

**RANCANG BANGUN APLIKASI *MOBILE LIBRARY*
MANAGEMENT BERBASIS *QR CODE*
DAN *CLOUD FIRESTORE***

SKRIPSI



**KARLA YUSMITA SALIPADANG
H13116521**

**PROGRAM STUDI ILMU KOMPUTER DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN
MAKASSAR
NOVEMBER 2020**

**RANCANG BANGUN APLIKASI *MOBILE LIBRARY*
MANAGEMENT BERBASIS *QR CODE*
DAN *CLOUD FIRESTORE***

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Sains pada
Program Studi Ilmu Komputer Departemen Matematika Fakultas Matematika
dan Ilmu Pengetahuan Alam Universitas Hasanuddin**

KARLA YUSMITA SALIPADANG

H13116521

**PROGRAM STUDI ILMU KOMPUTER DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS HASANUDDIN**

MAKASSAR

NOVEMBER 2020

LEMBAR PERNYATAAN KEONTETIKAN

Saya yang bertanda tangan dibawah ini memyatakan dengan sungguh-sungguh bahwa skripsi yang saya buat dengan judul:

RANCANG BANGUN APLIKASI MOBILE LIBRARY MANAGEMENT BERBASIS QR CODE DAN CLOUD FIRESTORE

adalah benar hasil karya saya sendiri, bukan hasil plagiat dan belum pernah di publikasikan dalam bentuk apapun

Makassar, 24 November 2020

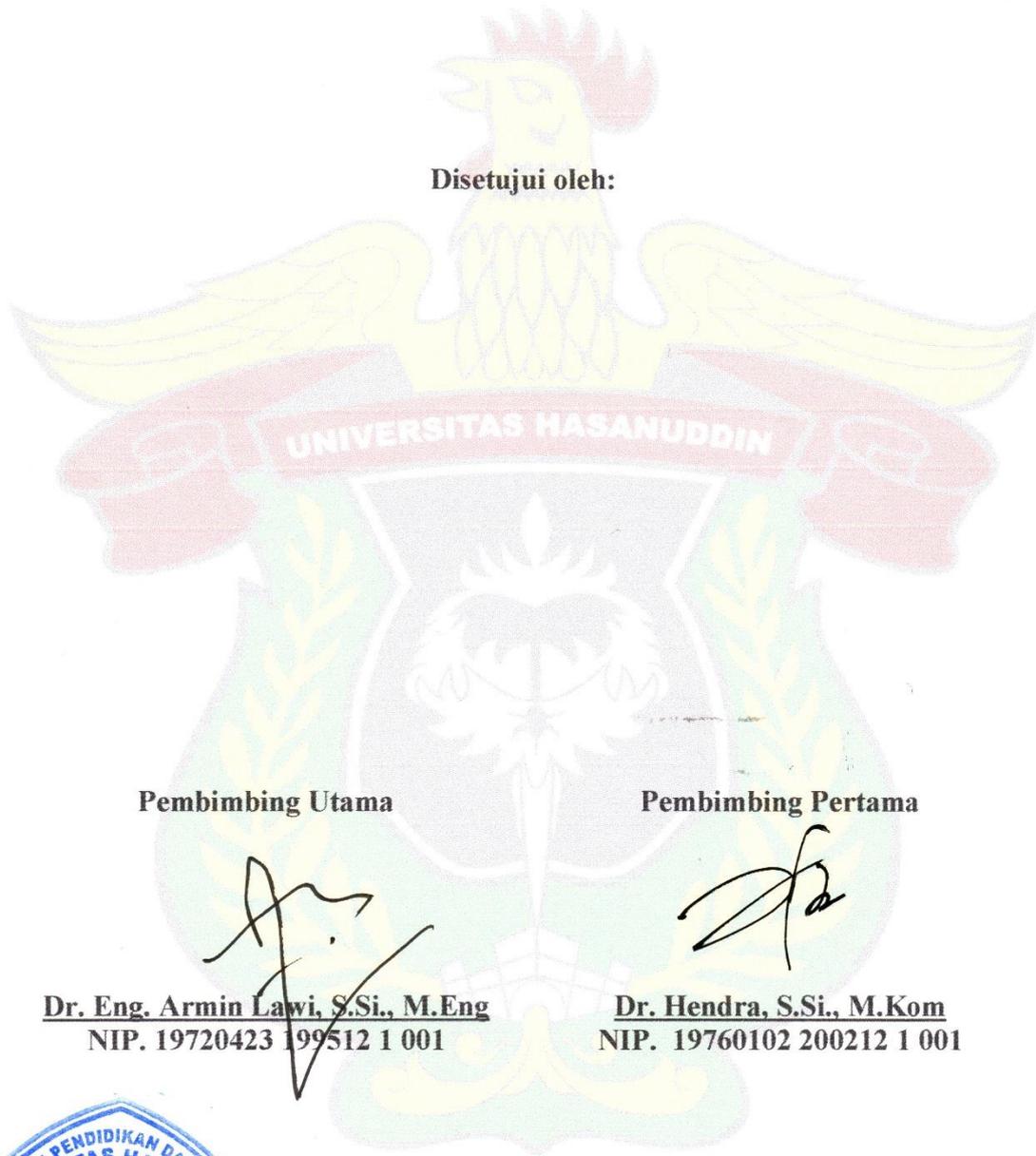


Karla Yusmita Salipadang

NIM. H131 16 521

**RANCANG BANGUN APLIKASI *MOBILE LIBRARY*
MANAGEMENT BERBASIS *QR CODE*
DAN *CLOUD FIRESTORE***

Disetujui oleh:



Pembimbing Utama

Pembimbing Pertama

Dr. Eng. Armin Lawi, S.Si., M.Eng
NIP. 19720423 199512 1 001

Dr. Hendra, S.Si., M.Kom
NIP. 19760102 200212 1 001



Pada 24 November 2020

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh:

Nama : Karla Yusmita Salipadang
NIM : H13116521
Program Studi : Ilmu Komputer
Judul Skripsi : Rancang Bangun Aplikasi *Mobile Library Management*
Berbasis *QR Code* dan *Cloud Firestore*

Telah berhasil dipertahankan di hadapan dewan penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Sains pada Program Studi Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

DEWAN PENGUJI

Tanda Tangan

1. Ketua : Dr. Eng. Armin Lawi, S.Si., M.Eng (.....)
2. Sekretaris : Dr. Hendra, S.Si., M.Kom (.....)
3. Anggota : Dr. Muhammad Hasbi, M.Sc. (.....)
4. Anggota : Andi Muhammad Anwar, S.Si., M.Si (.....)

Ditetapkan di : Makassar

Tanggal : 24 November 2020



KATA PENGANTAR

Segala puji dan syukur patut penulis panjatkan kehadiran Tuhan Yang Maha Esa karena hanya oleh berkat dan penyertaan-Nya sehingga penulis dapat menyelesaikan skripsi dengan baik dan tepat pada waktu-Nya. Dalam mengerjakan skripsi ini penulis merasa ada banyak tantangan yang harus dihadapi, namun semua itu dapat terselesaikan karena penyertaan Tuhan. Skripsi berjudul **Rancang Bangun Aplikasi *Mobile Library Management* Berbasis *QR Code* dan *Cloud Firestore***. Skripsi ini ditulis untuk memenuhi salah satu persyaratan kelulusan program Sarjana Sains (S.Si) pada Program Studi Ilmu Komputer, jurusan Matematika, Fakultas MIPA, Universitas Hasanuddin.

Penulis menyadari bahwa dalam menyelesaikan skripsi Tuhan turut bekerja melalui banyak orang. Penulis telah menerima banyak bantuan dan dukungan dari banyak pihak. Untuk itu pada kesempatan ini penulis ingin mengucapkan terima kasih atas dukungan dan kontribusi kepada

1. Bapak **Yunus Salipadang** dan Ibu **Mittin Tandidatu**, sebagai orang tua yang senantiasa memberi dukungan yang luar biasa, baik itu berupa doa, materi, dan nasehat-nasehat. Kiranya skripsi ini dapat menjadi bukti dan kebanggaan tersendiri kepada orangtua penulis, bahwa mereka telah berhasil mendidik anaknya hingga menjadi seorang sarjana. Juga kepada **Hapchris Vedelita Salipadang**, sebagai saudara yang terus mendukung penulis. Kiranya skripsi ini dapat menjadi motivasi untuk saudara agar tetap giat dalam menyelesaikan perkuliahan.
2. **Prof. Dr. Dwia Aries Tina Palubuhu**, selaku Rektor Universitas Hasanuddin beserta jajarannya.
3. **Dr. Eng. Amiruddin, M.Si**, selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam beserta jajarannya dan Pegawai Akademik Fakultas MIPA terkhusus **Bapak Sangkala** dan **Bapak Suardi** yang telah membantu penulis dalam menyelesaikan administrasi.
4. **Dr. Nurdin, S.Si., M.Si**, selaku Ketua Departemen Matematika beserta jajarannya dan pegawai akademik departemen Matematika. Serta kepada seluruh dosen pengajar departemen matematika yang telah memberikan ilmu dan nasehat selama penulis menjalani Pendidikan.

5. **Dr. Eng. Armin Lawi, S.Si., M.Eng** dan **Dr. Hendra, S.Si., M.Kom**, sebagai dosen pembimbing atas seluruh ilmu, bimbingan, motivasi, dan nasehat yang diberikan selama penulis membuat skripsi ini.
6. **Dr. Muhammad Hasbi, M.Sc** dan **Andi Muhammad Anwar, S.Si., M.Si**, sebagai dosen penguji yang telah memberikan saran dan kritikan yang membangun sehingga skripsi ini boleh jadi dengan baik.
7. **Grathya Sri Mulyani** dan **Sesilia Shindy Pandin** yang telah menjadi sahabat yang selalu mendukung dalam doa, memberikan motivasi, dan semangat selama menuntut ilmu bersama.
8. **Teman-teman Ilmu Komputer Angkatan 2016** atas kebersamaan dalam perjuangan selama menuntut ilmu.
9. **Nataniel Deo Pune'** yang selalu menemani dan memberikan semangat dan dukungan selama berkuliah dan terkhusus selama menyelesaikan skripsi ini.
10. **Dzul Fahmi M.** yang telah membantu menyelesaikan aplikasi android dan menjadi teman berdiskusi mengenai topik skripsi penulis.
11. **Rekan-rekan PPGT Jemaat Tello Batua** terkhusus kepada **Pdt. Anna Reguspin, Eben H. B. Patandean, Edwin A. Sitolong, Andreanus Torra**, yang telah memberikan motivasi dan doa.
12. **Kakak-kakak dan teman-teman Ilmu Komputer Universitas Hasanuddin** yang selalu terbuka untuk berdiskusi mengenai materi perkuliahan secara khusus topik skripsi penulis.

Pada kesempatan ini juga penulis menyampaikan permohonan maaf kepada seluruh pihak apabila selama masa perkuliahan penulis membuat kesalahan baik itu melalui kata atau tindakan. Penulis dengan hati yang lapang dada menerima setiap kritikan dan saran yang membangun agar dapat menyempurnakan skripsi ini. Besar harapan penulis bahwa skripsi ini dapat bermanfaat sebagai bahan referensi dalam dunia Pendidikan.

Makassar, November 2020

Penulis
Karla Yusmita Salipadang

PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Hasanuddin, saya yang bertanda tangan dibawah ini:

Nama : Karla Yusmita Salipadang

NIM : H13116521

Program Studi : Ilmu KOrputer

Departemen : Matematika

Fakultas : Matematika dan Ilmu Pengetahuan Alam

Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin Hak Prediktor Royalti Noneklusif (Non-Exclusive Royalty-free Right) atas tugas akhir saya yang berjudul:

“Rancang Bangun Aplikasi Mobile Library Management Berbasis QR Code dan Cloud Firestore”

Terkait dengan hal diatas, maka pihak universitas berhak menyimpan, mengalih-media/format-kan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di Makassar pada tanggal 25 November 2020

Yang menyatakan

Karla Yusmita Salipadang

ABSTRAK

Sistem perpustakaan konvensional memiliki beberapa kekurangan salah satunya adalah segala proses administrasi peminjaman dan pengembalian buku dilakukan secara manual yang bisa saja mengakibatkan *human error* saat memasukkan data. Untuk itu penulis ingin merancang sistem manajemen perpustakaan dengan memanfaatkan aplikasi android, QR Code, dan *Cloud Firestore*. Aplikasi android dan QR Code berfungsi sebagai *interface* yang akan membuat proses administrasi peminjaman dan pengembalian buku dilakukan secara otomatis. Sedangkan semua data yang ada di sistem akan di simpan dalam *Cloud Firastore*. Hasil dari penelitian ini adalah sebuah aplikasi yang dapat memudahkan proses peminjaman dan pengembalian. Selain itu aplikasi ini dapat memudahkan kerja pustakawan untuk menginput data buku. Sedangkan database yang terbentuk terdiri dari tiga koleksi yaitu buku, aktor, dan bookingList. Sedangkan QR Code berisi ID buku.

Kata Kunci: Sistem Perpustakaan, *Mobile Aplication*, QR Code, *Cloud Firestore*

ABSTRACT

The conventional library system has several drawbacks, one of which is that all administrative processes for borrowing and returning books are done manually which can cause human errors when entering data. For this reason, the author wants to design a library management system by utilizing the Android application, QR Code, and Cloud Firestore. The android application and QR Code function as an interface that will make the administrative process of borrowing and returning books carried out automatically. Meanwhile, all data on the system will be stored in Cloud Firastore. The result of this research is an application that can facilitate the borrowing and repayment process. In addition, this application can facilitate the work of librarians to input book data. While the database that was formed consisted of three collections, namely books, actors, and booking list. While the QR Code contains the ID of the book.

Keywords: Library System, Mobile Application, QR Code, Cloud Firestore

DAFTAR ISI

SAMPUL	ii
LEMBAR PERNYATAAN KEONTETIKAN.....	iii
HALAMAN PERSETUJUAN	Error! Bookmark not defined.
HALAMAN PENGESAHAN.....	v
KATA PENGANTAR	vi
PERNYATAAN PERSETUJUAN PUBLIKASI	viii
ABSTRAK.....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR	xiii
DAFTAR TABEL.....	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Penelitian	2
1.4 Manfaat Penelitian	2
1.5 Batasan Masalah	3
BAB II TINJAUAN PUSTAKA.....	4
2.1 Landasan Teori	4
2.1.1 Perpustakaan.....	4
2.1.2 Android.....	5
2.1.3 QR Code	6
2.1.4 NoSQL.....	10
2.1.5 Cloud Firestore	13
2.1.6 Konsep Model-View-Controller (MVC).....	14
2.1.7 UML (Unified Modeling Language).....	15
2.2 State of the Art.....	16
2.3 Kerangka Konseptual.....	18
BAB III METODE PENELITIAN.....	20
3.1 Waktu dan Lokasi Penelitian	20

3.2	Tahapan Penelitian.....	20
3.3	Rancangan Sistem.....	21
3.4	Software dan Alat	22
BAB IV HASIL DAN PEMBAHASAN		24
4.1	Analisa Kebutuhan Sistem.....	24
4.2	Desain Sistem Manajemen Perpustakaan	28
	4.2.1 Desain <i>Activity Diagram</i>	28
	4.2.2 Rancangan Database Firestore	37
4.3	Implementasi Sistem.....	38
	4.3.1 Pembuatan Database Firestore	38
	4.3.2 Pembuatan Aplikasi.....	38
4.4	Hasil.....	45
	4.4.1 Database Firestore	45
	4.4.2 Pembuatan <i>QR Code</i>	46
	4.4.3 Aplikasi Android	47
BAB V		62
KESIMPILAN DAN SARAN		62
5.1	Kesimpulan	62
5.2	Saran	63
DAFTAR PUSTAKA		64

DAFTAR GAMBAR

Gambar 2.1 Logo android Q	5
Gambar 2.2 Struktur QR Code.....	7
Gambar 2.3 Versi QR Code	8
Gambar 2.4 Struktur data cloud firestore	14
Gambar 2.5 Konsep Model-View-Controller (MVC).....	14
Gambar 3.1 Work flow diagram	21
<i>Gambar 3.2 Use Case diagram.....</i>	<i>22</i>
Gambar 4.1 Prosedur peminjaman buku secara konvensional.....	24
Gambar 4.2 Prosedur perpanjangan waktu peminjaman buku	24
Gambar 4.3 Prosedur pengembalian buku secara konvensional.....	25
Gambar 4.4 Rancangan prosedur peminjaman buku	26
Gambar 4.5 Rancangan prosedur pengembalian buku.....	27
Gambar 4.6 Activity diagram login	28
Gambar 4.7 Activity diagram peminjaman buku	29
Gambar 4.8 Activity diagram pengembalian buku	31
Gambar 4.9 Activity diagram mengubah data buku	32
Gambar 4.10 Activity diagram menambah data buku	33
Gambar 4.11 Activity diagram menghapus data buku.....	34
Gambar 4.12 Activity diagram mengubah data mahasiswa.....	35
Gambar 4.13 Activity diagram menambah data mahasiswa	36
Gambar 4.14 Rancangan database	37
Gambar 4.15 Halaman ringkasan project pada halaman firebase	39
Gambar 4.16 Formulir pendaftaran aplikasi pada firebase	39
Gambar 4.17 File build.gradle(:app).....	40
Gambar 4.18 File gradle tingkat project	40
Gambar 4.19 File gradle tingkat aplikasi	41
Gambar 4.20 File gradle untuk menambahkan dependensi	41
Gambar 4.21 File dalam project android.....	42

Gambar 4.22 Koleksi buku dalam Firestore	45
Gambar 4.23 Koleksi aktor dalam firestore	46
Gambar 4.24 Koleksi terpinjam dan wishlist dalam dokumen ID aktor.....	46
Gambar 4.25 Hasil QR Code ID buku	47
Gambar 4.26 Halaman Login.....	47
Gambar 4.27 Pemberitahuan jika password atau username salah.....	48
Gambar 4.28 Halaman utama admin.....	48
Gambar 4.29 Halaman utama pustakawan.....	49
Gambar 4.30 Halaman utama pengguna	49
Gambar 4.31 Proses scan QR Code	50
Gambar 4.32 Halaman detail buku.....	50
Gambar 4.33 Proses pencarian buku.....	51
Gambar 4.34 Pemberitahuan buku berhasil ditambah ke wishlist	52
Gambar 4.35 Pemberitahuan buku tidak dapat ditambah ke wishlist	52
Gambar 4.36 Fitur Booking pada Halaman Wishlist	53
Gambar 4.37 Buku Berhasil Dibooking.....	53
Gambar 4.38 Buku Tidak Dapat Dibooking	54
Gambar 4.39 Daftar mahasiswa yang mempunyai wishlist.....	55
Gambar 4.40 Pemberitahuan jika pustakawan memilih lebih dari tiga buku	56
Gambar 4.41 Pemberitahuan jika total buku yang akan dipinjam lebih dari tiga buku.....	56
Gambar 4.42 Halaman terpinjam pengguna.....	57
Gambar 4.43 Halaman konfirmasi pengembalian buku.....	58
Gambar 4.44 Formulir pendataan buku	59
Gambar 4.45 Halaman mengubah dan menghapus data buku	59
Gambar 4.46 Formulir pendaftaran mahasiswa	60
Gambar 4.47 Halaman detail mahasiswa	61

DAFTAR TABEL

Tabel 2.1 Jumlah Data yang Dimuat QR Code.....	8
Tabel 2.2 Perbedaan SQL dan NoSQL	11

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan zaman dan kemajuan teknologi membuat perilaku masyarakat menjadi berubah. Kebutuhan masyarakat akan penggunaan media informasi dan komunikasi dalam kehidupan semakin bertambah. Masyarakat menginginkan akses informasi yang cepat dan efisien. Pada zaman sekarang ini kita telah menjumpai bahwa teknologi telah ada dalam hampir setiap aspek kehidupan manusia. Sehingga diperlukan berbagai inovasi untuk memenuhi kebutuhan manusia. Salah satunya adalah dibidang perpustakaan.

Di Indonesia banyak di temukan perpustakaan konvensional/perpustakaan “tradisional” hampir disetiap kota, sekolah, universitas, bahkan kantor. Sistem pada perpustakaan konvensional yang ada saat ini masih perlu diperbaharui agar lebih efisien dan efektif baik bagi pengguna maupun bagi pustakawan. Salah satu contohnya adalah proses administrasi peminjaman buku yang masih dilakukan secara manual yang dapat menyebabkan kesalahan karena *human error*.

Berdasarkan hasil survei Polling Indonesia yang bekerja sama dengan Asosiasi Penyelenggara Jasa Internet Indonesia (APJII), jumlah pengguna internet di Indonesia tumbuh 10,12% pada tahun 2018 (Kompas, 2019). Pada tahun yang sama Lembaga riset digital marketing Emarketer memperkirakan jumlah pengguna aktif di Indonesia lebih dari seratus juta orang. Indonesia dapat menjadi negara dengan pengguna aktif *smartphone* terbesar keempat didunia setelah Cina, India, dan Amerika (Kominfo, 2019). Dari hasil survei diatas dapat disimpulkan bahwa masyarakat Indonesia juga memerlukan akses informasi yang cepat dan mudah. Sistem perpustakaan di Indonesia juga harus di sesuaikan dengan kebutuhan masyarakat. Untuk itu diperlukan inovasi dalam sistem perpustakaan konvensional.

Inovasi sistem perpustakaan yang penulis tawarkan adalah dengan membuat sebuah sistem manajemen perpustakaan yang memanfaatkan teknologi android, *QR code*, dan *cloud firestore*. Penulis tertarik untuk membuat sebuah aplikasi yang dapat digunakan pada *smartphone* dengan menggunakan sistem aplikasi

android. Android merupakan sistem operasi terbanyak yang digunakan pada *smartphone*. Saat ini *QR code* menjadi *tren* dalam bidang bisnis di Indonesia. Jika dibandingkan dengan *barcode*, *QR code* lebih memiliki banyak kelebihan terutama dalam hal kapasitas penyimpanan data dan estetika. *Cloud Firestore* merupakan database fleksibel yang digunakan untuk pengembangan aplikasi seluler. Produk ini dikeluarkan oleh Google.

Diharapkan sistem ini bekerja lebih cepat, akurat, efisien, dan *user friendly* sehingga dapat menyelesaikan permasalahan pada sistem perpustakaan konvensional. Diharapkan sistem ini dapat membantu pekerjaan pustakawan agar lebih efektif dan efisien. Selain itu pengguna juga mendapat menghemat waktu dan biaya saat ingin meminjam buku di perpustakaan.

1.2 Rumusan Masalah

Adapun rumusan masalah dari penelitian ini adalah :

1. Bagaimana membangun database perpustakaan menggunakan *Cloud Firestore*?
2. Bagaimana merancang dan membangun sistem manajemen perpustakaan berbasis Android yang dapat memberi kemudahan untuk melakukan administrasi di perpustakaan?

1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah :

1. Membangun database perpustakaan menggunakan *Cloud Firestore*.
2. Merancang dan membangun sistem manajemen perpustakaan berbasis Android yang dapat memberi kemudahan untuk melakukan administrasi di perpustakaan.

1.4 Manfaat Penelitian

Hasil dari penelitian ini diharapkan dapat memberikan manfaat sebagai berikut :

1. Jangka Pendek
 - a. Sebagai sarana yang dapat membantu pengguna untuk memudahkan proses peminjaman, pengembalian, dan pencarian buku.

- b. Sebagai sarana untuk memudahkan pustakawan dan admin dalam mengerjakan tugasnya yaitu mengontrol pengembalian dan peminjaman buku, memasukkan data buku, dan membuat surat keterangan bebas pustaka.
- c. Meningkatkan mutu pelayanan dari perpustakaan.

2. Jangka Panjang

- a. Mengurangi penggunaan kertas yang biasa digunakan untuk keperluan administrasi.
- b. Menghemat anggaran untuk membeli inventaris kantor seperti komputer dan dapat di ganti dengan menyediakan layanan internet yang berkualitas.

1.5 Batasan Masalah

Batasan masalah dalam penelitian ini adalah

- a. Penulis hanya membuat *usecase diagram* dan *activity diagram* sebagai dokumentasi aplikasi.
- b. Penulis menggunakan konsep *model-view-controller* dalam membuat aplikasi android.
- c. Penulis menggunakan Android Studio sebagai IDE (*Integrated Development Enviroment*) untuk membuat aplikasi android.
- d. Penulis tidak membahas *image processing QR code* dan proses *generate* kode menjadi *QR code*.

BAB II

TINJAUAN PUSTAKA

2.1 Landasan Teori

2.1.1 Perpustakaan

Secara etimologi perpustakaan berasal dari kata “Pustaka” yang berarti buku. Perpustakaan adalah sebuah ruangan atau gedung yang digunakan untuk menyimpan buku dan terbitan lain menurut tata susunan tertentu untuk pembaca dan bukan untuk dijual (Basuki, 1991). Secara garis besar perpustakaan mencakup unsur koleksi, penyimpanan, dan pemakaian (Prabowo, 2013). Dalam UU No. 43 tahun 2007 tentang Perpustakaan menyebutkan bahwa perpustakaan adalah institusi pengelola koleksi karya tulis, karya cetak, dan/atau karya rekam secara profesional dengan sistem yang baku guna memenuhi kebutuhan pendidikan, penelitian, pelestarian, informasi, dan rekreasi para pemustaka. Pemustaka adalah pengguna perpustakaan, yaitu perseorangan, kelompok orang, masyarakat, atau lembaga yang memanfaatkan fasilitas layanan perpustakaan. Pustakawan adalah seseorang yang memiliki kompetensi yang diperoleh melalui pendidikan dan/atau pelatihan kepustakawanan serta mempunyai tugas dan tanggung jawab untuk melaksanakan pengelolaan dan pelayanan perpustakaan .

Perpustakaan memiliki beberapa fungsi yaitu untuk menyimpan koleksi buku dan bahan pustaka. Koleksi ini dapat digunakan sebagai sumber informasi untuk keperluan penelitian dan pembelajaran bagi pengguna. Perpustakaan juga memiliki fungsi sebagai sarana rekreasi. Perpustakaan dapat menyediakan bahan bacaan yang menghibur seperti novel, cerita rakyat, dan puisi. Perpustakaan juga berperan sebagai sarana pendukung dalam pendidikan yang tidak dapat dikesampingkan manfaatnya (Harahap, 2018) karena merupakan sarana yang penting dalam upaya mencerdaskan kehidupan bangsa (Prabowo, 2013).

Manfaat perpustakaan adalah sebagai tempat pengumpul, penyimpan, dan pemeliharaan berbagai koleksi bahan pustaka dan sebagai sumber informasi. Menurut jenisnya perpustakaan di bagi menjadi empat jenis yaitu perpustakaan konvensional, perpustakaan hibrida, perpustakaan digital, dan perpustakaan *bookless* (Harahap, 2018).

2.1.2 Android

Android adalah sistem operasi berbasis Linux yang dirancang khusus untuk perangkat dengan layar sentuh seperti *smart phone* (Fu'ad, Kholil, & Wardhani, 2019). Android awalnya dikembangkan oleh Android, Inc pada tahun 2003 oleh Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Namun pada tahun 2005 Google membeli perusahaan ini dan menjadikannya sebagai anak perusahaan. Sejak tahun 2008 Android secara bertahap melakukan pengembangan untuk meningkatkan kinerja dari versi sebelumnya. Di tahun 2009 Google resmi merilis produk android versi 1.5 yaitu Cupcake pada bulan April, versi 1.6 Donut pada September, dan 2.0-2.1 Éclair pada Oktober. Uniknya nama setiap versi yang rilis merupakan nama makanan pencuci mulut yang manis secara alfabetis. Sampai saat ini versi terbaru dari android dirilis pada tanggal 3 September 2019 (Wikipedia^[1], 2020). Berbeda dengan versi sebelumnya, penamaan pada versi ini tidak lagi menggunakan nama makanan. Nama android versi 10 adalah Android Q dan logo android 10 dapat dilihat pada gambar 2.1.



Gambar 2.1 Logo android Q

Android merupakan salah satu sistem operasi yang populer digunakan produsen untuk peranti yang mereka produksi. Hal ini dikarenakan fitur yang dimiliki android memiliki tampilan yang menarik dan dapat digunakan sebagai alat multimedia. Selain itu android lebih mudah dikembangkan oleh para *developers* karena *source code* dari android bersifat *open source* (Dicoding Indonesia, 2017). Bahasa pemrograman yang digunakan untuk mengembangkan aplikasi android adalah Java, C++, dan Kotlin.

Pada umumnya IDE (*Integrated Development Environment*) yang digunakan untuk membuat aplikasi android adalah Android Studio yang berbasis IntelliJ IDEA. Android studio adalah IDE yang dikeluarkan oleh Google. Beberapa fitur yang ada pada Android studio yaitu *gradle*, *code completion*, dan *firebase*. Fitur-fitur ini dapat membantu *developer* dalam proses pembuatan aplikasi. Android Studio menggunakan bahasa pemrograman java dan kotlin (Dicoding Indonesia, 2017).

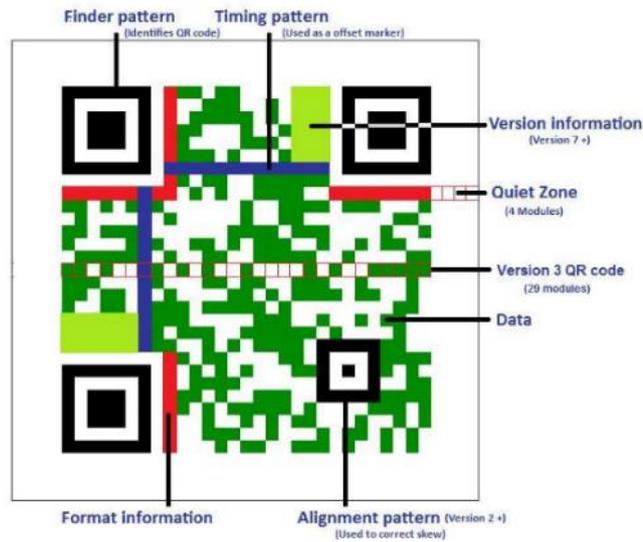
2.1.3 QR Code

QR (Quick response) code pertama kali dipublikasikan di Jepang pada tahun 1994 oleh perusahaan Denso Wave (Fu'ad, Kholil, & Wardhani, 2019). *QR code* semakin dikenal dan semakin sering digunakan. Awalnya *QR code* hanya digunakan untuk melacak kendaraan bagian di manufaktur. Namun saat ini *QR code* sudah digunakan untuk kepentingan komersial, pendidikan, dan kepentingan umum. Hal ini karena kemampuan *QR code* yang dapat menyampaikan informasi dan mendapat respon dengan cepat.

Gambar 2.2 menunjukkan bagian-bagian dari struktur *QR code* (Clivan, Sugiarto, & Sinsuw, 2019). Penjelasan bagian-bagian tersebut yaitu

- a. *Finder pattern* adalah tiga kotak kecil berwarna hitam di ketiga ujung *QR code* berfungsi sebagai identifikasi letak *QR code*. Dengan begitu *QR code* tetap dapat dibaca dari berbagai arah selama masih 360 derajat.
- b. *Timing pattern* berfungsi untuk mengidentifikasi koordinat pusat.
- c. *Version information* menunjukkan versi dari *QR code*.
- d. *Quiet zone* berfungsi untuk mempermudah pengenalan *QR code* oleh sensor CCD.
- e. *QR code version* adalah versi dari *QR code* yang digunakan.
- f. *Data* berfungsi untuk menyimpan data yang dikodekan.
- g. *Alignment pattern* berfungsi untuk memperbaiki penyimpangan *QR code* terutama distorsi nonlinier.

- h. *Format information* berfungsi sebagai informasi tentang *error correction level* dan *mask pattern*.



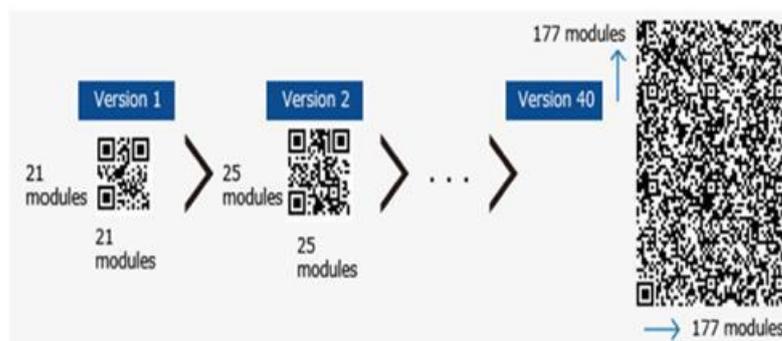
Gambar 2.2 Struktur QR Code

QR code adalah kode matriks dua dimensi sehingga dapat menyimpan informasi secara vertikal dan horizontal (Widayati, 2017). Itu sebabnya *QR code* memiliki kemampuan menyimpan data lebih banyak daripada *barcode*. Selain itu *QR code* dapat menyimpan jenis data angka, huruf, kanji, kana, hiragana, kode biner, dan simbol. Sedangkan *barcode* hanya dapat menyimpan huruf dan angka. Meskipun memiliki kapasitas penyimpanan yang besar namun ukuran tampilan *QR code* lebih kecil daripada *barcode*. *QR code* juga lebih tahan terhadap kerusakan. Walaupun sebagian simbol kotor atau rusak *QR code* masih dapat dibaca. Hal ini karena *QR code* mampu memperbaiki kesalahan sampai 30%.

Kapasitas penyimpanan data yang dapat disimpan bergantung pada versi *QR code*, ukuran *QR code*, dan tingkat *Error Correction Capability (ECC)* (Widayati, 2017). *Error Correction Capability* menunjukkan batas *QR code* masih dapat terbaca ketika *QR code* rusak. *QR code* dapat menghasilkan 40 versi yang berbeda dari versi 1 (21 x 21 modul) sampai versi 40 (177 x 177 modul). Jarak setiap versi adalah 4 modul. Modul mengacu pada titik hitam dan putih yang membentuk *QR code*. Semakin banyak data yang disimpan maka semakin besar modul *QR code* yang diperlukan. Tabel 2.1 menunjukkan jumlah data yang dapat dimuat *QR code* (versi 1 – 3). Gambar 2.3 menunjukkan beberapa versi *QR code*.

Tabel 2.1 Jumlah data yang dimuat QR Code

Versi QR Code	Modul	Tingkat Error Correction Capability	Jumlah Data (bits)
1	21x21	L	152
		M	128
		Q	104
		H	72
2	25x25	L	272
		M	224
		Q	176
		H	128
3	29x29	L	440
		M	352
		Q	272
		H	208



Gambar 2.3 Versi QR Code

QR code memiliki lima jenis (Widayati, 2017) yaitu

a. QR code Model 1 dan Model 2

QR code model 1 dapat menampung 1.167 angka dengan versi maksimum 14 (73 x 73 modul). QR code model 2 dapat menampung 7.089 angka dengan versi maksimum 40 (177 x 177 modul).

b. Micro QR code

Micro QR code dapat menyimpan 35 angka dengan versi maksimum M4 (17 x 17 modul). Micro QR hanya membutuhkan dua modul margin dan satu pola deteksi, sementara QR code biasa

membutuhkan tiga sudut simpol untuk pola deteksi posisi dan empat modul margin.

c. *iQR Code*

iQR code dapat menampung 40.000 angka dengan versi maksimum 61 (422 x 422 modul). Kode yang dihasilkan dapat berbentuk persegi atau persegi Panjang.

d. *SQRC*

SQRC dapat digunakan untuk menyimpan informasi pribadi yang digunakan untuk mengelola informasi internal perusahaan karena dilengkapi dengan fungsi pembatas.

e. *LogoQ*

LogoQ menggunakan logika *Since proprietary* pribadi yang dapat menggabungkan fitur desain tingkat tinggi.

QR code mudah digunakan karena tidak perlu menggunakan perangkat khusus *scan QR code*. *QR code* dapat dibaca menggunakan *smartphone* yang memiliki aplikasi pembaca *QR code*. Pengguna hanya harus mengaktifkan aplikasi pembaca *QR code* dan mengarahkan kamera ke *QR code* yang tersedia. Secara otomatis data yang ada di *QR code* akan ditampilkan pada layar *smartphone*. Dengan demikian informasi yang ada di *QR code* dapat dijangkau secara efektif dan efisien.

Pembuatan *QR code* juga terbilang mudah. Saat ini telah tersedia banyak aplikasi untuk membuat *QR code*. Pengguna dapat memasukkan logo, klip video, atau foto pada desain *QR code*. Hal ini tidak akan menghilangkan informasi yang akan di masukkan dalam *QR code*. Pengguna juga dapat membuat *QR code* berwarna sesuai dengan keinginan mereka. Karena kelebihan *QR code* diatas dibandingkan dengan *barcode* penulis lebih memilih untuk memanfaatkan *QR code*.

2.1.4 NoSQL

Konsep NoSQL muncul pada tahun 1998 ketika Carlo Strozzi menggunakannya untuk nama *open source database relational* yang tidak mengikuti standar SQL. Pada tahun 2009, Eric Evans memperkenalkan kembali istilah NoSQL ketika Johan Oskarsson dari Last.fm menyelenggarakan acara yang membahas tentang *distributed database open source*. Konsep NoSQL ada karena muncul berbagai kesulitan dalam penanganan perkembangan database menggunakan konsep *relational database*. Saat ini database tidak hanya berkembang secara vertikal (penambahan baris), tetapi juga berkembang secara horizontal (penambahan kolom). Namun dalam konsep RDBMS penambahan kolom merupakan masalah yang harus dihindari. Sehingga diperlukan konsep NoSQL yang tidak membutuhkan skema tabel untuk menghindari operasi *join* karena data berkembang secara horizontal (Pitri, 2019).

Perbedaan gambaran antara SQL dan NoSQL dapat dilihat dari beberapa sisi yaitu bahasa, skalabilitas dan struktur (Zulhimar, 2019). Dari segi bahasa SQL menggunakan bahasa kueri terstruktur untuk mendefinisikan dan memanipulasi data. Hal ini tentu saja memberikan kemudahan untuk mengubah satu kueri ke kueri lainnya karena aman dan baik jika kueri tersebut kompleks. SQL mengharuskan kita untuk mendefinisikan skema struktur data dan semua data harus mengikuti struktur yang telah didefinisikan. Sedangkan NoSQL memiliki skema yang dinamis dan fleksibel. Hal ini dikarenakan kita dapat membuat dokumen tanpa harus memperbaiki strukturnya karena setiap dokumen memiliki strukturnya sendiri. Sintaks dari database ke database lain bisa saja berbeda.

Berdasarkan skalabilitas SQL hanya dapat ditambah secara vertikal. Sedangkan NoSQL dapat ditambahkan secara vertikal maupun horizontal. Sehingga NoSQL tepat digunakan apabila datanya berukuran besar dan terus berubah. Struktur database SQL adalah tabel sedangkan NoSQL dapat berupa pasangan kunci-nilai, penyimpanan kolom, data grafik, atau dokumen. Perbedaan antara SQL dan NoSQL (Schaefer) dapat di lihat jelas pada tabel 2.2.

Tabel 2.2 Perbedaan SQL dan NoSQL

	SQL	NoSQL
Model Penyimpanan Data	Tabel (baris dan kolom)	Dokumen, nilai-kunci, grafik
Sejarah Pengembangan	Dikembangkan dengan berfokus untuk pengurangan duplikasi data	Dikembangkan dengan berfokus pada penambahan data secara horizontal dan untuk data yang sering berubah-ubah
Skema	Kaku	Fleksibel
Transaksi ACID multi-record	Mendukung	Kebanyakan tidak mendukung transaksi ACID multi-record
Data ke Pemetaan Objek	Membutuhkan ORM (Pemetaan relasional objek)	Kebanyakan tidak membutuhkan ORM

Kelebihan NoSQL adalah dapat menampung data yang terstruktur, semi terstruktur, dan tidak terstruktur secara efisien dalam skala besar. Hal ini karena NoSQL tidak mengenal skema tabel yang kaku. NoSQL dapat menggunakan OOP dalam manipulasi data. Jika NoSQL dijalankan di *multiple server* maka data akan tersebar secara otomatis dan merata keseluruh server.

Dalam database SQL data biasanya dinormalisasi sehingga kueri untuk satu objek atau entitas mengharuskan kita untuk menggabungkan data dari beberapa tabel. Ketika banyak tabel digabungkan maka akan membutuhkan operasi *join*. Namun dalam konsep NoSQL kueri bisa saja tidak membutuhkan operasi *join*, sehingga untuk menggabungkan tabel menjadi lebih mudah.

Kekurangan NoSQL adalah tidak mendukung transaksi ACID (*Atomacity, Consistency, Isolation, and Durability*) di beberapa dokumen, padahal beberapa aplikasi masih membutuhkan ACID di beberapa *record*. NoSQL lebih

menghabiskan banyak penyimpanan daripada SQL. Karena konsep NoSQL lebih diperhatikan kueri bukan untuk duplikasi data. Namun hal ini bukan merupakan masalah yang besar karena saat ini penyimpanan sangat murah.

Konsep NoSQL yang digunakan untuk model data spesifik dan memiliki skema yang fleksibel untuk membuat sebuah aplikasi. NoSQL berbeda dengan database SQL. Database SQL membuat data menjadi tabel yang terdiri dari baris dan kolom dimana tabel yang satu memiliki relasi dengan tabel yang lain. Sedangkan konsep NoSQL memiliki beragam model data seperti nilai kunci, dokumen, dan grafik. Jenis NoSQL berdasarkan model penyimpanan datanya (Amazon) yaitu

a. Nilai-kunci (*Value-key*)

Database ini disimpan dalam pasangan kunci dan nilai yang digunakan untuk *associative array*. Database nilai-kunci dapat dipartisi dan memungkinkan pengembangan horizontal pada skala yang tidak dapat dicapai oleh jenis database lain seperti penggunaan pada aplikasi *gaming* dan IoT. Kelebihannya adalah meskipun data dalam jumlah besar namun kueri pencariannya sederhana.

b. Dokumen

Data sering diwakili sebagai sebuah objek dokumen seperti JSON sehingga lebih mudah dikembangkan. Kelebihan database dokumen adalah mudah dikembangkan untuk menyimpan dan membuat kueri data dalam database. Sifat jenis database ini fleksibel, semi terstruktur, dan hierarkis. Jenis database ini bisa memanfaatkan pasangan *key-value* untuk mengakses dan menyimpan data.

c. Grafik

Database ini menyimpan data dalam *node-node* yang saling terhubung dengan *edge*. Database grafik bertujuan agar dapat membuat dan menjalankan aplikasi dengan dataset yang selalu terhubung jadi lebih mudah. Selain itu jenis penyimpanan data grafik dapat menganalisis hubungan anatra data yang terhubung. Beberapa contoh penerapannya

adalah jaringan sosial media, mesin rekomendasi, dan pendeteksian penipuan.

d. Object Database

Format database ini disimpan dalam objek-objek yang sama pengertiannya dengan objek dipemrograman berorientasi objek.

Konsep NoSQL sangat cocok digunakan untuk aplikasi yang memerlukan database yang fleksibel, dapat diskalakan, berkinerja tinggi, dan memiliki fungsionalitas tinggi. Sedangkan database SQL permintaan untuk menyimpan dan mengambil data dikomunikasikan menggunakan kueri yang sesuai dengan bahasa kueri terstruktur. Konsep NoSQL terbukti unggul dalam proses transaksi CRUD daripada database SQL. Namun database SQL lebih unggul dalam kemampuan untuk menghadapi transaksi *join/agregasi* (Bhaswara, Riyanarto, & Sunaryono, 2017).

2.1.5 Cloud Firestore

Cloud Firestore adalah *real time database* yang mendukung penyimpanan data dengan konsep No-SQL. *Firestore* merupakan salah satu solusi database yang *Firebase* tawarkan untuk pengembangan aplikasi seluler. *Firestore* juga dapat digunakan secara *offline* untuk seluler, iOS, dan situs web. Pembuatan kueri di *Cloud Firestore* bersifat efisien, ekspresif, dan fleksibel (Firestore^[1], 2019).

Akses data dapat dilindungi menggunakan aturan keamanan *Cloud Firestore* atau Pengelolaan Akses dan Identitas (IAM). Aturan keamanan ini lebih kuat dari yang ada pada *Firebase Realtime Database*. Jika dibandingkan dengan *Firebase Realtime Database*, *Firestore* memiliki fitur kueri yang lebih lengkap dan cepat dan fitur skala yang lebih mendalam.

Berbeda dengan *Firebase Realtime Database* yang memiliki struktur *parent* dan *child*, *Firestore* memiliki struktur yang hirarki dan fleksibel (Firestore^[2], 2019). Data akan disimpan dalam dokumen yang tersusun menjadi koleksi. Dokumen dalam koleksi memiliki nama yang bersifat unik. Setiap dokumen berisi pasangan kunci dan nilai. Gambar 2.4 merupakan ilustrasi dari struktur data *cloud firestore*. Jenis data yang disimpan dapat berupa string, angka, boolean, titik

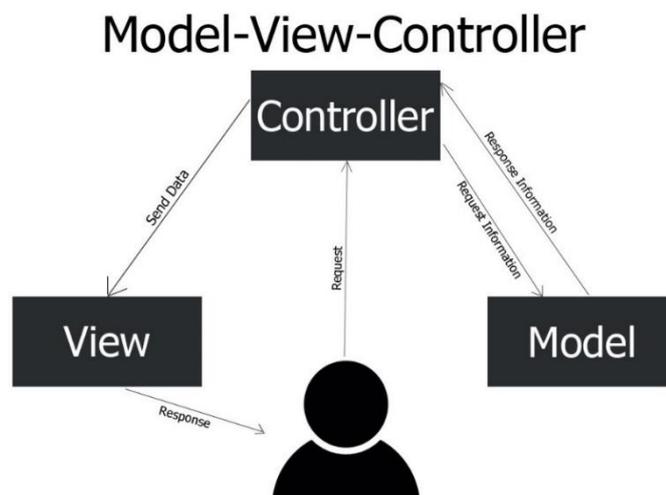
geografis, array, hingga objek yang kompleks dan bertingkat. Ukuran data yang dapat dimuat dalam satu dokumen tidak lebih dari satu MB (Firebase^[2], 2019). Itu sebabnya *Firestore* baik untuk menyimpan banyak data berukuran kecil. Koleksi merupakan wadah untuk menyimpan beberapa dokumen. Koleksi tidak dapat langsung berisi data. Dokumen dapat memuat koleksi.



Gambar 2.4 Struktur data cloud firestore

2.1.6 Konsep Model-View-Controller (MVC)

MVC adalah sebuah konsep yang diperkenalkan oleh Trygve Reenskaug untuk mengenkapsulasi data Bersama dengan pemrosesan (model), mengisolasi dari proses manipulasi (*controller*), dan tampilan (*view*) (Diana & Febrianti, 2017). Konsep *Model-View-Controller* adalah sebuah *design pattern* atau konsep untuk membuat sebuah aplikasi dengan memisahkan data (*model*) dari tampilan (*view*) dan proses (*controller*). MVC juga merupakan implementasi dari konsep OOP (Object Oriented Programming) (Wikipedia^[2], 2020). Konsep MVC biasanya digunakan pada pemrograman Java, Ruby, swift, dan PHP. Gambar 2.5 menunjukkan konsep MVC.



Gambar 2.5 Konsep Model-View-Controller (MVC)

1. Model

Model berisi data dan fungsi-fungsi yang akan mengatur proses data. Model bertugas untuk mengatur, menyiapkan, memanipulasi, dan mengorganisir data. Model bekerja berdasarkan instruksi dari *controller*.

2. View

View merupakan bagian yang mengatur tampilan ke pengguna atau merupakan *output* dari suatu aplikasi (*interface*).

3. Controller

Controller merupakan bagian yang menghubungkan *model* dan *view*. *Controller* berfungsi untuk menerima masukan dari pengguna. Setelah menerima masukan dari pengguna *controller* akan mendistribusikan data ke *model* dan *view* untuk di proses berdasarkan masukan tersebut.

Dengan menggunakan metode MVC aplikasi dapat dikembangkan secara terpisah dengan antarmuka pengguna. Keuntungan utama MVC adalah komponen antarmuka (*user interface*) dapat digunakan berulang kali (*reusable*), sehingga jika ada penambahan modul dalam aplikasi dapat dilakukan tanpa harus membangun kembali aplikasi (Diana & Febrianti, 2017). Implementasi pada aplikasi juga lebih sederhana. Konsep MVC lebih mudah dipelihara oleh tim pengembang berdasarkan spesifikasi pekerjaannya.

2.1.7 UML (Unified Modeling Language)

UML adalah alat yang digunakan untuk menganalisis kebutuhan fungsionalitas terhadap sistem berbasis objek yang akan dibangun (Codepolitan, 2016). UML bermanfaat sebagai dokumentasi agar mempermudah pengembangan aplikasi kedepannya. UML memiliki dua jenis diagram yaitu *Structural Diagram* dan *Behavioral Diagram* (Codepolitan, 2016). *Structural Diagram* diantaranya adalah *class diagram*, *object diagram*, *component diagram*, dan *deployment diagram*. *Behavioral diagram* yaitu *use case diagram*, *sequence diagram*, *collaboration diagram*, *statechart diagram*, dan *activity diagram*.

Usecase diagram terdiri dari tiga komponen yaitu *usecase*, aktor, dan sistem (Hutauruk, 2019). *Usecase* adalah fungsionalitas yang disediakan oleh sistem.

Aktor adalah sistem atau orang yang berinteraksi dengan *usecase*. Sistem menyatakan batas antara aktor yang menggunakan *usecase* dan fitur yang ada didalam sistem. *Usecase diagram* memiliki tiga relasi yaitu asosiasi, generalisasi, dan dependensi. Asosiasi adalah komunikasi antara aktor dan *usecase*. Generalisasi adalah relasi antara dua aktor atau dua *usecase* dimana salah satunya merupakan tambahan dari yang lain. Dependensi dibagi menjadi dua yaitu *extend* dan *include*. *Extend* merupakan relasi *usecase* tambahan ke sebuah *usecase* dimana *usecase* yang ditambahkan dapat berdiri sendiri walaupun tanpa *usecase* tambahan tersebut. *Include* adalah relasi *usecase* tambahan ke sebuah *usecase* dimana *usecase* yang ditambahkan memerlukan *usecase* tersebut untuk menjalankan fungsinya.

Activity diagram menggambarkan aliran kontrol sistem untuk melihat bagaimana sistem tersebut bekerja ketika dijalankan (Rizky, 2019). Ada beberapa komponen *activity diagram* yaitu *initial state*, *final state*, *activity*, *decision*, *merge*, *transition/association*, dan *synchronization*. *Initial state* adalah tanda bahwa aliran kerja *activity diagram* dimulai. *Final state* adalah bagian akhir dari suatu aliran kerja *activity diagram*. *Activity* merupakan pekerjaan yang dilakukan dalam aliran kerja. *Decision* berfungsi untuk menggambarkan pilihan kondisi yang dapat mengalir ke lebih dari satu jalur. *Merge* berfungsi untuk menggabungkan kembali aliran kerja yang dipecahkan oleh *decision*. *Transition/association* menghubungkan aktivitas selanjutnya dengan aktivitas sebelumnya. *Synchronization* terbagi dua yaitu *synchronization fork* untuk memecah behavior menjadi aktivitas paralel dan *synchronization join* untuk menggabungkan kembali aktivitas yang paralel.

2.2 State of the Art

Dalam penelitian yang dilakukan oleh Rasti Prathivi (2018) dijelaskan bahwa pengelolaan buku di perpustakaan memerlukan sistem yang cepat, akurat, dan harus mudah untuk diakses. *QR Code* dapat mendukung akses informasi yang cepat untuk mengetahui informasi buku. *QR code* memiliki kemampuan untuk menyimpan data dalam ukuran yang besar. Peneliti menggunakan *QR Code Generator* untuk mengkonversi data buku menjadi gambar QR. Selanjutnya

gambar QR akan dipasang di setiap buku. *QR code* menyimpan data ISBN, judul buku, pengarang, No. rak, No. lemari, dan informasi ketersediaan buku. Pengguna dapat membaca gambar QR dengan menggunakan *QR code Reader* yang di unduh gratis di Google Play.

Penelitian lain yang dilakukan untuk membuat sebuah aplikasi *website* pendataan perpustakaan dengan fitur *QR code* dilakukan oleh Tewuh Clivan (2019). Aplikasi ini dapat mempermudah pendataan buku yang di kumpul, dipinjam, dikembalikan, dan pendataan keanggotaan perpustakaan. Aplikasi *website* ini digunakan oleh petugas perpustakaan untuk melakukan pendataan buku, keanggotaan, dan transaksi peminjaman dan pengembalian buku. Fitur *QR code* digunakan petugas untuk mencari informasi data buku.

Dalam jurnal yang ditulis oleh Kolo (2018) dijelaskan bahwa peneliti akan membuat sistem manajemen perpustakaan berbasis android dan *website*. Sistem ini adalah sebuah perangkat lunak yang dapat memantau dan mengendalikan transaksi peminjaman buku di perpustakaan. Penggunaan aplikasi ini diharapkan dapat menghemat tenaga, meminimalisir terjadinya *human error*, dan dapat memberikan akses informasi dimana saja dan kapan saja oleh pengguna. Database yang digunakan adalah MySQL. Beberapa fitur yang tersedia adalah OPAC (Online Public Acces Catalog), perawatan, manajemen pencatatan, penyimpanan, pencarian berdasarkan judul, penulis, dll, dan keamanan dimana tidak sembarang orang dapat mengakses data dalam database. Aplikasi *website* digunakan oleh admin, sedangkan aplikasi android digunakan oleh pengguna.

Yana Henriana (2015) juga melakukan penelitian yang bertujuan untuk membuat aplikasi perpustakaan seluler berbasis android yang menyediakan layanan peminjaman secara online. Layanan ini dapat memudahkan pengguna untuk dapat meminjam buku tanpa menunggu dan mengantri. Aplikasi ini dapat digunakan sebagai media untuk mencari buku di perpustakaan dan melihat ketersediaan buku yang akan dipinjam. Peneliti menggunakan MySQL sebagai tempat penyimpanan data dan JSON sebagai penghubung antara database dan aplikasi android. Selain itu aplikasi ini membantu memantau status peminjaman dan tanggal pengembalian buku agar meminimalisir keterlambatan pengembalian

buku. Pengguna juga dapat memberikan kritik dan saran dengan mengisi formulir yang tersedia.

Penelitian untuk merancang dan membangun sistem informasi *Web AR* berbasis *QR code* untuk pengunjung di perpustakaan juga dilakukan oleh Imam Holeh (2016). Sistem informasi ini dirancang agar dapat memberikan informasi berdasarkan lokasi yang diberi *marker*. Penulis memanfaatkan kamera *smartphone* untuk mengakses informasi pada *QR code*. Metode pengembangan yang digunakan adalah metode pengembangan *software prototyping*. Hasil dari penelitian ini aplikasi dapat membantu kegiatan di perpustakaan dengan bantuan *marker* pada lokasi tertentu agar pengunjung mendapatkan data yang cepat dan akurat. Dalam penelitian ini penulis memberikan saran untuk membuat aplikasi Android sehingga tidak perlu menggunakan *browser*.

Dalam penelitian yang dilakukan oleh M. Nur Fu'ad (2019) melihat bahwa pengelolaan dan pelayanan perpustakaan masih dilakukan secara manual. Pengelolaan dan pelayanan yang dimaksud adalah transaksi peminjaman dan pengembalian buku. Penulis ingin membuat sebuah sistem yang tepat guna untuk meningkatkan pelayanan perpustakaan. Penulis memanfaatkan sistem operasi android dan teknologi *QR code*. Penulis membuat tampilan antar muka yaitu peminjaman dan pengembalian buku oleh *user* dan operasi transaksi yang dilakukan oleh petugas perpustakaan. Hasil penelitian membuktikan bahwa transaksi peminjaman dan pengembalian menggunakan aplikasi *QR code* lebih cepat daripada dilakukan secara manual.

2.3 Kerangka Konseptual

Pada sub bab ini dijelaskan kerangka konseptual dari penelitian.

Sistem perpustakaan konvensional perlu diperbaharui untuk memperbaiki mutu pelayanannya agar dapat membuat pengguna merasa nyaman dan memberi kemudahan dalam bertransaksi. Perbaikan mutu pelayanan juga dapat membantu pekerjaan pustakawan agar lebih cepat, efektif, dan efisien.



Banyak orang yang menggunakan *smartphone* karena bentuknya yang gampang untuk dibawa dan digunakan dimana saja. Sedangkan android merupakan sistem operasi yang banyak digunakan untuk *smartphone*.



QR (*Quick Respon*) Code memiliki kemampuan untuk menyimpan data dalam kapasitas yang besar. Selain itu *QR code* mudah dibuat. Hasil gambar *QR code* kecil sehingga mudah untuk disematkan di berbagai tempat atau benda. Selain itu hasil gambarnya terlihat lebih modern dan estetik. Akses untuk mendapat informasi yang ada di *QR code* juga mudah, hanya menggunakan kamera *smartphone*.



Pembaharuan yang dilakukan pada sistem perpustakaan konvensional adalah membuat sebuah sistem informasi perpustakaan dengan memanfaatkan android dan *QR code*. Aplikasi android dapat digunakan oleh pustakawan, admin, dan pengguna. Aplikasi ini di harapkan dapat membantu pustakawan untuk mendata buku dan mengelola transaksi peminjaman. Sedangkan untuk pengguna di harapkan dapat membantu dalam proses peminjaman dan pengembalian buku agar dapat menghemat waktu dan tenaga.