i>, berfungsi untuk menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer.</p>
6	<p><i>Decision (Keputusan)</i></p> 	<p>Simbol <i>decision</i>, berfungsi untuk menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya/tidak.</p>
7	<p><i>Terminal</i> (mulai atau berhenti)</p> 	<p>Simbol <i>terminal</i>, berfungsi untuk menyatakan permulaan atau akhir suatu program.</p>
8	<p><i>Predefined process</i></p> 	<p>Simbol <i>predefined process</i>, berfungsi untuk menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.</p>
9	<p><i>Keying operation</i></p> 	<p>Simbol <i>keying operation</i>, berfungsi untuk menyatakan segala jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai keyboard.</p>
10	<p><i>Offline-storage</i></p> 	<p>Simbol <i>offline-storage</i>, berfungsi untuk menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu</p>
11	<p><i>Manual input</i></p> 	<p>Simbol <i>manual input</i>, berfungsi untuk memasukkan data secara manual dengan menggunakan online keyboard.</p>

12	<p style="text-align: center;"><i>Input/output</i></p> 	<p>Simbol <i>input/output</i>, berfungsi untuk menyatakan proses input atau output tanpa tergantung jenis peralatannya.</p>
13	<p style="text-align: center;"><i>Magnetic tape</i></p> 	<p>Simbol <i>magnetic tape</i>, berfungsi untuk menyatakan input berasal dari pita magnetis atau output disimpan ke pita magnetis.</p>
14	<p style="text-align: center;"><i>Disk storage</i></p> 	<p>Simbol <i>disk storage</i>, berfungsi untuk menyatakan input berasal dari disk atau output disimpan ke disk.</p>
15	<p style="text-align: center;"><i>Document</i></p> 	<p>Simbol <i>document</i>, berfungsi untuk mencetak keluaran dalam bentuk dokumen (melalui printer).</p>
16	<p style="text-align: center;"><i>Punched card</i></p> 	<p>Simbol <i>punched card</i>, berfungsi untuk menyatakan input berasal dari kartu atau output ditulis ke kartu.</p>