TUGAS AKHIR – KI141502

PENGGUNAAN *QR CODES* UNTUK MENCEGAH FRAUD PADA PROSES TRANSAKSI ANTARA PENYEDIA JASA TUTOR DENGAN KONSUMEN JASA TUTOR PADA APLIKASI FINDING – TUTOR BERBASIS ANDROID

I NYOMAN PANDE WAHYU DHARMAWAN
NRP 5113100002

Dosen Pembimbing I
Dr.tech.Ir.Raden Venantius Hari Ginardi,M.Sc.

Dosen Pembimbing II
-

Departemen Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

TUGAS AKHIR – KI141502

PENGGUNAAN *QR CODES* UNTUK MENCEGAH FRAUD PADA PROSES TRANSAKSI ANTARA PENYEDIA JASA TUTOR DENGAN KONSUMEN JASA TUTOR PADA APLIKASI FINDING – TUTOR BERBASIS ANDROID

I NYOMAN PANDE WAHYU DHARMAWAN
NRP 5113100002

Dosen Pembimbing I
Dr.tech.Ir.Raden Venantius Hari Ginardi,M.Sc.

Dosen Pembimbing II
-

Departemen Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017

(Halaman ini sengaja dikosongkan)

UNDERGRADUATE THESES – KI141502

FRAUD AVOIDANCE USING *QR CODES* ON TRANSACTION PROCESS ON FINDING – TUTOR APPLICATION IN ANDROID SYSTEM

I NYOMAN PANDE WAHYU DHARMAWAN
NRP 5113100002

First Advisor
Dr.tech.Ir.Raden Venantius Hari Ginardi,M.Sc.

Second Advisor
-

Department of Informatics
Faculty of Information Technology
Sepuluh Nopember Institute of Technology
Surabaya 2017

(Halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN

PENGGUNAAN QR CODES UNTUK MENDETEKSI FRAUD PADA PROSES TRANSAKSI ANTARA PENYEDIA JASA TUTOR DENGAN KONSUMEN JASA TUTOR PADA APLIKASI FINDING – TUTOR BERBASIS ANDROID

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Manajemen Informasi
Program Studi S-1 Departemen Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh:
I NYOMAN PANDE WAHYU DHARMAWAN
NRP: 5113 100 002

Disetujui oleh Pembimbing Tugas Akhir:

1. Dr.tech.Ir.Raden Venatius Hari Ginardi, M.Sc .....
(NIP. 196505181992031003) (Pembimbing 1)

KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
FAKULTAS TEKNOLOGI INFORMASI
DEPARTEMEN TEKNIK INFORMATIKA

SURABAYA
JUNI, 2017

(Halaman ini sengaja dikosongkan)

PENGGUNAAN *QR CODES* UNTUK MENCEGAH FRAUD PADA PROSES TRANSAKSI ANTARA PENYEDIA JASA TUTOR DENGAN KONSUMEN JASA TUTOR PADA APLIKASI FINDING – TUTOR BERBASIS ANDROID

Nama Mahasiswa	**:**	**I NYOMAN PANDE WAHYU DHARMAWAN**
NRP	**:**	**5113100002**
Jurusan	**:**	**Teknik Informatika FTIF-ITS**
Dosen Pembimbing 1	**:**	**Dr. tech.Ir. Raden Venatius Hari Ginardi, M.Sc.**
Dosen Pembimbing 2	**:**	**-**

Abstrak

Aplikasi Finding – Tutor merupakan aplikasi dengan basis Android. Aplikasi ini membantu masyarakat dalam mencari tutor – tutor yang dibutuhkan untuk muridnya. Disamping itu, Finding – Tutor juga membantu penyedia jasa tutor dalam melakukan pemasaran atas jasa yang dibuatnya. Aplikasi ini membantu murid untuk melakukan pemesanan transaksi yang sesuai dengan kriteria murid yang dibutuhkan. Penyedia jasa tutor juga dapat memilih transaksi sesuai dengan kemampuan yang dapat diajarkannya. Akan tetapi, aplikasi ini harus dapat menangani fraud yang terjadi yaitu kondisi dimana penyedia jasa tutor tidak datang ke lokasi transaksi dan merugikan murid. Hal ini dikarenakan apabila tidak ada fitur yang mendeteksi bahwa murid dan penyedia jasa tutor telah bertemu, penyedia jasa tutor dapat memberikan data palsu bahwa penyedia jasa tutor telah berada di lokasi transaksi. Untuk itu dibutuhkan fitur untuk menangani fraud tersebut, dan Aplikasi Finding – Tutor ini menggunakan fitu scan QR Codes. Cara kerja dari fitur ini adalah ketika transaksi diambil oleh penyedia jasa tutor maka transaksi data tersebut akan secara otomatis berubah menjadi QR Codes. Setelah itu, murid harus melakukan scan QR

Codes dari pemilik penyedia jasa tutor tersebut untuk memulai transaksi.

Hasil uji coba fitur ini menunjukkan bahwa fitur tersebut berjalan dengan baik. Fitur ini dapat secara otomatis merubah data transaksi menjadi QR Codes dan dapat dilakukan scan QR Codes. Meskipun demikian, pengembangan lebih lanjut masih dibutuhkan untuk mengimbangi berkembangnya teknologi, serta berubahnya kebutuhan pengguna terhadap aplikasi ini.

Kata Kunci: Android, Fraud, Perangkat Bergerak, QR Codes, Startup.

FRAUD AVOIDANCE USING *QR CODES* ON TRANSACTION PROCESS ON FINDING – TUTOR APPLICATION IN ANDROID SYSTEM

Student's Name	**: I NYOMAN PANDE WAHYU DHARMAWAN**
Student's ID	**: 5113100002**
Department	**: Teknik Informatika FTIF-ITS**
First Advisor	**: Dr. tech.Ir. Raden Venatius Hari Ginardi, M.Sc.**
Second Advisor	**: -**

Abstract

Finding – Tutor application is an app with Android basis. This application helps people in finding some tutors that are needed for students. In addition, Finding – Tutor also helps tutor service providers in marketing their services. This application helps the stundent to create a transaction order according with the students' criteria. The tutor provider can also choose transaction according to the ability he / she can teach. However, this application should be able to handle a fraud that occurs in a condition where the tutor provider services does not come to the location of the transaction. This condition happens because if there is no feature detecting that the student and tutor service provider has met, the tutor service provider can make a false data. This kind of false data is a tutor service provider give a data to system that the tutor service provider has been in the transaction location where the real condition is different. For that occasion, Finding Tutor application require a feature to handle this fraud. To do so, Finding – Tutor uses QR Codes scanner to handle this fraud. This feature works when the transaction is taken by the tutor service provider. Then the transaction data will automatically be generated into QR Codes. After that, the student must scan the QR Codes from tutor service provider to start the transaction.

The test results of this feature indicate that the feature is running well. This feature can automatically convert over transaction data into QR Codes and can scan the QR Codes itself. Nevertheless, further development is still needed to keep pace with the development of technology, as well as the chaning needs of users of this application.

Keywords: Android, Fraud, Mobile, QR Codes, Startup

KATA PENGANTAR

Puji syukur setinggi – setingginya saya ucapkan kepada Tuhan yang Maha Esa, yang telah memberikan berkat dan kelancaran sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul "Penggunaan *QR Codes* untuk Mendeteksi Fraud pada Proses Transaksi antara Penyedia Jasa Tutor dengan Konsumen Jasa Tutor pada Aplikasi Finding – Tutor Berbasis Android" dengan baik.

Terselesaikannya Tugas Akhir ini tidak terlepas dari bantuan dan dukungan banyak pihak, Oleh karena itu melalui lembar ini penulis ingin mengucapkan terima kasih dan penghormatan kepada:

1. Tuhan yang Maha Esa atas berkah yang tiada habisnya sehingga penulis dapat menyelesaikan Tugas Akhir ini dengan baik.
2. Ayah dan Ibu penulis, Nyoman Astawa dan Ayu yang tiada hentinya memberikan dukungan doa, moral, dan material kepada penulis sehingga penulis dapat menyelesaikan Tugas Akhir ini.
3. Kakak penulis, Eka Bayu dan Made Rossalita serta keluarga tercinta yang telah memberikan dukungan dan semangatnya kepada penulis. Terlebih kepada Made Rossalita terima kasih atas tab android yang diberikan. Sehingga penulis dapat melakukan uji coba Tugas Akhir ini.
4. Bapak Dr. Tech. Ir. Raden Venatius Hari Ginardi, M. Sc selaku pembimbing 1 yang telah membantu, membimbing, dan memotivasi penulis dalam menyelesaikan Tugas Akhir dengan baik dan penuh kesabaran.
5. Bapak Darlis Herumurti, S.Kom., M.Kom. selaku kepala jurusan Teknin Informatika ITS dan segenap dosen dan karyawan Teknik Informatika ITS yang telah memberikan ilmu dan pengalaman kepada penulis selama menjalani masa studi di Teknik Informatika ITS.

6. Sahabat penulis, adik – adik TC 2014, khususnya Annisa yang dengan baik telah memberikan dukungan moral, tenaga, dan waktu dan juga menjadi tempat yang baik untuk mencurahkan semua kesedihan selama proses pengerjaan Tugas Akhir ini.
7. Teman seperjuangan penulis di dalam kelompok besar Tugas Akhir dengan topik rmk MI ini. Teman – teman kelompok penulis, Riska Adhita, Syah Dia Putri, Rifqi Nur Fadhilah, Astidhita Nuraini Latifah, dan Luwandino Wismar yang telah saling membantu, menghibur, dan memberi semangat di dalam proses pelaksanaan Tugas Akhir hingga penyusunan buku ini.
8. Teman-teman TC 2013, kakak-kakak TC 2011 & 2012 dan yang tidak dapat disebutkan satu per satu yang selalu membantu, menghibur, menjadi tempat bertukar ilmu serta pembelajaran baru dan berjuang bersama-sama penulis.
9. Serta semua pihak yang yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa laporan Tugas Akhir ini masih memiliki banyak kekurangan. Oleh karena itu dengan segala kerendahan hati penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan penulis kedapannya. Selain itu, penulis berharap laporan Tugas Akhir ini dapat berguna bagi pembaca secara umum.

Surabaya, Juni 2017

I Nyoman Pande Wahyu Dharmawan

DAFTAR ISI

LEMBAR PENGESAHAN v
Abstrak vii
Abstract ix
KATA PENGANTAR xi
DAFTAR ISI xiii
DAFTAR GAMBAR xvii
DAFTAR TABEL xix
DAFTAR KODE SUMBER xxi
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 2
1.3 Batasan Permasalahan 2
1.4 Tujuan 3
1.5 Manfaat 3
1.6 Metodologi 3
1.6.1 Penyusunan Proposal Tugas Akhir 3
1.6.2 Studi Literatur 4
1.6.3 Implementasi Perangkat Lunak 4
1.6.4 Pengujian dan Evaluasi 4
1.6.5 Penyusunan Buku 4
1.7 Sistematika Penulisan Laporan 5
BAB II TINJAUAN PUSTAKA 7
2.1 Penelitian Terkait 7
2.2 *QR Codes* 8
2.2.1 *QR Codes Reader* 9
2.2.2 *QR Codes Generator* 10
2.2.3 Perbandingan antara *QR Codes* dengan Teknologi Sejenis 13
2.3 Android 14
2.3.1 *Volley Library* 15
2.3.2 *Zxing Library* 15
2.4 *CodeIgniter Framework* 15
2.5 *MySQL Database* 16

2.6 Pemrograman Java 16
2.7 Startup 17
BAB III PERANCANGAN SISTEM 19
3.1 Analisis 19
3.1.1 Deskripsi Umum Aplikasi 19
3.1.2 Analisis Kebutuhan Sistem 25
3.1.3 Analisis Aktor 26
3.1.4 Kasus Penggunaan 26
A. Mengambil Transaksi (UC-0001) 27
B. Melakukan Scan QR Codes (UC-0003) 29
C. Membatalkan Transaksi (UC-0004) 31
3.2 Perancangan Sistem 34
3.2.1 Perancangan Basis Data 35
3.2.2 Perancangan Tampilan Antarmuka 37
A. Rancangan Antarmuka Halaman Menampilkan QR Code yang telah di generate 38
B. Rancangan Antarmuka Halaman Scan QR Code 39
C. Rancangan Antarmuka Halaman Batal Transaksi 40
3.2.3 Perancangan Arsitektur Aplikasi 41
BAB IV IMPLEMENTASI 43
4.1 Lingkungan Implementasi 43
4.2 Implementasi 44
4.2.1 Implementasi Pembuatan *QR Codes* 44
A. Implementasi Kode Program dalam Melakukan *Generate QR Codes* pada Android Studio 44
B. Implementasi Kode Program *Backend* Menggunakan CodeIgniter 51
C. Implementasi Tampilan Antarmuka Fitur *Generate QR Codes* 57
4.2.2 Implementasi *Scan QR Codes* 62
A. Implementasi Kode Program *Scan QR Codes* pada Android Studio 63
B. Implementasi Kode Program *Backend* pada Fitur *Scan QR Codes* 69

C. Implementasi Tampilan Antarmuka Fitur *Scan QR Codes* 73
4.2.3 Implementasi Pembatalan Transaksi 76
A. Implementasi Kode Program Pembatalan Transaksi pada Android Studio 76
B. Implementasi Kode Program *Backend* pada Fitur Pembatalan Transaksi 83
C. Implementasi Tampilan Antarmuka Fitur Pembatalan Transaksi 88
BAB V UJI COBA DAN EVALUASI 95
5.1 Lingkungan Pengujian 95
5.2 Skenario Pengujian Fungsionalitas 96
5.2.1 Pengujian Pembuatan *QR Codes* 96
5.2.2 Pengujian *Scan QR Codes* 97
5.2.3 Pengujian Pembatalan Transaksi 99
5.3 Skenario Pengujian Kehandalan *QR Codes* 101
BAB VI KESIMPULAN DAN SARAN 105
6.1 Kesimpulan 105
6.2 Saran 106
DAFTAR PUSTAKA 107
LAMPIRAN 109
BIODATA PENULIS 119

(Halaman ini sengaja dikosongkan)

DAFTAR GAMBAR

Gambar 2.1 Contoh QR Codes.....8
Gambar2.2 Struktur QR Codes [2].....9
Gambar 2.3 Diagram Alir Proses Pembacaan QR Codes [1].....10
Gambar 2.4 Diagram Alir Proses Pembuatan QR Codes [1] 11
Gambar 3.1 Flowchart Keseluruhan Murid Mencari Tutor.....21
Gambar 3.2Flowchart Keseluruhan Tutor Mencari Murid.....22
Gambar 3.3 Diagram Alir Generate QR Codes.....24
Gambar 3.4 Diagram Alir Scan QR Codes 24
Gambar 3.5 Diagram Kasus Penggunaan.....27
Gambar 3.6 Diagram Aktivitas Mengambil Transaksi.....28
Gambar 3.7 Diagram Aktivitas Melakukan Scan QR Codes 31
Gambar 3.8 Diagram Aktivitas Membatalkan Transaksi 34
Gambar 3.9 Conceptual Data Model.....36
Gambar 3.10 Physical Data Model.....37
Gambar 3.11 Rancangan Antarmuka Halaman Detil Transaksi..39
Gambar 3.12 Rancangan Antarmuka Halaman QR Codes.....39
Gambar 3.13 Rancangan Antarmuka Halaman Scan QR Codes.40
Gambar 3.14 Rancangan Antarmuka Halaman Durasi Transaksi 40
Gambar 3.15 Rancangan Antarmuka Halaman Pembatalan Transaksi Penyedia Jasa Tutor 41
Gambar 3.16 Rancangan Antarmuka Halaman Pembatalan Transaksi Murid 41
Gambar 3.17 Arsitektur Sistem.....42
Gambar 4.1 List Transaksi yang Tersedia.....58
Gambar 4.2 Detil Transaksi.....59
Gambar 4.3 QR Codes yang telah di Ubah Secara Otomatis oleh Sistem.....60
Gambar 4.4 QR Codes Gagal di – Scan 61
Gambar 4.5 Transaksi Masih Berjalan 62
Gambar 4.6 Detil Penyedia Jasa Tutor.....74
Gambar 4.7 Proses Scan QR Codes 75
Gambar 4.8 Countdown Durasi Transaksi 75

Gambar 4.9 Penyedia Jasa Tutor Menekan Tombol Pembatalan Transaksi 89
Gambar 4.10 Notifikasi Penyedia Jasa Tutor Membatalkan Transaksi 90
Gambar 4.11 Notifikasi Punish 91
Gambar 4.12 Murid Membatalkan Transaksi..... 92
Gambar 4.13 Notifikasi pada Halaman Penyedia Jasa Tutor 93
Gambar 4.14 Murid Terkena Punish 94

DAFTAR TABEL

Tabel 2.1 Contoh Data yang telah di Encode 12
Tabel 2.2 Perbedaan QR Code dengan Barcode dan RFID [3] ... 13
Tabel 3.1 Daftar Kebutuhan Fungsional Sistem..... 25
Tabel 3.2 Daftar Kode Kasus Penggunaan..... 26
Tabel 3.3 Spesifikasi Kasus Penggunaan Mengambil Transaksi 27
Tabel 3.4 Spesifikasi Kasus Penggunaan Melakukan Scan QR Codes 29
Tabel 3.5 Spesifikasi Kasus Pengguna Membatalkan Transaksi 32
Tabel 4.1 Spesifikasi lingkungan implementasi 43
Tabel 5.1 Spesifikasi Lingkungan Uji Coba..... 95
Tabel 5.2 Skenario Uji Coba Pembuatan QR Codes 96
Tabel 5.3 Skenario Uji Coba Scan QR Codes 98
Tabel 5.4 Skenario Uji Coba Pembatalan Transaksi 99
Tabel 5.5 Hasil Uji Coba Kehandalan dalam Scan QR Codes . 101
Tabel A.1 Uji Coba 1 109
Tabel A.2 Uji Coba 2 111
Tabel A.3 Uji Coba 3 112
Tabel A.4 Uji Coba 4 114
Tabel A.5 Uji Coba 5 116

(Halaman ini sengaja dikosongkan)

DAFTAR KODE SUMBER

Kode Sumber 4.1 Penambahan ZXING library ke dalam build.gradle Aplikasi 45
Kode Sumber 4.2 Data Persiapan untuk diubah Menjadi QR Codes 46
Kode Sumber 4.3 Pemindahan Data dari Variabel QR_CODES ke Variabel S 46
Kode Sumber 4.4 Membuat QR Codes 47
Kode Sumber 4.5 Countdown Durasi QR Codes 48
Kode Sumber 4.6 Validasi Durasi QR Codes 50
Kode Sumber 4.7 Input Data Transaksi ke Database 52
Kode Sumber 4.8 Data yang ditampung dari Android Studio ke Controller 53
Kode Sumber 4.9 Menambahkan Library Volley 53
Kode Sumber 4.10 Model dalam Mengambil Data di Database Sesuai QR Codes Penyedia Jasa Tutor 54
Kode Sumber 4.11Controller dalam Mengirimkan Nilai Status . 55
Kode Sumber 4.12 Penginisialisasian Variabel untuk Menunjuk Halaman Controller 56
Kode Sumber 4.13 Pengaplikasian Volley pada Pembuatan QR Codes 57
Kode Sumber 4.14 Inisialisasi Fungsi QR Codes 64
Kode Sumber 4.15 Kondisi Ketika Melakukan Scan 65
Kode Sumber 4.16 Kondisi Status Transaksi 66
Kode Sumber 4.17 Inisialisasi Variabel untuk Menyimpan Durasi Transaksi 66
Kode Sumber 4.18 Tahapan Sintaks dalam Melakukan Perhitungan Mundur 68
Kode Sumber 4.19 Update Status Transaksi 69
Kode Sumber 4.20 Mengambil Durasi Transaski 70
Kode Sumber 4.21 Controller Pengolahan Data Transaksi dari Model 71
Kode Sumber 4.22 Konfigurasi Memanggil Controller pada Android Studio 72

Kode Sumber 4.23 Implementasi Pemakaian Backend pada Android Studio untuk Mendapatkan Nilai Status dan Durasi 73
Kode Sumber 4.24 Pembatalan Transaksi.....77
Kode Sumber 4.25 Autentifikasi Pembatalan Transaksi.....79
Kode Sumber 4.26 Notifikasi pada Halaman Murid.....80
Kode Sumber 4.27 Murid Membatalkan Transaksi 82
Kode Sumber 4.28 Notifikasi pada Halaman Penyedia Jasa Tutor 83
Kode Sumber 4.29 Merubah Status Menjadi Cancel pada Database 84
Kode Sumber 4.30 Kondisi Penyedia Jasa Tutor Membatalkan Transaksi 85
Kode Sumber 4.31 Mengembalikan Transaksi ke dalam List.....86
Kode Sumber 4.32 Menghapus Transaksi dari Database.....86
Kode Sumber 4.33 Mengambil Data Transaksi dari Database....87
Kode Sumber 4.34 Pengambilan Data Setiap 5 Detik Sekali.....88

(Halaman sengaja dikosongkan)

(Halaman sengaja dikosongkan)

BAB I
PENDAHULUAN

1.1 Latar Belakang

Teknologi informasi memberikan pengaruh yang sangat besar pada kehidupan manusia hamper di setiap bidang kehidupan. Khususnya pada bidang penyediaan jasa dengan pemanfaatan teknologi informasi. Banyak bisnis – bisnis *startup* bermunculan dengan menawarkan suatu jasa yang dibungkus dengan penggunaan teknologi aplikasi berbasis *mobile*. Namun, bisnis yang mengandalkan aplikasi *mobile* ini masih banyak yang belum memiliki sistem penanganan kecurangan dari sistem aplikasi yang dibuat. Salah satu contohnya adalah aplikasi Finding – Tutor. Salah satu fraud yang dapat terjadi adalah memperdayai sistem dengan tidak bertemu ataupun bertatap muka antara penyedia jasa tutor dan pengguna jasa tutor. Namun, sistem membaca bahwa penyedia jasa tutor tersebut telah bertemu dan telah melakukan tutor. Sehingga setelah itu pengguna jasa tutor dapat memberikan rating palsu atau bohong untuk meningkatkan nilai rating penyedia jasa tersebut. Untuk itu perlu adanya fitur tambahan yang dapat menangani kecurangan atau penyalahgunaan dari sistem aplikasi tersebut. Penulis menambahkan fitur verifikasi pada aplikasi Finding – Tutor ini dengan menggunakan teknologi *QR Codes*.

Sementara itu, penggunaan *QR Codes* pada era ini sudah tidak asing lagi di mata masyarakat umum. Hal ini memudahkan pengguna *QR Codes* untuk menyimpan data yang dibutuhkan dalam bentuk image dua dimensi. Cara kerja dari fitur verifikasi ini dibagi dalam dua tahap. Tahap pertama yang dilakukan adalah membuat sebuah *QR Codes* dari transaksi yang diambil. *QR Codes* ini akan digunakan oleh penyedia jasa tutor untuk memulai transaksi dengan murd. Kemudian pada tahap selanjutnya, murid harus melakukan *scan QR Codes* milik penyedia jasa tutor untuk memulai transaksi. Dengan adanya fitur ini, baik penyedia jasa tutor maupun murid harus bertemu secara langsung untuk memulai

transaksi. Transaksi akan dimulai dengan diawali oleh murid melakukan *scan QR Codes* milik penyedia jasa tutor.

1.2 Rumusan Masalah

Rumusan masalah yang terdapat pada Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana aplikasi dapat mengubah data berupa username dari penyedia jasa tutor dan id transaksi tutor yang diambil menjadi sebuah *QR Codes* yang nantinya akan digunakan pada saat proses validasi sebelum memulai kegiatan tutor?
2. Bagaimana aplikasi dapat melakukan scan pada *QR Codes* penyedia jasa tutor untuk menanggulangi fraud yang terjadi ketika penyedia jasa tutor tidak sebenarnya melakukan tutor?
3. Bagaimana *QR Codes* yang telah di – encode memiliki masa aktif selama satu jam?
4. Bagaimana aplikasi dapat melakukan scan pada *QR Codes* penyedia jasa tutor untuk memulai durasi dan menghentikan durasi pelaksanaan tutor?

1.3 Batasan Permasalahan

Batasan masalah yang terdapat pada Tugas Akhir ini adalah sebagai berikut:

1. *QR Codes* yang telah di – *encode* memiliki masa aktif pemakaian selama satu jam.
2. Aplikasi menggunakan kamera *handphone* untuk melakukan *scan QR Codes*.
3. Menggunakan perangkat lunak Android Studio versi 2.1 keatas.
4. Aplikasi dapat digunakan dengan perangkat Android versi 4.0 ke atas.
5. Pemakaian aplikasi pada lingkup daerah Surabaya.

1.4 Tujuan

Adapun tujuan dari pembuatan Tugas Akhir ini adalah :

1. Memastikan agar tidak terjadinya *fraud* berupa kebohongan yang dilakukan oleh penyedia jasa tutor dengan tidak secara langsung bertemu dengan konsumen jasa tutor untuk melakukan proses tutor.
2. Memastikan bahwa durasi proses tutor sesuai dengan durasi awal yang diminta oleh konsumen tutor.

1.5 Manfaat

Adapun manfaat dari pembuatan Tugas Akhir ini adalah sebagai berikut:

1. Menanggulangi *fraud* yang terjadi pada aplikasi *Finding – Tutor* mengenai proses bertemu langsung antara konsumen jasa tutor dengan penyedia jasa tutor.
2. Menanggulangi *fraud* yang terjadi pada aplikasi *Finding – Tutor* mengenai durasi proses tutor.

1.6 Metodologi

Pembuatan Tugas Akhir ini dilakukan dengan menggunakan metodologi sebagai berikut:

1.6.1 Penyusunan Proposal Tugas Akhir

Tahapan awal dari pengerjaan tugas akhir ini adalah penyusunan proposal tugas akhir. Pada proposal tugas akhir ini, penulis mengajukan penggunaan *QR codes* untuk mendeteksi fraud pada proses transaksi antara penyedia jasa tutor dengan konsumen jasa tutor pada aplikasi *Finding – Tutor* berbasis android studi kasus Surabaya.

1.6.2 Studi Literatur

Tahapan selanjutnya adalah tahapan mencari informasi dari studi literatur berupa referensi paper atau referensi lainnya. Disini referensi yang didapatkan banyak berupa *paper, journal, internet*, maupun buku materi – materi kuliah.

1.6.3 Implementasi Perangkat Lunak

Pembangunan aplikasi akan dilakukan dengan menggunakan Android Studio versi 2.1 ke atas dengan bahasa pemrograman Java. Untuk jenis perangkat mobile itu sendiri menggunakan android versi 4.0 ke atas.

1.6.4 Pengujian dan Evaluasi

Pengujian akan dilakukan oleh dua orang pengguna. Pengguna pertama sebagai penyedia jasa tutor dan pengguna kedua sebagai konsumen jasa tutor. Pengujian ini dilakukan untuk menguji apakah aplikasi berjalan sesuai dengan spesifikasi yang telah didefinisikan atau tidak. Kriteria – kriteria yang akan diujikan berupa waktu yang dibutuhkan untuk membaca *QR Codes* dengan keadaan normal, kondisi cacat, rusak, dan dirotasi. Selain itu kriteria lainnya berupa error yang didapatkan ketika membaca qr code tersebut.

1.6.5 Penyusunan Buku

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat.

1.7 Sistematika Penulisan Laporan

Sistematika penulisan laporan Tugas Akhir adalah sebagai berikut:

1. Bab I. Pendahuluan
   Bab ini berisikan penjelasan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi, dan sistematika penulisan dari pembuatan Tugas Akhir.
2. Bab II. Tinjauan Pustaka
   Bab ini berisi kajian teori dari metode dan algoritma yang digunakan dalam penyusunan Tugas Akhir ini. Secara garis besar, bab ini berisi tentang *QR Codes*, *QR Codes Reader*, *QR Codes Generator*, Perbandingan antara *QR Codes* dan aplikasi sejenis, Android, *Volley library*, *Zxing library*, *CodeIgniter framework*, *MySQL database*, Pemrograman Java dan Startup.
3. Bab III. Perancangan Perangkat Lunak
   Bab ini berisi pembahasan mengenai perancangan dari fitur Pembuatan *QR Codes*, *Scan QR Codes*, dan Pembatalan Transaksi yang digunakan pada Aplikasi *Finding Tutor*.
4. Bab IV. Implementasi
   Bab ini menjelaskan implementasi baik kode program pada Aplikasi di Android Studio, kode program pada *Backend*, maupun implementasi tampilan dari fitur Pembuatan *QR Codes*, *Scan QR Codes*, dan Pembatalan Transaksi yang digunakan pada Aplikasi *Finding Tutor*.
5. Bab V. Hasil Uji Coba dan Evaluasi
   Bab ini berisikan hasil uji coba fungsionalitas dari fitur Pembuatan *QR Codes*, *Scan QR Codes*, dan Pembatalan Transaksi yang digunakan pada Aplikasi *Finding Tutor*. Selain itu terdapat juga hasil uji coba kehandalan dari *QR Codes* itu sendiri.

6. Bab VI. Kesimpulan dan Saran
   Bab ini merupakan bab yang memaparkan kesimpulan dari hasil uji coba yang dilakukan. Selain itu bab ini juga menjelaskan saran untuk pengembangan perangkat lunak ke depannya.
7. Daftar Pustaka
   Bab ini berisi daftar pustaka yang dijadikan literatur dalam Tugas Akhir.

BAB II
TINJAUAN PUSTAKA

Bab ini berisi pembahasan mengenai teori-teori dasar yang digunakan dalam Tugas Akhir. Teori-teori yang dipakai diantaranya adalah *QR Codes*, *QR Codes Reader*, *QR Codes Generator*, Perbandingan antara *QR Codes* dan aplikasi sejenis, Android, *Volley library, Zxing library, CodeIgniter framework, MySQL database,* Pemrograman Java dan Startup.

2.1 Penelitian Terkait

Penggunaaan *QR Codes* sebagai salah satu fitur dalam proses verifikasi sudah banyak dikembangkan dan digunakan dalam banyak aplikasi. Contohnya adalah aplikasi *whatsapp* dan *startup* bernama *eventnook.* Pada aplikasi *website whatsapp,* ketika pengguna ingin melakukan *login* pada akunnya, pengguna tersebut harus melakukan *scan QR Codes* yang ditampilkan pada *website whatsapp* tersebut. Setelah berhasil membacanya, maka secara otomatis sistem melakukan *login* terhadap akun tersebut. Hal ini untuk memudahkan proses *login* dan juga menanggulangi bahwa pengguna tidak benar – benar berada di tempat tersebut. Selain itu juga terdapat *startup* yang berada di Singapura bernama *eventnook. Startup* ini bergerak di bidang yang sama seperti *even organizer. Eventook* banyak dipercaya oleh event – event untuk membantunya dalam proses pendaftaran. Pengguna yang ingin mendaftar event tersebut pertama harus melakukan pendaftaran secara online. Setelah itu, pihak *eventnook* akan memberikan *QR Codes. QR Codes* ini difungsikan sebagai verifikasi oleh pendaftar untuk melakukan daftar ulang dengan proses melakukan *scan QR Codes* pada alat yang disediakan oleh pihak *eventnook.* Namun, aplikasi yang dibuat oleh pihak *eventnook* memiliki kekurangan pada kamera yang disediakan oleh pihak *eventnook* yang lambat dalam membaca *QR Codes* sehingga berdampak pada antrean panjang pada proses registrasi ulang. Untuk itu, Tugas Akhir ini juga

bertujuan untuk menangani masalah dalam hal proses *scan QR Codes.*

2.2 *QR Codes*

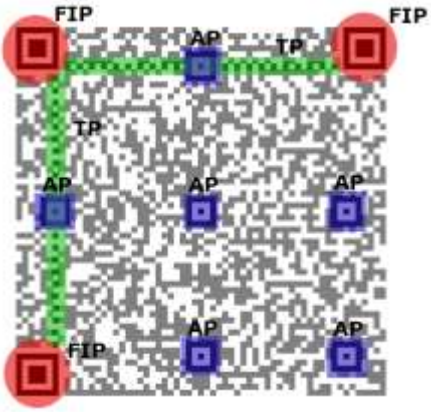
QR Codes merupakan singkatan dari *Quick Response Code*, atau dapat berupa kode respon cepat. *QR Codes* adalah *image* berupa matriks dua dimensi yang memiliki kemampuan untuk menyimpan data di dalamnya. *QR Codes* ini merupakan evolusi dari kode batang atau *Barcode*. Perbedaannya terletak dari data yang disimpan lebih banyak dibandingkan dengan *Barcode*. *QR Codes* dikembangkan oleh Denso Corporation, sebuah perusahaan Jepang yang banyak bergerak di bidang otomotif. *QR Codes* ini dipublikasikan pada tahun 1994 dengan tujuan untuk pelacakan kendaraan di bagian manufaktur dengan cepat dan dengan mendapatkan suatu respon yang cepat pula [1]. Gambar 2.1 adalah contoh dari sebuah *QR Codes.*

Gambar 2.1 Contoh QR Codes

QR Codes sendiri memiliki struktur yang berbeda dengan *Barcode*. Menurut standar ISO/IEC 18004, *QR Codes* didefinisikan berupa symbol yang memiliki struktur yang terdiri dari data, informasi versi dari *QR Codes* itu sendiri, dan pengecekan kode *error* [2]. Sebuah *QR Codes* memiliki struktur bagian berupa:

- *Quiet Zone* yang mengitari simbol *QR Codes* yang direpresentasikan dengan warna hijau [2].
- *3 Finder Patterns* (FIP) yang terletak di ujung-ujung *QR Codes* yang direpresentasikan dengan warna merah [2].
- *2 Timing Patterns* (TP) yang terletak di antara FIP yang direpresentasikan dengan warna biru [2].
- *N Allignment Patterns* (AP) yang berada didalam area data [2].

Gambar 2.2 menjelaskan struktur dari *QR Codes*.

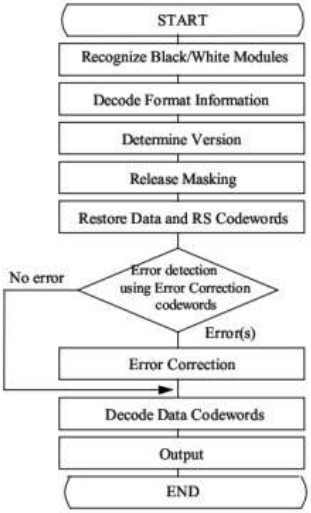


Gambar2.2 Struktur QR Codes [2]

2.2.1 *QR Codes Reader*

QR Codes Reader diperlukan agar pengguna *QR Codes* dapat membacat data yang disimpan oleh *QR Codes*. Terdapat banyak aplikasi yang tergolong sebagai *QR Codes Reader* yang dapat digunakan baik harus diinstal pada perangkat *mobile* ataupun tersedia secara online. Aplikasi tersebut diantaranya seperti *Kaywa Reader*, *Dakode*, dan *Zxing*. Di dalam aplikasi tersebut terdapat

fungsi yang mengonversikan *QR Codes* menjadi teks yang dapat dibaca manusia. Secara garis besar proses pembacaan *QR Codes* dimulai dengan mengenali modul pada warna hitam atau putih yang terdapat pada ujung – ujung *QR Codes.* Setelah itu, sistem akan memastikan versi yang digunakan pada *QR Codes* tersebut. Kemudian sistem akan memanggil fungsi pendeteksi *error* yang terdapat pada *QR Codes*. Lalu, pada akhirnya sistem akan mengembalikan nilai berupa hasil dari *QR Codes* yang telah terbaca. Gambar 2.3 menjelaskan diagaram alir proses pembacaan *QR Codes*.

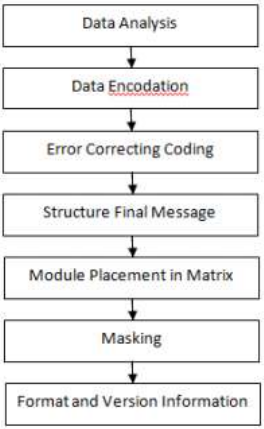


Gambar 2.3 Diagram Alir Proses Pembacaan QR Codes [1]

2.2.2 *QR Codes Generator*

Untuk merubah data yang telah ditentukan menjadi sebuah *QR Codes*, diperlukan sebuah software untuk men – *generate* atau merubah data berupa teks dan angka menjadi *QR Codes*. Beberapa aplikasi untuk merubah data menjadi *QR Codes* diantaranya *Kaywa QR-Code, SnapMaxe QR Code Generator*. Secara garis

besar untuk membuat *QR Codes*, langkah – langkah pembuatannya merupakan kebalikan dari proses pembacaan *QR Codes*. Proses dimulai dengan menganalisis dan menyiapkan data yang akan diubah menjadi *QR Codes*. Setelah itu, sistem akan melakukan proses pengecekan *error* pada data inputan. Setelah itu data – data inputan tersebut akan dibuat menjadi matrix dan dilakukan proses pembungkusan menjadi *QR Codes*. Setelah itu dilakukan proses pembuatan format dan pemberian informasi pada versi *QR Codes*. Gambar 2.4 menjelaskan diagram alir proses pembuatan *QR Codes*. Selain itu, setiap kata memiliki *QR Codes* yang berbeda – beda. Hal ini berdasarkan *error correction*, dan fungsi *masking* yang ada ketika pembuatan *QR Codes*. Tabel 2.1 menjelaskan bebarapa contoh kata – kata yang di *encode* menjadi *QR Codes*.



Gambar 2.4 Diagram Alir Proses Pembuatan QR Codes [1]

Tabel 2.1 Contoh Data yang telah di Encode

No	Contoh Kata	Hasil *Encode*
1	Halo	
2	Selamat Pagi	
3	5113100002-Haha	

Berdasarkan Tabel 2.1, setiap kata memiliki *QR Codes* yang berbeda – beda. Perbedaan yang paling mencolok terdapat pada *N Allignment Patterns* yang berada di dalam data. Semakin banyak data yang harus di ubah menjadi sebuah *QR Codes*, semakin banyak pula isi *N Allignment Patterns* yang berada di dalam *3 Finder Patterns*. Hal ini dikarenakan *N Allignment Patterns* merepresentasikan data – data yang diubah ke dalah sebuah *QR Codes*.

2.2.3 Perbandingan antara *QR Codes* dengan Teknologi Sejenis

Disamping teknologi *QR Codes* yang digunakan pada aplikasi *Finding Tutor*, terdapat teknologi – teknologi lainnya yang sejenis dengan *QR Codes*. Diantaranya adalah *Barcode* dan RFID. Tabel 2.2 menjelaskan perbedaan antara ketiga teknologi tersebut.

Tabel 2.2 Perbedaan QR Code dengan Barcode dan RFID [3]

Kategori	QR Codes	Barcode	RFID
Kapasitas Data	<= 7000 karakter	<= 20 karakter	<= 1000 karakter
Kecepatan Scan	Cepat, dapat dirotasi 90 derajat, harus di scan manual	Cepat, tidak dapat dirotasi 90 derajat, harus di scan manual	Cepat tanpa harus di scan manual
Sinkronisasi dengan Kamera HP	Ya	Tidak	TIdak
Harga	Relatif murah, sudah ada library untuk mengaplikasikan *QR Codes*	Relatif murah	Relatif lebih mahal karena RFID harus ditempelkan

	berdasarkan data yang diinginkan		ke alat untuk di scan

Pada Tabel 2.2 dijelaskan bahwa *QR Codes* memiliki keunggulan dalam hal kapasitas data yang disimpan. *QR Codes* dapat menyimpan lebih dari 7000 karakter. Di samping itu, *QR Codes* cepat dalam hal memberikan data hasil *scan* nya dan dapat merotasi hingga 90 derajat dalam hal melakukan *scan*. Hal ini lebih baik dibandingkan dengan *Barcode* yang tidak dapat memberikan data hasil *scan* apabila di rotasi hingga 90 derajat. Selain itu, *QR Codes* dapat di sinkronisasikan dengan kamera HP. Dikarenakan aplikasi *Finding Tutor* merupakan aplikasi berbasis android, maka *QR Codes* merupakan teknologi yang cocok tanpa harus mensinkronisasikan dengan alat *scanner* khusus lainnya.

2.3 Android

Android adalah sebuah software yang digunakan pada sebuah handphone, tablet, dan perangkat keras yang terdapat pada mobil sebagai alat entertainment. Android di publikasikan pada tahun 2003 dan telah menjadi sistem operasi yang sangat popular. Android sendiri merupakan sistem operasi berbasis *open source* dimana pengguna android dapat dengan leluasa melakukan perubahan – perubahan pada fitur yang diberikan oleh android untuk kepentingannya sendiri [4]. Selain itu, *open source* dimaksudkan pada kode android ini bisa didapatkan secara gratis dan Google merilis kodenya sendiri dibawah Lisensi Apache. Kode dengan sumber terbuka dan lisensi perizinan pada Android memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator nirkabel, dan pengembang aplikasi [4]. Sistem Operasi android, dari tahun ke tahun, telah di kembangkan oleh perusahaan Google yang sebagai perusahaan utama dalam pembuatan perubahan yang terjadi pada sistem operasi Android.

2.3.1 *Volley Library*

Volley adalah sebuah *library* yang dimaksudkan untuk menangani masalah jaringan atau koneksi. *Volley* dibangun oleh perusahaan Google dan dipublikasikan pada masa Google I/O 2013. *Volley* digunakan untuk mengatasi kehilangan pada Android SDK, dimana dibutuhkan sebuah kelas jaringan yang mampu bekerja tanpa menggangu pengguna [5]. Kelebihan yang diberikan *Volley library* diantaranya adalah *Volley* bagus untuk mengoperasikan operasi jaringan yang menggunakan model RPC yang mengisi sebuah UI dan baik untuk menjalankan fungsi *background RPCs* [6].

2.3.2 *Zxing Library*

Zxing library adalah sebuah *library* yang mendukung pembuatan *decoding* dan pembuatan *barcodes* seperti *QR Codes*, PDF 417, EAN, UPC, Aztech, Data Matrix, dan Codabar. *Library* ini menggunakan Java sebagai basis bahasa pemrogramannya [7]. Untuk membuat sebuah aplikasi menggunakan *QR Codes*, pengguna cukup memasukkan *library* ini kedalah IDE yang digunakan. Setelah itu, pengguna cukup memanggil fungsi yang diperlukan seperti fungsi untuk men – *generate* data menjadi *QR Codes* ataupun sebaliknya. *Zxing* juga mendukung IDE selain Java seperti .Net 2.0, 3.5, 4.0, 4.5, dan 4.6. Selain itu, *Zxing* dapat dipakai pada Windows Phone 7.0, 7.1, Unity, dan Xamarin Android [7].

2.4 *CodeIgniter Framework*

CodeIgniter Framework merupakan sebuah framework yang berbasis PHP yang memiliki karakteristik sebagai framework yang kuat dan diperuntukkan untuk *developer* yang ingin membuat aplikasi secara elegan dan mudah di dalam pembuatan website dengan fitur yang penuh [8]. Penggunaan *CodeIgniter* pada

aplikasi *Finding Tutor* adalah sebagai *backend* aplikasi tersebut. *Backend* yang dimaksud adalah jaringan penghubung antara program aplikasi yang dibuat pada *android studio* dengan basis data yang digunakan yaitu *MySQL database*. Dengan adanya *backend* ini, aplikasi dapat mengambil data, memasukkan data, dan merubah data yang ada pada basis data yang telah dibuat.

2.5 *MySQL Database*

MySQL adalah basis data yang bersifat *open source*. MySQL dibangun, didukung, dan dipasarkan oleh Oracle Corporation. Basis data ini memiliki struktur untuk menampung data. Untuk memasukkan, menghapus, dan mengubah data yang sedang ingin dibangun, diperlukan sebuah sistem manajemen basis data dan MySQL adalah salah satunya. MySQL ini mendukung data dalam bentuk *relational database*. Sebuah *relational database* disimpan di dalam MySQL dalam bentuk tabel – tabel yang saling berhubungan. Ciri – ciri tabel dari MySQL ini adalah sama seperti ciri – ciri tabel pada umumnya. Memiliki baris, kolom dan nama tabel itu sendiri. Sampai saat ini, hampir seluruh aplikasi berbasis website membutuhkan sebuah basis data untuk di – *hosting* di sebuah website. Salah satu aplikasi yang mendukung MySQL Database adalah PHPmyadmin. PHPmyadmin menyediakan fitur untuk menambah, menghapus, dan mengubah tabel dalam sebuah basis data. Disamping itu, PHPmyadmin juga menyediakan fitur kepada pengguna untuk melakukan fungsi *sql query*, mencari baris ataupun kolom sebuah tabel dan melakukan *import* ataupun *import* pada basis data yang telah dibuat [9].

2.6 Pemrograman Java

Pemrograman Java adalah bahasa programming berbasis Java yang telah diakui di komunitas dunia baik komunitas pengembang aplikasi maupun komunitas penyedia jasa konten internet. Pemrograman Java memberikan kemudahan bagi

pengembang aplikasi untuk mengembangkan *code* yang dibuat tanpa harus memberikan "*port"* pada setiap perangkat keras dan perangkat lunak yang digunakan [10]. Bahasa pemrograman *Java* dirancang untuk portabilitas maksimal dengan meminimalisir impelementasi pada *dependency*. Contohnya pada sebuah tipe data berupa Int, merupakan 32-bit dari sebuah integer terlepas dari arsitektur CPU yang digunakan [10]. Pemrograman Java hingga saat ini telah diterapkan di banyak aplikasi baik aplikasi berbasis website maupun aplikasi berbasis mobile. Aplikasi *Finding Tutor* yang berjalan di android ini dibuat menggunakan bahasa pemrograman Java.

2.7 Startup

Startup adalah tahap awal di dalam siklus perusahaan di mana pembuat perusahaan memindahkan ide bisnis yang telah dibuat untuk mendapatkan pembiayaan, meletakkan dasar struktur bisnis, dan memulai operasi atau perdagangan [11]. Pada aplikasi *Finding Tutor* ini, agar dapat masuk ke dalam ranah startup, alur bisnis harus ditetapkan agar dapat memberikan pembiayaan atau keuntungan.

(Halaman ini sengaja dikosongkan)

BAB III
PERANCANGAN SISTEM

Bab ini menjelaskan tentang analisis ,perancangan dan pembuatan sistem yang akan dibangun. Sistem perangkat lunak yang dibuat pada Tugas Akhir ini adalah penggunaan qr codes untuk mendeteksi fraud pada proses transaksi antara penyedia jasa tutor dengan konsumen jasa tutor pada aplikasi finding – tutor berbasis android. Pada bab ini, terdapat tahap analisis yang membahas persiapan yang akan menjadi pokok pikiran pembuatan aplikasi. Selain itu, tahap perancangan sistem akan membahas dasar – dasar dalam pembuatan aplikasi, yang meliputi perancangan basis data, tampilan antar muka halaman aplikasi, dan perancangan alur proses pada aplikasi ini. Pada bab ini pula akan dijelaskan gambaran umum sistem dalam bentuk *flowchart.*

3.1 Analisis

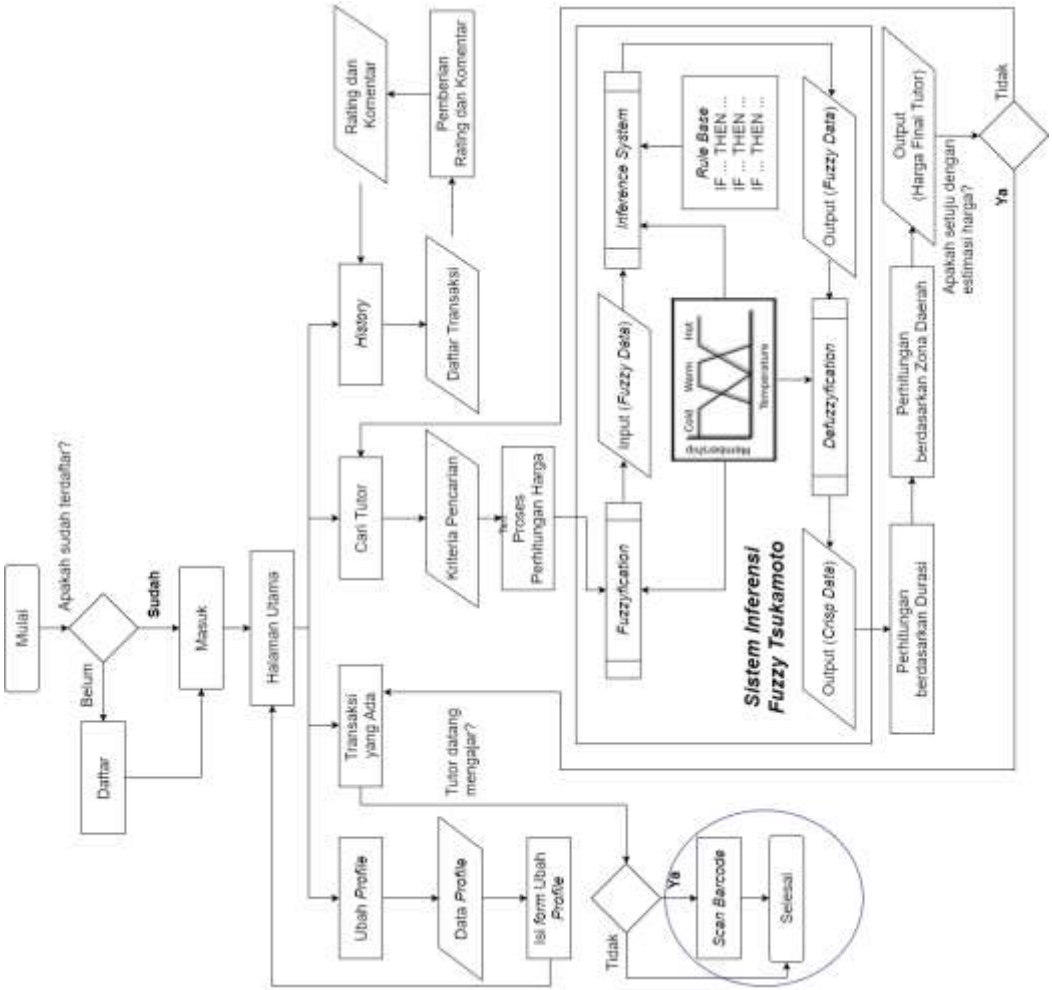
Pada tahap analisis ini yang akan dibahas meliputi analisis masalah, analisis kebutuhan, deskripsi umum sistem, dan kasus penggunaan sistem yang dibuat.

3.1.1 Deskripsi Umum Aplikasi

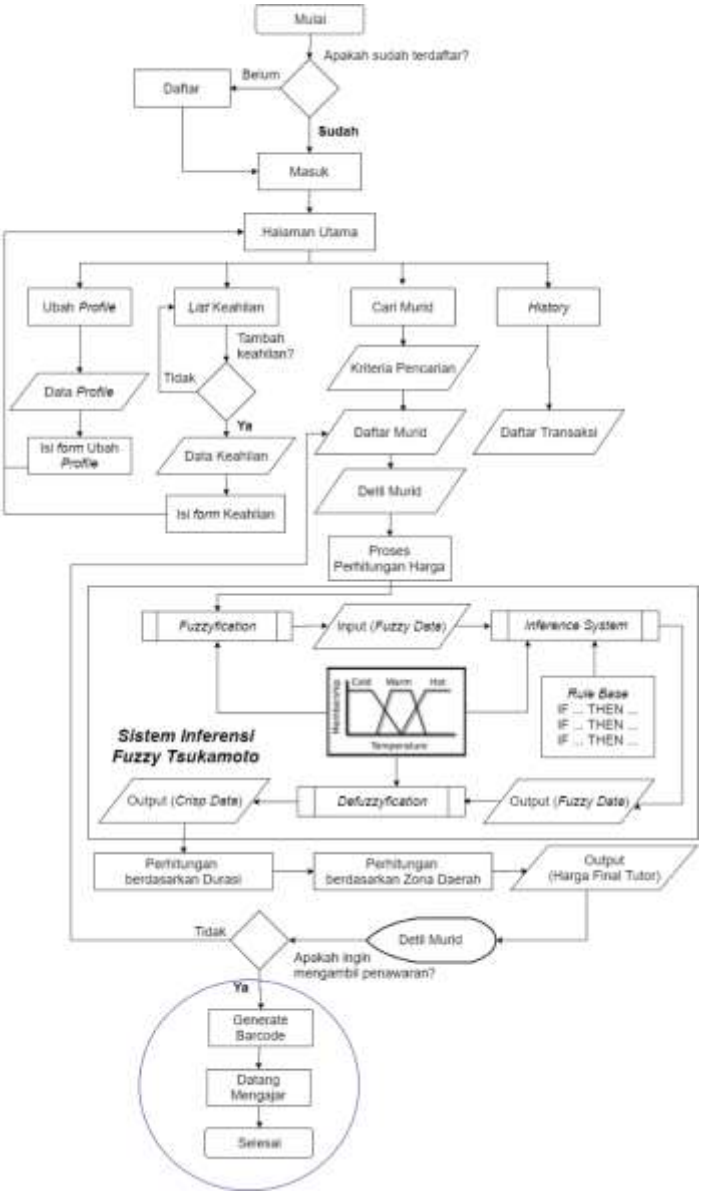
Aplikasi *Finding Tutor* merupakan aplikasi berbasis android yang dibuat untuk membantu murid dalam mencari penyedia jasa tutor. Murid dengan menggunakan aplikasi ini dapat memilih penyedia jasa tutor sesuai dengan kriteria yang dimasukkan beserta dengan mata pelajaran yang ingin diajarkan. Selain itu, aplikasi ini juga mengukur tingkat urgenitas transaksi yang ingin dilakukan murid kepada penyedia jasa tutor. Aplikasi ini berjalan pada cakupan wilayah daerah Surabaya. Aplikasi ini dimulai dengan murid memasukkan data pengguna dan kemudian memesan transaksi. Pada pemesanan transaksi, murid akan memesan sesuai dengan kriteria kebutuhan dari murid tersebut. Mulai dari tanggal

transaksi, kemudian jam dimulainya proses tutor, dan lokasi transaksi. Selain itu murid dapat juga memilih mata pelajaran yang diinginkan untuk di tutorkan dan juga murid dapat memilih jenis kelamin dari tutor tersebut. Hal ini untuk menghindari apabila seorang murid wanita tidak ingin di tutorkan oleh laki – laki. Selain itu murid juga dapat memilih usia maksimal tutor yang diperbolehkan untuk mengambil transaksinya.

Pada aplikasi *Finding Tutor* ini terdapat tiga modul terkait proses pembuatannya, yaitu modul penentuan prioritas transaksi, modul penentuan harga, dan modul pada buku ini yaitu pencegahan *fraud* pada proses transaksi. Pada modul pertama murid ketika ingin melakukan transaksi akan disediakan pilihan berupa prioritas yang diinginkan, apakah jenis kelamin dari tutor ataupun waktu transaksi. Kriteria yang tersedia berupa jenis kelamin dari tutor, tanggal mulainya transaksi, dan usia maksimal dari tutor yang diperbolehkan untuk mengambil transaksinya. Selanjutnya setelah murid selesai melakukan proses memasukkan data transaksinya, siste akan masuk kepada modul penentuan harga. Pada modul ini, aplikasi *Finding Tutor* membuat sistem menampilkan harga yang fluktutatif kepada murid. Harga yang diberikan pada tanggal proses transaksi yang mepet dengan tanggal ketika memesan memiliki harga yang lebih mahal ketimbang dengan pemesanan pada tanggal lebih dari dua minggu. Setelah murid menyetujuinya, maka alur sistem berlanjut pada modul pencegahan fraud. *Fraud* pada aplikasi ini salah satunya adalah tidak hadirnya penyedia jasa tutor sesuai dengan jadwal pada transaksi yang telah dimasukkan oleh murid. Namun, sistem tetap mencatat bahwa penyedia jasa tutor telah datang dan telah menyelesaikan transaksi yang diambil. Hal ini dapat memberikan kerugian pada murid karena sistem telah mencatat transaksi telah dilakukan namun pada sebenarnya berbeda. Gambar 3.1 dan Gambar 3.2 menjelaskan sistem Finding – Tutor secara keseluruhan. Penanganan *fraud* pada Gambar 3.1 dan Gambar 3.1 direpresentasikan dengan melingkari proses yang digunakan pada Tugas Akhir ini.



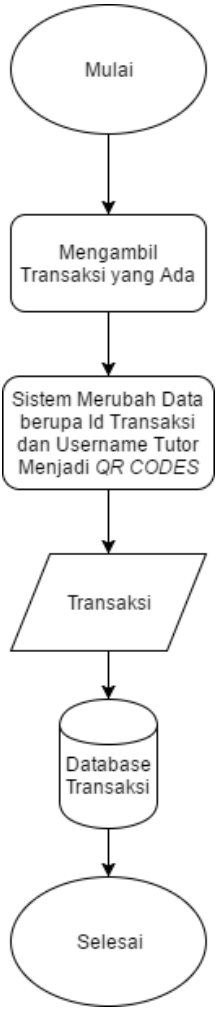
Gambar 3.1 Flowchart Keseluruhan Murid Mencari Tutor



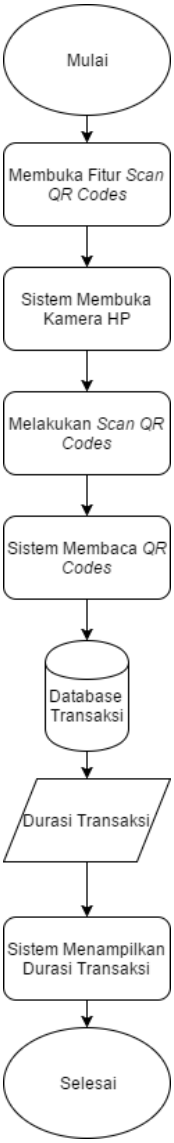
Gambar 3.2Flowchart Keseluruhan Tutor Mencari Murid

Untuk penanganan *fraud* tersebut, diperlukan sebuah fitur untuk menjamin kedua belah pihak bertemu bertatap muka secara langsung sesuai dengan waktu transaksi yang telah ditentukan. Pada aplikasi *Finding Tutor*, setelah penyedia jasa tutor memilih transaksi, tampilan aplikasi penyedia jasa tutor akan memunculkan *QR Codes*. *QR Codes* ini dibuat berdasarkan gabungan antara id transaksi dan username penyedia jasa tutor. Setelah itu, untuk memulai transaksi *QR Codes* pada penyedia jasa tutor harus terlebih dahulu di *scan* pada aplikasi *Finding Tutor* pihak murid. Setelah itu, transasksi baru bisa berjalan dengan memunculkan waktu *countdown* pada aplikasi *Finding Tutor* pihak murid. Kemudian, setelah waktu *countdown* memunculkan angka nol yang berarti transaksi habis, murid harus melakukan *scan* kembali untuk memberitahu ke sistem bahwa transaksi telah selesai. Apabila murid tidak melakukan *scan* untuk kedua kalinya, maka penyedia jasa tutor tidak dapat mengambil transaksi yang lain karena sistem menganggap transaksi masih sedang berjalan. Disamping itu, *QR Codes* yang dibuat memiliki masa aktif satu jam hingga murid melakukan *scan* pertama. Apabila murid tidak melakukan *scan* untuk pertama kalinya lebih dari satu jam, maka penyedia jasa tutor dianggap membatalkan transaksi dan *QR Codes* hilang otomatis dari sistem dan penyedia jasa tutor mendapat hukuman karena telah membatalkan transaksi. Gambar 3.3 dan Gambar 3.4 menggambarkan diagram alir sistem.

Selain itu, baik pihak penyedia jasa tutor maupun murid, dapat membatalkan transaksi yang ada. Apabila penyedia jasa tutor membatalkan transaksi yang telah diambil, penyedia jasa tutor tidak dapat mengambil transaksi lainnya selama satu menit. Apabila murid membatalkan transaksi yang telah diambil oleh penyedia jasa tutor, murid akan terkena hukuman berupa penambahan biaya sebesar Rp20.000.



Gambar 3.3 Diagram Alir Generate QR Codes



Gambar 3.4 Diagram Alir Scan QR Codes

3.1.2 Analisis Kebutuhan Sistem

Kebutuhan utama dari sistem ini adalah sistem dapat menangani *fraud* yang terjadi dan memastikan penyedia jasa tutor dan murid bertemu bertatap muka secara langsung. Secara rinci, daftar kebutuhan fungsional dapat dilihat pada Tabel 3.1.

Tabel 3.1 Daftar Kebutuhan Fungsional Sistem

Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi
F-0001	Mengambil transaksi	Penyedia jasa tutor dapat mengambil transaksi yang ada yang berasal dari basis data transaksi.
F-0002	Melihat *QR Codes* yang telah di - *generate*	Penyedia jasa tutor mendapatkan *QR Codes* sesuai dengan transaksi yang diambil. Sistem sebelumnya mengubah data id transaksi dan username penyedia jasa tutor menjadi sebuah *QR Codes*.
F-0003	Melakukan *scan QR Codes*	Murid dapat melakukan *scan QR Codes* yang dimiliki oleh penyedia jasa tutor.
F-0004	Membatalkan transaksi	Penyedia jasa tutor dapat membatalkan transaksi yang telah diambil. Murid juga dapat membatalkan transaksi yang telah diambil oleh penyedia jasa tutor.

3.1.3 Analisis Aktor

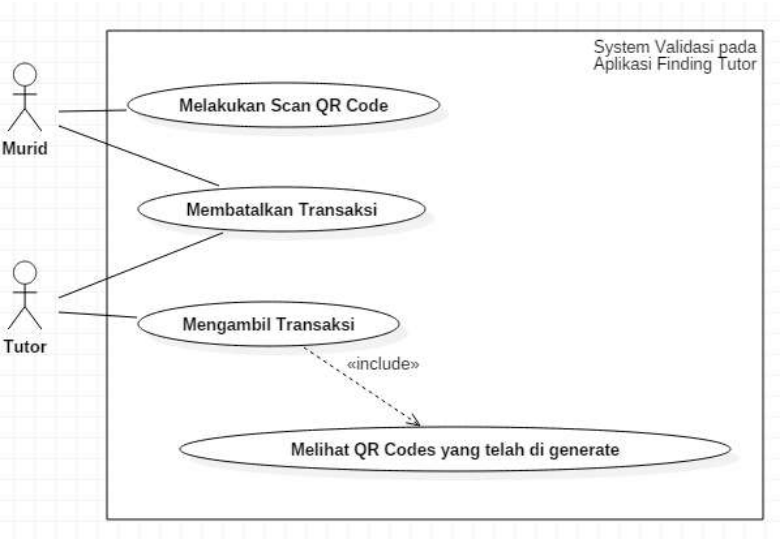
Aktor adalah pihak – pihak yang berinteraksi secara langsung dengan sistem. Pada aplikasi *Finding Tutor* ini memiliki dua aktor yang berinteraksi secara langsung, yaitu penyedia jasa tutor dan murid.

3.1.4 Kasus Penggunaan

Berdasarkan daftar kebutuhan fungsional sistem dan deskripsi umum aplikasi yang telah dibuat, dibentuklah diagram kasus penggunaan yang dimaksudkan untuk memenuhi kebutuhan fungsional yang telah dipaparkan. Kasus penggunaan, untuk lebih rinci, dijelaskan pada Tabel 3.2 dan digambarkan pada Gambar 3.5.

Tabel 3.2 Daftar Kode Kasus Penggunaan

Kode Kasus Penggunaan	**Nama**	**Aktor**
UC-0001	Mengambil transaksi	Penyedia Jasa Tutor
UC-0002	Melihat *QR Codes* yang telah di - *generate*	Penyedia Jasa Tutor
UC-0003	Melakukan *scan QR Codes*	Murid
UC-0004	Membatalkan transaksi	Penyedia Jasa Tutor, Murid



Gambar 3.5 Diagram Kasus Penggunaan

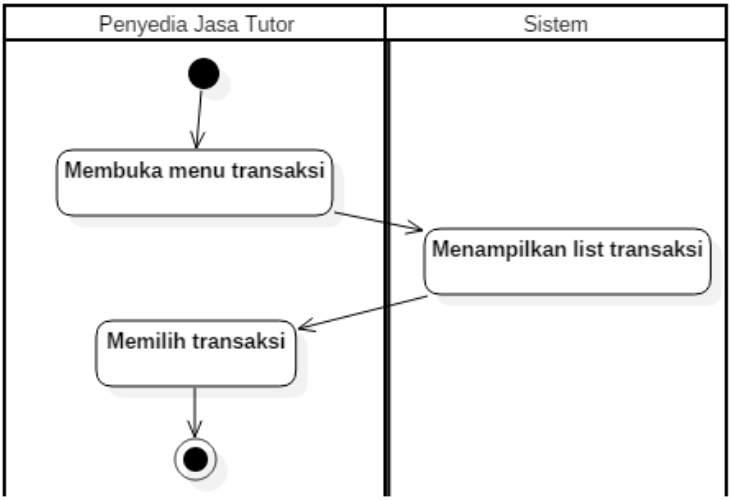
A. Mengambil Transaksi (UC-0001)

Pada kasus penggunaan ini, mula – mula penyedia jasa tutor memilih transaksi yang telah tersedia pada sistem. Setelah itu transaksi yang dipilih diambil oleh penyedia jasa tutor. Tabel 3.3 menjelaskan spesifikasi kasus penggunaannya dan Gambar 3.6 menjelaskan diagram aktivitasnya

Tabel 3.3 Spesifikasi Kasus Penggunaan Mengambil Transaksi

Kode	UC-0001
Nama	Mengambil Transaksi
Deskripsi	Penyedia jasa tutor mengambil transaksi yang telah dipilih
Tipe	Fungsional
Pemicu	Penyedia jasa tutor memilih menu transaksi
Aktor	Penyedia Jasa Tutor

Kondisi Awal	Penyedia jasa tutor melihat daftar transaksi	
Kondisi Akhir	Penyedia jasa tutor memilih transaksi yang sesuai	
Alur Kejadian Normal	**Penyedia Jasa Tutor**	**Sistem**
	1. Memilih Menu Transaksi	
		2. Menampilkan list transaksi yang tersedia
	3. Memilih transaksi	
Alur Kejadian Alternatif	-	



Gambar 3.6 Diagram Aktivitas Mengambil Transaksi

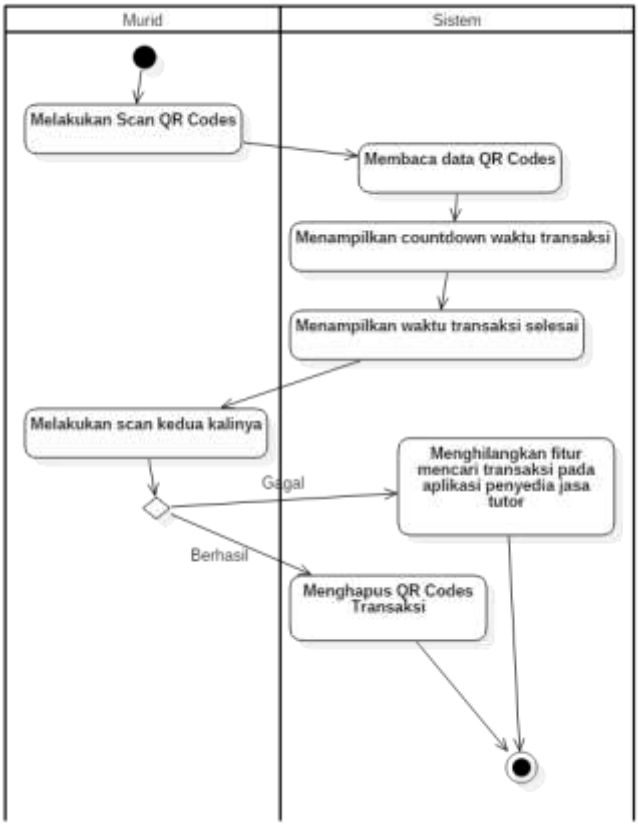
B. Melakukan Scan QR Codes (UC-0003)

Pada kasus penggunaan ini, murid melakukan *scan QR Codes* yang dimiliki oleh penyedia jasa tutor untuk memasitkan bahwa penyedia jasa tutor datang bertemu langsung untuk melakukan transaksi. Tabel 3.4 menjelaskan spesifikasi kasus penggunaannya dan Gambar 3.7 menjelaskan diagram aktivitasnya.

Tabel 3.4 Spesifikasi Kasus Penggunaan Melakukan Scan QR Codes

Kode	UC-0003	
Nama	Melakukan *scan QR Codes*	
Deskripsi	Murid melakukan *scan QR Codes* milik penyedia jasa tutor untuk memulai transaksi	
Tipe	Fungsional	
Pemicu	Penyedia jasa tutor datang dengan QR Codes Transaksi	
Aktor	Murid	
Kondisi Awal	Penyedia jasa tutor datang sebelum masa berlaku *QR Codes* telah habis	
Kondisi Akhir	Transaksi selesai	
Alur Kejadian Normal	**Murid**	**Sistem**
	1. Melakukan *scan QR Codes* untuk memulai transaksi	
		2. Membaca data *QR Codes*

		3. Menampilkan *countdown* transaksi
		4. Menampilkan waktu transaksi telah selesai
	5. Melakukan *scan QR Codes* untuk kedua kalinya	
	A.5 Jika murid gagal melakukan *scan QR Codes* untuk kedua kalinya	
		6. Sistem menghapus *QR Codes* transaksi
Alur Kejadian Alternatif	-	
Eksepsional	**Murid**	**Sistem**
		A.5.1. Sistem menghilangkan fitur mencari transaksi pada aplikasi penyedia jasa tutor



Gambar 3.7 Diagram Aktivitas Melakukan Scan QR Codes

C. Membatalkan Transaksi (UC-0004)

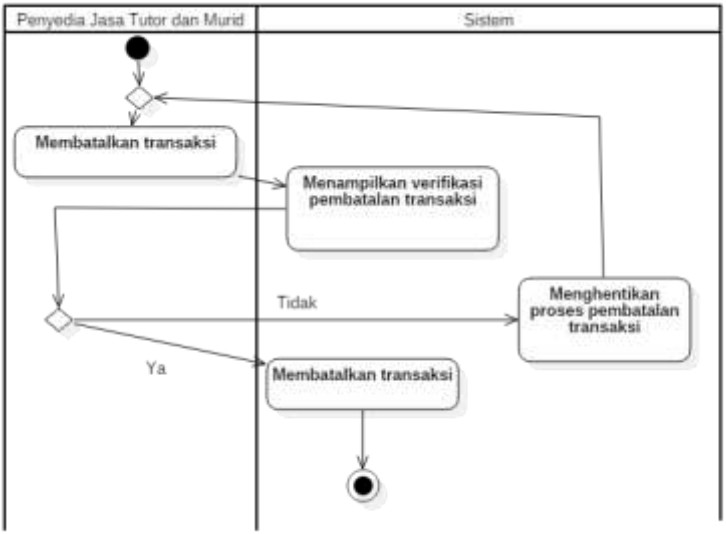
Pada Kasus penggunaan ini, baik murid maupun penyedia jasa tutor membatalkan transaksi yang sedang berjalan. Namun, setelah membatalkan transaksi, baik murid maupun penyedia jasa tutor akan mendapatkan *punishment. Punishment* yang diberikan kepada murid adalah penambahan harga sebesar RP20.000,00 pada proses transaksi berikutnya. Sementara itu *punishment* yang

diberikan kepada penyedia jasa tutor adalah penyedia jasa tutor tidak dapat mengambil transaksi selanjutnya selama satu menit kedepan. Tabel 3.5 menjelaskan spesifikasi kasus penggunaannya dan Gambar 3.8 menjelaskan diagram aktivitasnya.

Tabel 3.5 Spesifikasi Kasus Pengguna Membatalkan Transaksi

Kode	UC-0004	
Nama	Membatalkan transaksi	
Deskripsi	Penyedia jasa tutor atau murid membatalkan transaksi	
Tipe	Fungsional	
Pemicu	Penyedia jasa tutor atau murid menekan tombol batal transaksi	
Aktor	Penyedia Jasa Tutor atau Murid	
Kondisi Awal	Transaksi sedang berlangsung	
Kondisi Akhir	Transaksi dibatalkan	
Alur Kejadian Normal	**Murid atau Penyedia Jasa Tutor**	**Sistem**
	1. Baik penyedia jasa tutor atau murid ingin membatalkan transaksi.	
		2. Sistem menampilkan verifikasi pembatalan transaksi.
	3. Penyedia jasa tutor atau murid	

	membatalkan transaksi.	
	A.3. Penyedia jasa tutor atau murid menghentikan proses pembatalan transaksi.	
		4. Sistem membatalkan transaksi yang terjadi.
Alur Kejadian Alternatif	**Murid atau Penyedia Jasa Tutor**	**Sistem**
	A.4. Menghentikan proses Pembatalan Transaksi	
		A.4.1. Membatalkan proses pembatalan transaksi
		A.4.2. Kembali ke alur nomor



Gambar 3.8 Diagram Aktivitas Membatalkan Transaksi

3.2 Perancangan Sistem

Tahap ini berisi perancangan pada basis data, tampilan antarmuka, dan perancangan arsitektur sistem yang diharapkan dapat memenuhi tujuan dari pengembangan aplikasi ini. Dalam membangun aplikasi ini, dukungan dalam sector lingkungan juga perlu diperhatikan. Berikut adalah informasi mengenai lingkungan pengembangan aplikasi.

Perangkat keras:

- Komputer : Prosesor Intel® Core™ i5-CPU (2.60 GHz), RAM 12 GB dan memori sekunder 4 GB
- Perangkat Bergerak : Android version 5.0.2, CPU Octa-core(2.0GHz), RAM 3 GB, internal memory 32GB

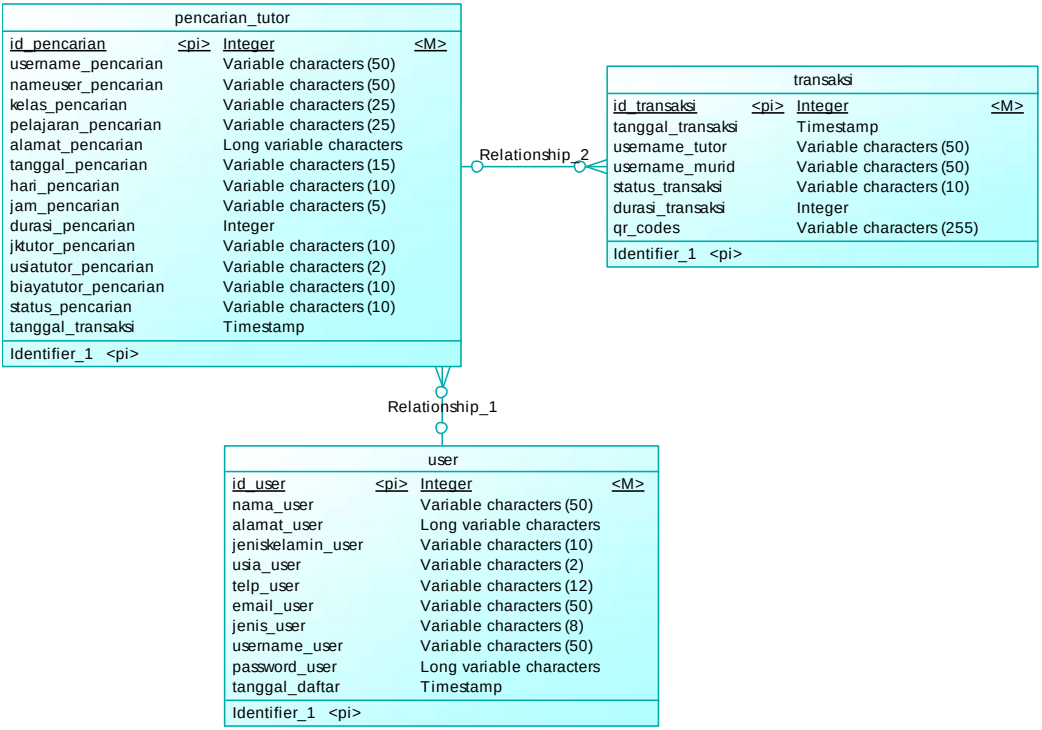
Perangkat lunak:

- Sistem Operasi : Windows 10 pro
- IDE : Android Studio
- Basis Data : MySQL

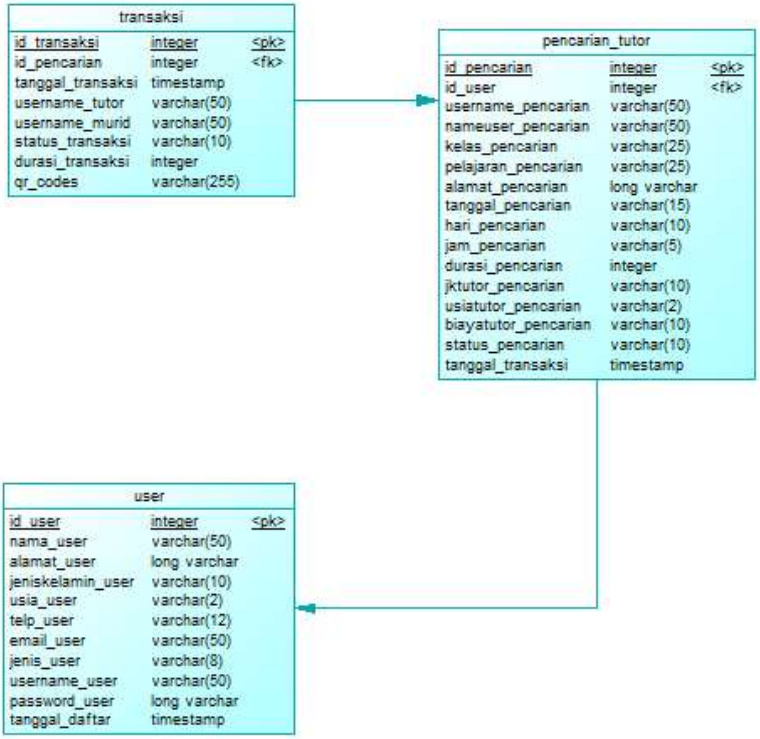
3.2.1 Perancangan Basis Data

Subbab ini menjelaskan akan rancangan basis data yang digunakan untuk menopang aplikasi validasi *Finding Tutor* tersebut. Pada basis data ini, terdapat satu tabel utama berupa tabel transaksi yang menampung data data transaksi mulai dari id transaksi, id pencarian tutor, hingga qr codes yang telah diubah. Tabel transaksi ini berhubungan dengan dua tabel lainnya yaitu tabel pencarian tutor dan tabel user. Tabel pencarian tutor menjelaskan bahwa ketika murid melakukan trasansaksi dan melakukan proses permintaan pencarian tutor maka informasi – informasi yang diberikan oleh murid akan masuk ke dalam tabel pencarian tutor tersebut. Tabel pencarian tutor berisi informasi – informasi umum seperti pelajaran pencarian, hari, durasi dan tanggal pencarian, tanggal transaksi, dan juga nama serta *username* dari murid yang melakukan pencarian. Kemudian, Tabel pencarian ini juga terhubung dengan tabel user yang berisi data umum mengenai user yang menggunakan aplikasi ini. User pada aplikasi ini dibedakan menjadi dua jenis yaitu murid dan pencarian tutor. Informasi umum seperti *username* dan *password* dari user, nama, alamat, serta tanggal daftar disimpan di tabel ini. Setelah itu variabel *id_user* disimpan ke dalam tabel pencarian tutor agar kedua tabel tersebut terhubung. Setelah itu, variabel *id_pencarian* disimpan kedalam tabel transaksi agar tabel transaksi dan tabel pencarian tutor terhubung. Gambar 3.9 menjelaskan CDM gambaran perancangan basis data validasi *Finding Tutor* ini dan

Gambar 3.10 menjelaskan PDM gambarang perancangan basis datanya.



Gambar 3.9 Conceptual Data Model



Gambar 3.10 Physical Data Model

3.2.2 Perancangan Tampilan Antarmuka

Subbab ini menjelaskan rancangan antarmuka dari aplikasi *Finding Tutor* yang akan berinteraksi secara langsung dengan pengguna.

A. Rancangan Antarmuka Halaman Menampilkan QR Code yang telah di generate

Halaman ini merupakan halaman tampilan *QR Codes* yang telah diubah oleh sistem. Pertama, penyedia jasa tutor membuka halaman transaksi dan memilih transaksi yang ada. Kemudian sistem secara otomatis akan menampilkan *QR Codes* yang telah diubah. *QR Codes* ini memiliki masa waktu satu jam. Apabila *QR Codes* tidak di *scan* sampai batas waktu satu jam maka transaksi dianggap gagal dan *QR Codes* hangus. Kemudian, pada halaman yang berisi *QR Codes* ini, tedapat 3 tombol *back* yang berfungsi kembali ke halaman awal, *transaksi dibatalkan* yang berfungsi untuk membatalkan transaksi, dan *transaksi selesai* yang berfungsi untuk mengakhiri transaksi yang telah dilakukan. Tombol *transaksi selesai* hanya dapat digunakan apabila transaksi benar – benar telah selesai, yaitu dengan waktu transaksi telah habis dan murid telah dua kali melakukan *scan QR Codes.* Apabila penyedia jasa tutor menekan tombol *transaksi selesai*, namun sistem masih membaca bahwa transaksi masih sedang berjalan, sistem akan memberikan notifikasi ke layar penyedia jasa tutor bahwa transaksi masih sedang berjalan dan transaksi tidak dapat diberhentikan. Hal ini difungsikan untuk menghindari masalah yang timbul dikarenakan penyedia jasa tutor melakukan kecurangan dengan sengaja menekan tombol *transaksi selesai* meskipun transaksi masih belum selesai. Hal ini juga dapat merugikan pihak murid karena transaksi yang berjalan tidak sesuai dengan transaksi yang diminta oleh murid. Pada tombol *transaksi dibatalkan*, penyedia jasa tutor dapat melakukan pembatalan transaksi apabila transaksi yang telah diambil ingin di batalkan. Namun, apabila penyedia jasa tutor menekan pembatalan transaksi, maka penyedia jasa tutor akan mendapatkan hukuman dari sistem. Gambar 3.11 menjelaskan rancangan antarmuka halaman detil transaksi dan Gambar 3.12 menjelaskan rancangan antarmuka halaman *QR Codes*.

Gambar 3.11 Rancangan Antarmuka Halaman Detil Transaksi

Gambar 3.12 Rancangan Antarmuka Halaman QR Codes

B. Rancangan Antarmuka Halaman Scan QR Code

Pada halaman ini murid dapat melakukan *scan* pada *QR Codes* milik penyedia jasa tutor untuk menandakan transaksi dimulai dan transaksi telah selesai. Setelah murid menekan tombol *scan*, aplikasi finding tutor akan secara otomatis menggunakan kamera *handphone* untuk melakukan *scan.* Setelah *scan* pertama berhasil, maka akan muncul halaman *countdown* dari transaksi tersebut. Isi dari *countdown* transaksi tersebut adalah waktu transaksi yang diambil dari database transaksi yang telah di masukkan terlebih dahulu oleh murid ketika ingin memesan tutor. Setelah durasi habis maka sistem akan kembali ke halaman scan untuk melakukan *scan* yang kedua kalinya yang menandakan transaksi telah selesai. Apabila murid tidak melakukan *scan* untuk yang kedua kalinya, halaman *QR Codes* pada penyedia jasa tutor tidak akan bisa hilang dan penyedia jasa tutor tidak dapat mengambil transaski yang lain karena sistem menganggap bahwa

transaksi belum selesai. Gambar 3.13 dan Gambar 3.14 menjelaskan rancangan dari proses murid memulai transaksi.

Gambar 3.13 Rancangan Antarmuka Halaman Scan QR Codes

Gambar 3.14 Rancangan Antarmuka Halaman Durasi Transaksi

C. Rancangan Antarmuka Halaman Batal Transaksi

Pada halaman ini baik murid maupun penyedia jasa tutor dapat membatalkan transaksi yang ada. Halaman pembatalan transaksi ini dibedakan antara penyedia jasa tutor dengan murid. Pada penyedia jasa tutor, tombol batal transaksi terdapat pada halaman *QR Codes* yang telah dibuat, sementara tombol batal transaksi pada murid terdapat pada tombol *cancel* yang berada di halaman *detil transaksi*. Fungsi yang diberikan oleh kedua tombol ini adalah sama, yaitu baik murid maupun penyedia jasa tutor dapat melakukan pembatalan transaksi. Murid akan mendapatkan hukuman ketika membatalkan transaksi begitu juga sebaliknya apabila penyedia jasa tutor membatalkan transaksi. Gambar 3.15

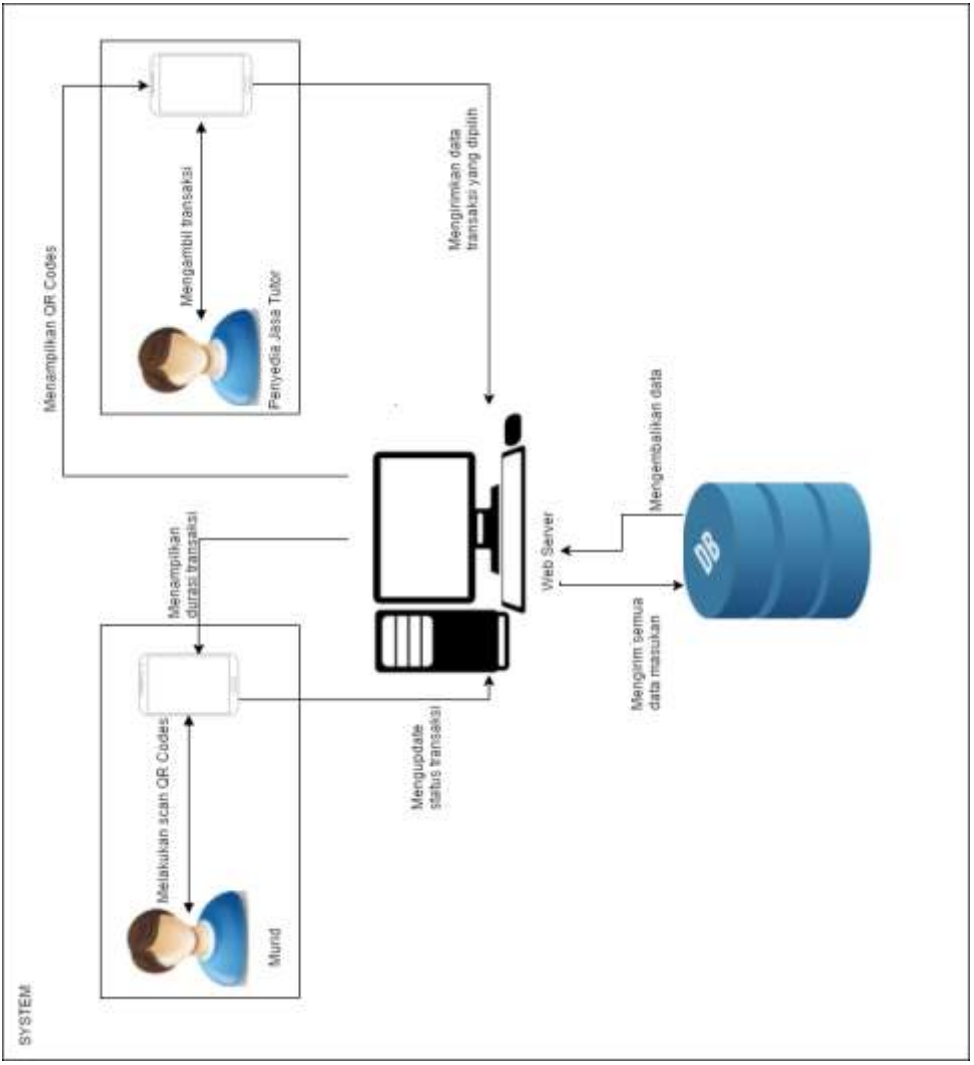
dan Gambar 3.16 menjelaskan rancangan antarmuka dari pembatalan transaksi.

Gambar 3.15 Rancangan Antarmuka Halaman Pembatalan Transaksi Penyedia Jasa Tutor

Gambar 3.16 Rancangan Antarmuka Halaman Pembatalan Transaksi Murid

3.2.3 Perancangan Arsitektur Aplikasi

Aplikasi *Finding Tutor* ini adalah aplikasi berbasis android yang bertujuan untuk memudahkan murid dalam mencari tutor dan juga memudahkan tutor dalam memasarkan jasanya. Namun, agar tidak terjadi *fraud* pada saat melakukan transaksi dibutuhkan sistem validasi, berupa *QR Codes.* Agar sistem validasi ini berjalan semestinya, maka dibutuhkan sebuat *web service* dan *database. Web service* digunakan untuk menghubungkan antara aplikasi pada handphone penyedia jasa tutor ataupun murid dengan *database* dari aplikasi tersebut. *Web service* juga berperan untuk meminta data dan mengirimkan data kepada *database.* Arsitektur sistem dapat dilihat pada Gambar 3.17.



Gambar 3.17 Arsitektur Sistem

BAB IV
IMPLEMENTASI

Bab ini akan membahas implementasi sistem sesuai dengan analisis dan perancangan sistem validasi pada aplikasi *Finding Tutor*. Implementasi yang dijelaskan meliputi lingkungan pembangunan perangkat lunak, kode sumber utama, dan implementasi *library* yang digunakan dalam pengerjaan tugas akhir. Bahasa pemrograman yang digunakan adalah Java dengan tambahan *Volley library* untuk membantu menghubungkan aplikasi pada android dengan *database* dan *Zxing library* untuk membuat *QR Codes.*

4.1 Lingkungan Implementasi

Implementasi penggunaan *QR Codes* untuk mendeteksi fraud pada proses transaksi antara penyedia jasa tutor dengan konsumen jasa tutor pada aplikasi finding – tutor berbasis android menggunakan spesaifikasi perangkat keras dan perangkat lunak seperti yang ditunjukkan pada Tabel 4.1.

Tabel 4.1 Spesifikasi lingkungan implementasi

Perangkat	Jenis Perangkat	Spesifikasi
Perangkat Keras	Prosesor	Prosesor Intel® Core™ i5-CPU (2.60 GHz)
	Memori	4 GB 1600
Perangkat Lunak	Sistem Operasi	Windows 10 Pro
	Perangkat Pengembang	Android Studio

4.2 Implementasi

Pada subbab implementasi akan menjelaskan bagaimana pembangunan perangkat lunak secara detail dan menampilkan kode sumber yang digunakan mulai dari fitur pembuatan *QR Codes*, mekanisme scan *QR Codes,* hingga pembatalan transaksi. Ketiga fitur ini memiliki subbab yang sama yaitu subbab pertama berisi kode sumber pada aplikasi di Android Studio. Subbab kedua berisi kode sumber *backend* aplikasi, dan subbab ketiga berisi implementasi antarmuka dari fitur tersebut.

4.2.1 Implementasi Pembuatan *QR Codes*

Seperti yang telah dijelaskan pada bab perancangan sistem, pihak penyedia jasa tutor setelah mengambil transaksi yang terdapat pada aplikasi *Finding Tutor*, aplikasi *Finding Tutor* secara otomatis akan men – *generate* data berupa id transaksi yang diambil dan username penyedia jasa tutor yang mengambil transaksi tersebut. Pada subbab ini, terdapat tiga subbab baru yang menjelaskan impelementasi kode program aplikasi pada android studio dalam melakukan *generate QR Codes*, implementasi tampilan pada aplikasi *Finding Tutor*, dan implementasi kode program *backend* yang menghubungkan database aplikasi *Finding Tutor* dan kode program pada android studio.

A. Implementasi Kode Program dalam Melakukan *Generate QR Codes* pada Android Studio

Untuk merubah data yang telah disiapkan kedalam sebuah *QR Codes* pada Android studio, hal pertama yang dilakukan adalah menambahkan library *ZXING* ke dalam *build.gradle* pada project yang telah dibuat. Kode Sumber 4.1 menjelaskan cara menambahkan *ZXING* library ke dalam *build.gradle* aplikasi yang dibuat. *ZXING* library ini digunakan sebagai library utama dalam

pembuatan *QR Codes* ataupun dalam *scan QR Codes. Zxing Library* yang digunakan adalah *ZXING* dengan tipe 3.2.1.

```
compile 'com.google.zxing:core:3.2.1'
compile 'com.journeyapps:zxing-android-
embedded:3.2.0@aar'
```

Kode Sumber 4.1 Penambahan ZXING library ke dalam build.gradle Aplikasi

Setelah menambahkan kode program pada Kode Sumber 4.1 kedalam *build.gradle* aplikasi yang dibuat, aplikasi pada Android Studio harus di – *restart* agar library yang dimasukkan dapat bekerja. Kemudian untuk merubah data yang telah ada untuk dijadikan sebuah *QR Codes*, penyedia jasa tutor pertama mengambil transaksi yang ada. Kemudian id transaksi yang telah dipilih digabung dengan username dari penyedia jasa tutor tersebut dengan ditambahkan tanda "- "diantara id transaksi yang dipilih dan username penyedia jasa tutor. Id transaksi yang dipilih ini pertama dimasukkan kedalam sebuah *bundle* yang telah dibuat. Lalu, data tersebut dimasukkan ke dalam variabel bernama *QR_CODES* untuk dikirim ke halaman selanjutnya. Kode Sumber 4.2 menjelaskan kode program untuk menampung data yang telah ada kedalam sebuah variabel.

```
protected Map<String, String> getParams() throws
AuthFailureError {
    Map<String, String> params = new HashMap<>();
    params.put("id_pencarian",
String.valueOf(bundle.getInt("id")));
    return params;
}

Intent genIntent = new
Intent(DetilMuridActivity.this,
GeneratorActivity.class);
genIntent.putExtra("QR_CODES",
String.valueOf(bundle.getInt("id"))+"-
"+db.getUsername());
```

```
startActivity(genIntent);
finish();
```

Kode Sumber 4.2 Data Persiapan untuk diubah Menjadi QR Codes

Pada Kode Sumber 4.2 juga terdapat kode program yang menjelaskan pemindahan data variabel *QR_CODES* dari satu *activity* menuju *activity* selanjutnya. *Activity* pada Android Studio memiliki arti yang sama dengan halaman pada website – website. Variabel *genIntent* yang diniliasiasi merupakan variabel dengan tipe *Intent* yang dimaksudkan untuk memindahkan data pada variabel *QR_CODES* dari *activity* – nya sekarang yaitu *DetilMuridActivity* ke halaman lainnya yaitu *GeneratorActivity*. Selanjutnya pada halaman *GeneratorActivity* data pada variabel *QR_CODES* disimpan ke dalam variabel baru bernama *s*. Gambar 4.3 menjelaskan kode program untuk pemindahan data dari variabel *QR_CODES* menuju *s*.

```
s = getIntent().getStringExtra("QR_CODES");
```

Kode Sumber 4.3 Pemindahan Data dari Variabel QR_CODES ke Variabel S

Setelah itu, data pada variabel s dapat di – *generate* menjadi *QR Codes*. Kode Sumber 4.4 menjelaskan kode program untuk mengubah data pada varibel *s* menjadi *QR Codes*. Pada Kode Sumber 4.4 untuk membuat *QR Codes*, pertama, penulis menginisialisasi variabel *multiFormatWriter* dengan tipe data *MultiFormatWriter*. Variabel ini digunakan untuk memanggil kelas berupa *MultiFormatWriter* yang berisikan format pada *QR Codes* yang digunakan untuk membuat / meng – *encode QR Codes*.

```
MultiFormatWriter multiFormatWriter = new
MultiFormatWriter();
try{
    BitMatrix bitMatrix =
multiFormatWriter.encode(s,
BarcodeFormat.QR_CODE,200,200);
```

```
    BarcodeEncoder barcodeEncoder = new
BarcodeEncoder();

    Bitmap bitmap =
barcodeEncoder.createBitmap(bitMatrix);
    image.setImageBitmap(bitmap);
}
catch (WriterException e){
    e.printStackTrace();
}
waktu(s);
```

Kode Sumber 4.4 Membuat QR Codes

Pada Kode Sumber 4.4 setelah melakukan inisialisasi pada variabel *multiFormatWriter*, apabila berhasil, maka sistem akan masuk ke dalam fungsi try yang mana akan membuat *BitMatrix* sebagai tempat penampungan data untuk dijadikan *QR Codes.* Lalu, pada kode program *BarcodeFromat*, dimaksudkan untuk membuat data pada variabel *s* menjadi sebuah *QR Codes* dengan ukuran 200 x 200. Setelah itu data akan di – *encode* dan dibuat matrix pada *BarcodeEncoder.* Setelah itu data ditampung menjadi *image* dengan format *bitmap.* Pada kode program *catch*, apabila proses penginisialisasi pada variabel *multiFormatWriter* terjadi *error*, maka proses pembuatan *QR Codes* tidak berhasil. Setelah *QR Codes* terbuat, maka kode program berupa *waktu(s)* akan dijalankan. Kode program ini berfungsi untuk memberikan durasi waktu pada *QR Codes.* Apabila *QR Codes* tidak di – *scan* dalam kurun waktu 2 jam, maka transaksi dianggap batal dan *QR Codes* akan terhapus dari halaman penyedia jasa tutor. Kode Sumber 4.5 menjelaskan hitung mundur waktu untuk durasi *QR Codes*. Pada Kode Sumber 4.5, variabel *countDownTimer* diinisialisasi dengan tipe data *CountDownTimer*. Setelah itu waktu *countdown* diinisialisasi menjadi 120 miliseconds. Setelah itu, variabel *hour* berfungsi untuk membagi waktu yang ada dengan 3600000 untuk didapatkan waktu jam yang ada. Kemudian variabel *sisaHour* berfungsi untuk menyimpan sisa pembagian antara variabel *hour* dengan variabel *millisUntilFinished* yang

telah diinisialisasi awal. Begitu pula untuk variabel *minute*, *sisaMinute*, dan *second*. Fungsi *countDownTimer.start()* berfungsi untuk memulai perhitungan mundur tersebut.

```
private void waktu(final String tes) {
    final CountDownTimer countDownTimer = new
CountDownTimer(120 * 60 * 1000, 1000) {
        @Override
        public void onTick(long
millisUntilFinished) {
            long hour = millisUntilFinished /
3600000;
            long sisaHour = millisUntilFinished %
3600000;

            long minute = sisaHour / 60000;
            long sisaMinute = sisaHour % 60000;


            long second = sisaMinute / 1000;
            if(second%5 == 0) {
                Log.d("jalan","jalan");
                getData(tes,3);
            }
            tombol(tes);
        }
        @Override
        public void onFinish() {
            getData(tes,2);
        }
    }; countDownTimer.start();
```

Kode Sumber 4.5 Countdown Durasi QR Codes

Selanjutnya, pada Kode Sumber 4.5 setelah selesai melakukan perhitungan mundur, maka sistem akan masuk ke fungsi *onFinish*. Pada fungsi ini sistem akan mengirimkan nilai berupa *tes* dan 2 ke dalam fungsi lain yaitu fungsi *getData*. Kode Sumber 4.6 menjelaskan proses yang terjadi pada fungsi *getData*. Pada Kode Sumber 4.6, parameter yang digunakan untuk melalukan validasi apakah *QR Codes* telah di – *scan* oleh murid adalah status pada variabel *status*. Data pada *status* ini diambil dari

database. Apabila *status* yang diambil masih bernilai nol dan nilai *temp* adalah dua, maka *QR Codes* berarti masih belum di – *scan* dan secara otomatis *QR Codes* hangus. Transaksi secara otomatis dibatalkan. Pada awalnya nilai *temp* bernilai satu. Nilai dua pada *temp* didapat dari fungsi sebelumnya yaitu fungsi *onFinish* pada kode program sebelumnya dimana fungsi tersebut mengirimkan nilai dua. Pada Kode Sumber 4.6 juga terdapat variabel *gen_btn*. Variabel ini, yang merupakan variabel bantu dari tombol *transaksi selesai* pada tampilan antarmuka *generator activity*, digunakan oleh penyedia jasa tutor untuk mengakhiri transaksi apabila waktu transaksi telah selesai. Variabel *gen_btn* ini baru dapat berfungsi apabila nilai dari variabel *status* adalah dua. Hal ini dimaksudkan agar penyedia jasa tutor tidak dengan sengaja memberikan data palsu ke sistem bahwa transaksi sudah selesai. Meskipun pada kondisi nyata transaksi masih belum selesai. Agar variabel *status* menjadi dua *QR Codes* penyedia jasa tutor harus di – *scan* sebanyak dua kali. Pada Kode sumber 4.6, apabila *temp* tetap bernilai satu dan *status* pada database telah bernilai dua, maka transaksi dianggap telah selesai dan penyedia jasa tutor dapat mengakhiri transaksi dan variabel *gen_btn* berfungsi. Namun, apabila *status* pada database masih bernilai satu, maka variabel *gen_btn* tidak berfungsi dan apabila penyedia jasa tutor menekan tombol *transaksi selesai* pada tampilannya, maka notifikasi berupa "transaksi masih sedang berjalan" akan muncul pada tampilan aplikasi penyedia jasa tutor.

```
gen_btn = (Button) findViewById(R.id.btnEnd);
private void getData(final String s, final int
temp) {

final String status =
jsonObject.getString("status");

if (status.equals("0") && temp == 2) {
    Toast.makeText(GeneratorActivity.this, "QR
Codes gagal di scan dalam kurun waktu yang telah
ditentukan!", Toast.LENGTH_SHORT).show();
    Toast.makeText(GeneratorActivity.this, "
```

```
Transaksi dibatalkan!",
Toast.LENGTH_SHORT).show();
    finish();
}

else if(temp == 1) {
    if (status.equals("2")) {
        Toast.makeText(GeneratorActivity.this,
"Transaksi telah Selesai Terima Kasih Telah
Menggunakan Aplikasi Ini",
Toast.LENGTH_SHORT).show();
        finish();
    } else {
        Toast.makeText(GeneratorActivity.this,
"Transaksi masih sedang berjalan",
Toast.LENGTH_SHORT).show();
    }
}


if (status.equals("0") && temp==2) {
Toast.makeText(TransaksiMuridActivity.this,
"Pentutor tidak datang dalam kurun waktu yang
telah ditentukan!", Toast.LENGTH_SHORT).show();
Toast.makeText(TransaksiMuridActivity.this, "
Transaksi dibatalkan!",
Toast.LENGTH_SHORT).show();
finish();
```

Kode Sumber 4.6 Validasi Durasi QR Codes

Selain itu, pada halaman murid akan muncul notifikasi bahwa penyedia jasa tutor tidak datang pada waktu yang ditentukan dan transaksi dibatalkan. Pada Kode Sumber 4.6, sistem akan terus mengecek status pada database dan nilai *temp* pada sistem di Android Studio. Apabila status sama dengan nol dan *temp* sama dengan 2 pada Kode Sumber 4.6 akan memunculkan notifikasi "Transaksi dibatalkan."

B. Implementasi Kode Program *Backend* Menggunakan CodeIgniter

Agar aplikasi pada Android Studio dapat mengambil data dari database MySQL, dibutuhkan penghubung antara database dengan aplikasi pada Android Studio. Penghubung ini dinamakan *backend*. *Backend* yang digunakan mengimplementasikan *Framework CodeIgniter*. *Framework* ini membagi dua kode sumber menjadi dua bagian. Bagian pertama yaitu *model* yang berisi kode sumber untuk mengambil data dari database dengan. Selanjutnya, bagian kedua adalah *controller. Controller* ini berisi kode sumber yang mengirimkan data hasil dari *model* ke aplikasi pada Android Studio ataupun sebaliknya. *Controller* juga dapat mengambil data dari aplikasi pada Android Studio untuk dijadikan parameter pada *model.* Kode Sumber 4.7 menjelaskan kode pada *model* untuk mengirim data transaksi yang telah terjadi ke dalam database. Parameter yang dimasukkan berupa id transaksi, username tutor, durasi, dan qrcodes yang didapat dari aplikasi di Android Studio. Kode Sumber 4.7 menjelaskan kode memasukkan data ke database dari *model.*

```
public function createTransaction($idpencarian,
$usernametutor, $durasi, $qrcodes)
{
$query = "INSERT INTO `transaksi`(`id_transaksi`,
`id_pencariantutor`,username_tutor`,`status_trans
aksi`, `durasi_transaksi`, `qr_codes`) VALUES
(NULL,'$idpencarian','$usernametutor',0,'$durasi'
,'$qrcodes')";


if($this->db->query($query))
{

  return true;
}

else
{
```

```
  return false;
}
}
```

Kode Sumber 4.7 Input Data Transaksi ke Database

Parameter dari *model* tersebut, didapatkan dari data yang dikirimkan oleh aplikasi pada Android Studio menuju *controller*. Kode Sumber 4.8 menjelaskan proses pengambilan data dari aplikasi pada Android Studio dan mengirimkannya ke *model*. Pada Kode Sumber 4.8, fungsi *isset(_POST[])* berguna untuk menampung data dari Android Studio. Data yang di tampung adalah id pencarian, username tutor, durasi, dan qr_codes. Setelah itu data akan dikirimkan ke *model* dengan nama fungsinya adalah *createTransaction*. Apabila berhasil, maka sistem akan mengembalikan pesan berisi “Transaction registered succesfully”. Apabila gagal, sistem akan mengembalikan pesan berisi “Required fileds are missing”.

```
if(isset($_POST['id_pencarian']) and
isset($_POST['username_tutor'])
                            and
isset($_POST['durasi']) and
                isset($_POST['qr_codes']))
               {
               $result = $this->Transaksi_model-
>createTransaction($_POST['id_pencarian'],

       $_POST['username_tutor'],
                            $_POST['durasi'],
                            $_POST['qr_codes']);
                      if($result)
                        {
                            $respon['error'] =
false;
                            $respon['message'] =
"Transaction registered successfully";
                        }
               }
                   else
                   {
```

```
                        $respon['error'] = true;
                        $respon['message'] =
"Required fields are missing";
                    }
```

Kode Sumber 4.8 Data yang ditampung dari Android Studio ke Controller

Untuk menghubungkan antara *backend* dengan aplikasi pada Android Studio, penulis menggunakan library yaitu *Volley*. Agar library ini dapat digunakan, terlebih dahulu library tersebut dimasukkan ke dalam *build.gradle* pada aplikasi di Android Studio. Kode Sumber 4.9 menjelaskan cara menambahkan library *Volley* pada *build-gradle.*

```
compile 'com.android.volley:volley:1.0.0'
```

Kode Sumber 4.9 Menambahkan Library Volley

Kemudian, untuk menjalankan durasi *QR Codes* pada aplikasi di Android Studio, dibutuhkan nilai status. Pengambilan status sesuai dengan *QR Codes* penyedia jasa tutor tersebut, agar sistem mengetahui apakah *QR Codes* telah di – scan oleh murid, sistem mengambil data status dari database. Untuk itu diperlukan *backend* didalamnya. Kode sumber 4.10 menjelaskan kode program pada *model* untuk mengambil nilai sesuai dengan *QR Codes* penyedia jasa tutor. Parameter pada kode program tersebut adalah *QR Codes*, setelah itu diambil seluruh data pada tabel transaksi di database dimana nilai *QR Codes* di database sama dengan nilai *QR Codes* parameter.

```
<?php
        class TimeServer_model extends CI_Model
        {
                function __construct()
                {
                        parent::__construct();
                }
```

```
                public function
getDurasiByTransaksi($qr_codes)
                {
                        $query = "SELECT * FROM
transaksi WHERE qr_codes = '$qr_codes'";
                        $hasil = $this->db-
>query($query);
                        return $hasil->result();
                }
        }
?>
```

Kode Sumber 4.10 Model dalam Mengambil Data di Database Sesuai QR Codes Penyedia Jasa Tutor

Kemudian setelah mendapatkan data transaksi dari database, data tersebut dikirimkan ke aplikasi pada Android Studio melewati *controller*. Kode Sumber 4.11 menjelaskan proses pengiriman data menuju Android Studio. Setelah mengambil *QR Codes* dan dimasukkan ke dalam variabel isset($_POST['qr_codes']), variabel *$user* memanggil data pada *model* pada fungsi *getDurasiByTransaksi*. Kemudian fungsi *getDurasiByTransaksi* mengembalikan nilai data transaksi dan diambil data *status* dan dimasukkan ke dalam variabel *$current_date['status']*. Setelah itu variabel *$current_date* dikirim menuju aplikasi di Android Studio dan diubah tipe datanya menjadi *JSONObject*.

```
<?php
        class TimeServer_controller extends
CI_Controller
        {
                function __construct()
                {
                        parent::__construct();
                        $this->load-
>model('TimeServer_model');
                }
                public function index()
                {
                        $response = array();
```

```
                            $current_date = array();

         if($_SERVER['REQUEST_METHOD']=='POST')
                            {

         if(isset($_POST['qr_codes']) )
                                        {
                                                    $user =
$this->TimeServer_model-
>getDurasiByTransaksi($_POST['qr_codes']);
                                                    foreach
($user as $row ) {

         $current_date['status'] = $row-
>status_transaksi;
                                                    }
                                        }
                            }
                            else
                            {

$current_date['error'] = true;

$current_date['message'] = "Invalid Request";
                            }
                            echo
json_encode($current_date);

                  }
         }
?>
```

Kode Sumber 4.11Controller dalam Mengirimkan Nilai Status

Lalu, agar halaman dalam Android Studio memanggil fungsi pada *controller* yang benar, sebelumnya harus terdapat variabel yang menginisialisasi pemanggilan file *controller* tersebut. Kode Sumber 4.12 menjelaskan variabel pemanggilan file *controller* yang sesuai dengan Kode Sumber 4.11. Variabel yang digunakan adalah variabel *TIMESERVER*. *TIMESERVER* ini akan menjalankan seluruh fungsi – fungsi pada *controller* yang bersangkutan.

```
public static final String TIMESERVER =
ROOT_URL+"TimeServer_controller";
```

Kode Sumber 4.12 Penginisialisasian Variabel untuk Menunjuk Halaman Controller

Selanjutnya library *Volley* dipanggil dan digunakan untuk mengambil data dari *controller*. Kode sumber 4.13 menjelaskan pengaplikasian *Volley* pada Android Studio. Hal pertama yang dilakukan adalah menginisialisasi variabel bernama *stringRequest* dengan tipe data *String Request*. Variabel *stringRequest* ini memanggil variabel *TIMESERVER* yang menunjuk *controller* yang digunakan. Setelah itu, setelah mendapatkan response, variabel *current_data* diinisialisasi dengan tipe data *JSONObject* sesuai dengan tipe data pada *controller*. Kemudian, data status yang dikirimkan dari *controller*, diterima dan ditampung ke dalam variabel *status*.

```
StringRequest stringRequest = new
StringRequest(Request.Method.POST,
Connect.TIMESERVER, new
Response.Listener<String>() {
            @Override
            public void onResponse(String
current_data)
        }, new Response.ErrorListener() {
            @Override
            public void
onErrorResponse(VolleyError error) {
                //    progresDialog.hide();

Toast.makeText(getApplicationContext(),
error.getMessage(), Toast.LENGTH_LONG).show();
            }
        }) {
            @Override
            protected Map<String, String>
getParams() throws AuthFailureError {
                Map<String, String> params = new
HashMap<>();
                params.put("qr_codes", s);
```

```
                return params;
            }
        };

        RequestQueue requestQueue =
Volley.newRequestQueue(this);
        requestQueue.add(stringRequest);
```

Kode Sumber 4.13 Pengaplikasian Volley pada Pembuatan QR Codes

Pada Kode Sumber 4.13 juga terdpat fungsi *getParams* bertipe data Map. Hal ini difungsikan untuk mengirimkan nilai *QR Codes* yang aktif ke database sebagai parameter. Selanjutnya, variabel *requestQueue* yang bertipe data *RequestQueue* berfungsi untuk membuat *Volley Request* yang baru dan menjalankannya.

C. Implementasi Tampilan Antarmuka Fitur *Generate QR Codes*

Pada subbab ini, penulis memasukkan tampilan – tampilan pada aplikasi di Android Studio pada pembuatan *QR Codes*. Pada Gambar 4-1, dijelaskan list transaksi yang tersedia. List transaksi yang tersedia ini sebelumnya disaring sesuai dengan kriteria dan keinginan milik penyedia jasa tutor. Kriteria – kriteria yang digunakan berupa jarak temputh antara penyedia jasa tutor dengan lokasi transaksi yang ingin dilakukan, harga transaksi yang dimunculkan, jenis kelamin dari murid yang melakukan transaksi. Karena dengan begitu penyedia jasa tutor dapat mempertimbangkan dari kriteria tersebut, mana yang paling cocok dan sesuai dengan kriteria dari penyedia jasa tutor. Kemudian, setelah memilih kriteria yang sesuai, secara otomatis sistem akan menampilkan transaksi berdasarkan kriteria dari databse untuk ditampilkan ke dalam halaman aplikasi dari penyedia jasa tutor. Setelah itu, penyedia jasa tutor memilih transaksi yang sesuai dengan kriterianya dan kemudian menekan tombol *detail*. Tombol *detail* akan membuka halaman yang lain yang berisi detail dari transaksi yang tidak dimunculkan pada halaman di Gambar 4.1.

Apabila detail transaksi tidak sesuai dengan harapan penyedia jasa tutor, penyedia jasa tutor dapat kembali ke halaman list transaksi yang tertera pada Gambar 4.1.

Gambar 4.1 List Transaksi yang Tersedia

Setelah penyedia jasa tutor menekan tombol detail, maka detail transaksi akan muncul di halaman penyedia jasa tutor sesuai dengan Gambar 4.2. Pada Gambar 4.2 detail transaksi yang dimunculkan berupa nama murid yang ingin melakukan tutor, alamat dari murid tersebut, jarak dari tempat penyedia jasa tutor sekarang menuju alamat yang tertera pada halaman penyedia jasa tutor, dan durasi transaksi yang ingin dilakukan oleh murid. Selain itu juga terdapat pelajaran apa yang ingin di tutorkan, pelajaran pada ruang lingkup kelas berapa, tanggal, hari, dan jam dari transaksi yang diinginkan oleh murid untuk dilakukan. Terakhir, terdapat informasi mengenai biaya yang diterima oleh penyedia jasa tutor apabila mengambil transaksi tersebut. Semua data tersebut disimpan di database dan ditampilkan di halaman detail

transaksi ini. Apabila penyedia jasa tutor merasa sesuai dengan transaksi tersebut, penyedia jasa tutor dapat menekan tombol *accept* untuk memberitahukan ke sistem bahwa penyedia jasa tutor tersebut setuju untuk mengambil transaksi tersebut.

Gambar 4.2 Detil Transaksi

Setelah penyedia jasa tutor mengambil transaksi tersebut, maka sistem akan berpindah ke halaman yang berisi *QR Codes* yang telah di ubah oleh sistem untuk ditampilkan sebagai acuan penyedia jasa tutor dalam memulai transaksi. Gambar 4.3 menjelaskan halaman sistem menampilkan *QR Codes*. *QR Codes* itu sendiri dihasilkan dari gabungan antara id transaksi yang diambil dan username dari pihak penyedia jasa tutor dengan ditambahkan dengan simbol "-" diantara id transaksi dan username. *QR Codes* ini berfungsi sebagai alat untuk memulai transaksi. Agar tidak terjadi kecurangan dimana penyedia jasa tutor membohongi sistem dengan berpura – pura sudah bertemu dengan pihak murid dan sudah selesai melakukan transaksi tutor. Untuk itu

sebelum transaksi dilakukan, murid harus melakukan *scan QR Codes* milik penyedia jasa tutor untuk memberitahukan ke sistem bahwa baik penyedia jasa tutor maupun pihak murid telah bertemu secara langsung. Setelah itu, transaksi baru dapat dilakukan setelah *QR Code* di – *scan.* Kemudian, apabila sudah dilakukan *scan* transaksi dihitung dimulai dan waktu transaksi akan berjalan. Selain itu pada Gambar 4.3 terdapat tombol *transaksi selesai* apabila transaksi sudah selesai dan tombol *transaksi dibatalkan* apabila transaksi ingin dibatalkan oleh penyedia jasa tutor.

Gambar 4.3 QR Codes yang telah di Ubah Secara Otomatis oleh Sistem

QR Codes yang telah di ubah secara otomatis oleh sistem ini memiliki masa aktif selama satu jam. Apabila *QR Codes* milik penyedia jasa tutor ini tidak di – *scan* dalam kurun waktu satu jam, maka transaksi dianggap batal dan *QR Codes* tersebut hangus secara otomatis dari sistem. Gambar 4.4 menjelaskan halaman *QR Codes* yang telah hangus karena tidak di – *scan* dalam kurun waktu

satu jam. Cara ini dilakukan agar penyedia jasa tutor tidak datang dengan waktu yang tidak tepat dan dapat merugikan pihak murid. Sehingga dibutuhkan durasi atau masa aktif dari *QR Codes* itu sendiri. Setelah *QR Codes* hangus dan terhapus secara otomatis dari sistem, penyedia jasa tutor dianggap gagal dalam mengerjakan transaksi tersebut.

Gambar 4.4 QR Codes Gagal di – Scan

Gambar 4.4 juga menjelaskan bahwa setelah *QR Codes* hangus, halaman pada aplikasi penyedia jasa tutor akan balik menuju halaman list transaksi yang ada dengan ditambahkan notifikasi "*QR Codes* gagal di scan dalam kurun waktu yang telah di tentukan." Kemudian, pada halaman *QR Codes* terdapat tombol transaksi selesai. Tombol ini dimaksudkan untuk pemberitauhan ke sistem bahwa transaksi telah selesai. Sehingga penyedia jasa tutor dapat memberhenitkan transaksi tersebut. Namun untuk menekan tombol *transaksi selesai*, *QR Codes* penyedia jasa tutor diharuskan sudah di – *scan* sebanyak dua kali. Apabila hal ini tidak dilakukan,

maka transaksi dianggap masih sedang berlangsung. Gambar 4.5 menjelaskan transaksi masih sedang berlangsung, tetapi penyedia jasa tutor menekan tombol *transaksi selesai*. Pada Gambar 4.5, notifikasi "Transaksi masih sedang berjalan" akan muncul di halaman penyedia jasa tutor. Hal ini dikarenakan transaksi masih sedang berlangsung, namun penyedia jasa tutor menekan tombol *transaksi selesai*.

Gambar 4.5 Transaksi Masih Berjalan

4.2.2 Implementasi *Scan QR Codes*

Setelah penyedia jasa tutor mengambil list permohonan transaksi dari seorang murid, murid tersebut akan mendapatkan informasi secara lengkap biodata dari penyedia jasa tutor yang mengambil transaksi tersebut. Lalu, murid diharuskan untuk melakukan *scan QR Codes* milik tutor pada tampilan aplikasi pada handphone tutor. Hal ini dilakukan untuk menandakan transaksi akan dimulai dan pihak penyedia jasa tutor dan murid benar – benar

bertemu secara langsung. Setelah murid melakukan *scan QR Codes* untuk yang pertama kalinya, maka aplikasi *Finding Tutor* setelah itu akan berpindah secara otomatis ke halaman perhitungan mundur durasi transaksi. Setelah durasi transaksi habis, aplikasi pada murid akan secara otomatis kembali ke halaman *scan QR Codes* untuk melakukan *scan* kedua kalinya. Tujuannya untuk menandakan transaksi telah selesai.

Pada subbab ini terdapat tiga subbab baru. Subbab pertama berisi penjelasan tentang kode program pada *scan QR Codes* di Android Studio. Kemudian pada subbab kedua, terdapat penjelasan yang berisi kode program *backend* antara database dengan fitur *scan* pada Android Studio. Terakhir, subbab ketiga menjelaskan tampilan antarmuka pada fitur *scan QR Codes.*

A. Implementasi Kode Program *Scan QR Codes* pada Android Studio

Pada subbab ini murid sebelum memulai transaksi diharuskan untuk melakukan *scan QR Codes* yang dimiliki oleh penyedia jasa tutor. Setelah *QR Codes* milik penyedia jasa tutor di – *scan*, transaksi otomatis berjalan dan *countdown* transaksi akan muncul di halaman murid. Untuk dapat melakukan *scan*, aplikasi *Finding Tutor* ini memakai kamera pada android milik murid. Kode Sumber 4.14 menjelaskan tahapan persiapan dalam melakukan *scan QR Codes*. Pada Kode Sumber 4.14 terdapat *command import zxing.integration.android.IntentIntegrator* dan *import zxing.integration.android.IntentResult.* Kedua *command* ini berfungsi untuk memanggil library *zxing* yang telah ditambahkan ke dalam android studio. Selain itu terdapat fungsi yaitu *setDesiredBarcoceFormat(IntentIntegrator.QR_Codes_Type)* yang berfungsi sebagai informasi ke dalam sistem bahwa tipe barcode yang akan di scan berupa *QR Codes.* Selanjutnya, fungsi berupa *setPrompt("scan")* dan *setCameraId(0)* berfungsi sebagai perintah untuk mengakses kamera handphone milik murid. Kemudian, fungsi berupa *setBarcodeImageEnabled(false)*

berfungsi untuk menonaktifkan *QR Codes* yang telah di – *scan* untuk disimpan dan fungsi berupa *initiateScan* berfungsi untuk memulai aktivitas *scan.*

```
import
com.google.zxing.integration.android.IntentIntegr
ator;
import
com.google.zxing.integration.android.IntentResult
;

integrator.setDesiredBarcodeFormats(IntentIntegra
tor.QR_CODE_TYPES);
integrator.setPrompt("Scan");
integrator.setCameraId(0);
integrator.setBeepEnabled(false);
integrator.setBarcodeImageEnabled(false);
integrator.initiateScan();
```

Kode Sumber 4.14 Inisialisasi Fungsi QR Codes

Selanjutnya, pada tampilan aplikasi murid akan otomatis terbuka kamera handphone milik murid. Kode Sumber 4.15 menjelaskan fungsi yang digunakan pada saat melakukan *scan.* Terdapat dua kondisi yaitu kondisi apabila murid membatalkan proses *scan* dan kondisi apabila berhasil. Pada kondisi *if(result.getContent() == null)* yang berarti murid membatalkan proses *scan*, maka akan masuk ke dalam kode sumber selanjutnya yaitu *finish(); Toast.makeText(this, "You canceled the scanning", Toast.LENGTH_LONG).show();*. *Command* ini dimaksudkan bahwa proses scan selesai dan secara otomatis pada layer murid akan muncul notifikasi berupa "*You canceled the scanning".* Kondisi kedua yaitu pada fungsi *else*, setelah *QR Codes* berhasil di – *scan*, maka akan masuk ke dalam fungsi selanjutnya yaitu fungsi *next.*

```
if(result.getContents()==null){
        finish();
        Toast.makeText(this, "You cancelled the
```

```
scanning", Toast.LENGTH_LONG).show();
    }
    else {
        //Toast.makeText(this,
result.getContents(),Toast.LENGTH_LONG).show();
        next(result.getContents());
    }
}
```

Kode Sumber 4.15 Kondisi Ketika Melakukan Scan

Pada fungsi *next* yang dijelaskan pada Kode Sumber 4.16, terdapat dua kondisi sesuai dengan nilai *status* yang diambil pada database. Apabila status yang diambil adalah 1, maka sistem akan menjalankan fungsi *Intent aIntent = new Intent(ReaderActivity.this, CountdownActivity.class).* Fungsi ini berguna untuk memindahkan halaman dari halaman sekarang yaitu halaman *ReaderActivity* ke halaman *CountdownActivity*. Selain itu, halaman ini juga mengirimkan nilai durasi ke halaman *CountdownActivity* dengan menjalakan fungsi *aIntent.putExtra("**durasi**", jsonObject.getString("**durasi**")).* Setelah itu sistem akan menjalankan fungsi *startActivity(aIntent)* yang berarti menjalakan proses pemindahan halaman menjadi halaman *CountdownActivity* dan mengakhiri halaman *ReaderActivity* dengan menjalakan sintaks *finish().*

```
if(jsonObject.getString("status").equals("1")) {
                        Intent aIntent = new
Intent(ReaderActivity.this,
CountdownActivity.class);

aIntent.putExtra("durasi",
jsonObject.getString("durasi"));
                        startActivity(aIntent);
                        finish();
                    } else
if(jsonObject.getString("status").equals("2")){

Toast.makeText(ReaderActivity.this, "Transaksi
telah Selesai Terima Kasih Telah Menggunakan
Aplikasi Ini", Toast.LENGTH_SHORT).show();
```

```
finish();
                        }
```

Kode Sumber 4.16 Kondisi Status Transaksi

Selain itu, pada Kode Sumber 4.16 terdapat kondisi kedua. Apabila status yang diambil dari database sama dengan 2, maka sistem membaca bahwa transaksi telah selesai berlangsung sehingga akan menjalankan fungsi *Toast.makeText(ReaderActivity.**this**, **"Transaksi telah Selesai Terima Kasih Telah Menggunakan Aplikasi Ini"**, Toast.**LENGTH_SHORT**).show(),* yang berarti memunculkan notifikasi bahwa transaksi telah selesai ke halaman aplikasi milik murid. Selain itu, juga terdapat fungsi *finish()* yang digunakan untuk mengakhiri halaman *ReaderActivity*.

Kemudian pada kondisi dimana murid melakukan *scan* pertama kali, Selanjutnya halaman murid akan berpindah ke halaman *countdown* yang akan menampilkan hitung mundur durasi transaksi. Sebelumnya, untuk mengetahui jumlah durasi pada transaksi tersebut, sistem mengirimkan nilai berupa durasi transaksi ke halaman *countdown.* Untuk dapat digunakan, diperlukan sebuah variabel untuk menampung nilai durasi yang dikirimkan. Kode Sumber 4.17 menjelaskan inisialisasi dan sintaks menyimpan nilai durasi yang dikirimkan oleh halaman sebelumnya. Pada Kode Sumber 4.17, variabel *s* menyimpan durasi transaksi dengan adanya fungsi *s = getIntent.getStringExtra("durasi").*

```
String s;
s = getIntent().getStringExtra("durasi");
```

Kode Sumber 4.17 Inisialisasi Variabel untuk Menyimpan Durasi Transaksi

Kemudian, untuk melakukan perhitungan mundur, digunakan fungsi *countdownTimer*. Sebelumnya, dilakukan inisialisasi terlebih dahulu sebagai tempat penampung untuk menampilkan waktu yang tersisa. Kode Sumber 4.18 menjelaskan

tahapan sintaks dalam melakukan perhitungan mundur. Hal pertama yang dilakukan adalah menginisialisasi variabel yaitu *textView* yang berisi nilai dari variabel *s* dengan menjalankan fungsi *textView.setText(s)*. Selanjutnya, nilai dari variabel *textView* dimasukkan kedalah fungsi *CountDownTimer* pada fungsi *CountDownTimer countDownTimer = **new** CountDownTimer(Integer.parseInt(**textView**.getText().toString()) *60*1000, 1000)*. Setelah itu, fungsi selanjutnya yang dijalankan adalah fungsi *onTick* yaitu menjankan proses hitung mundur. Variabel *hour* berfungsi untuk membagi waktu yang ada dengan 3600000 untuk didapatkan waktu jam yang ada. Kemudian variabel *sisaHour* berfungsi untuk menyimpan sisa pembagian antara variabel *hour* dengan variabel *millisUntilFinished* yang telah diinisialisasi awal. Begitu pula untuk variabel *minute*, *sisaMinute*, dan *second*. Fungsi *countDownTimer.start()* berfungsi untuk memulai perhitungan mundur tersebut. Hasil dari seluruh pembagian jam, menit, detik tersebut akan ditampilkan di halaman murid pada variabel *textView* dengan menjalankan sintaks *textView.setText(hour* hour+":"+minute+":"+second).

```
textView = (TextView)
findViewById(R.id.textView6);

textView.setText(s);

CountDownTimer countDownTimer = new
CountDownTimer(Integer.parseInt(textView.getText(
).toString())*60*1000, 1000) {
            @Override
            public void onTick(long
millisUntilFinished) {
                long hour =
millisUntilFinished/3600000;
                long sisaHour =
millisUntilFinished%3600000;

                long minute = sisaHour/60000;
                long sisaMinute = sisaHour%60000;
```

```
                long second = sisaMinute/1000;


textView.setText(hour+":"+minute+":"+second);
            }

            @Override
            public void onFinish() {

Toast.makeText(CountdownActivity.this, "Waktu
Transaksi telah Selesai",
Toast.LENGTH_SHORT).show();
                Intent tIntent = new
Intent(CountdownActivity.this,
ReaderActivity.class);
                startActivity(tIntent);
                finish();
            }
        };
        countDownTimer.start();
    }
}
```

Kode Sumber 4.18 Tahapan Sintaks dalam Melakukan Perhitungan Mundur

Pada Kode Sumber 4.18, terdapat kondisi ketika waktu telah selesai yaitu pada fungsi *onFinish()*. Fungsi ini akan menampilkan notifikasi berupa waktu transaksi telah selesai pada sintaks *Toast.makeText(CountdownActivity.**this**, **"Waktu Transaksi telah Selesai"**, Toast.**LENGTH_SHORT**).show()*. Selain itu juga terdapat sintaks yang akan mengembalikan halaman menuju halaman *ReaderActivity* setelah selesai durasi transaksi dengan menjalankan sintaks *Intent(CountdownActivity.**this**, ReaderActivity.**class**); startActivity(tIntent)*. Dengan menjalankan sintaks tersebut, sistem akan secara otomatis kembali ke halaman sebelumnya dan murid dapat melakukan *scan QR Codes* kembali untuk menandakan transaksi telah selesai.

B. Implementasi Kode Program *Backend* pada Fitur *Scan QR Codes*

Backend yang digunakan pada fitur ini berupa penambahan status transaksi sebesar 1 secara otomatis apabila *QR Codes* di – *scan.* Selain itu, terdapat pula fungsi pada *backend* yang digunakan untuk mengambil status transaksi pada *QR Codes* yang di – scan. Kode Sumber 4.19 menjelaskan sintaks *backend* pada *model* untuk melakukan update status transaksi. Parameter yang digunakan pada Kode Sumber 4.19 adalah *QR Codes* yang di – *scan* yang di kirimkan dari aplikasi pada Android Studio. Setelah itu akan menjalankan sintaks *$query = "UPDATE `transaksi` SET `status_transaksi`= `status_transaksi`+1 WHERE `qr_codes` = '$qr_codes'".* Sintaks ini berfungsi untuk memperbaharui nilai status transaksi ditambahkan 1 dengan kondisi *QR Codes* pada database sama dengan *QR Codes* pada parameter.

```
public function updateStatus($qr_codes)

                  {

$query = "UPDATE `transaksi` SET
`status_transaksi`= `status_transaksi`+1 WHERE
`qr_codes` = '$qr_codes'";

if($this->db->query($query))

                            {
                                     return true;
                            }

else
                            {
                                     return false;
                            }
}
```

Kode Sumber 4.19 Update Status Transaksi

Selain itu, fungsi lainnya pada *model* di *backend* yaitu fungsi untuk mengambil durasi transaksi sesuai dengan *QR Codes* parameter. Kode Sumber 4.20 menjelaskan sintaks mengambil durasi dari database. Sintaks *$query = "SELECT * FROM transaksi WHERE qr_codes = '$qr_codes'",* berfungsi untuk mengambil semua data dari database dimana *QR Codes* pada database sama dengan *QR Codes* dari parameter. Setelah itu diambil nilai atau hasilnya dengan menjalankan sintaks *$hasil = $this->db->query($query).* Terakhir dikirimkan menuju *controller* dengan menjalankan sintaks *return $hasil->result().*

```
public function getDurasiByTransaksi($qr_codes)
                {
                                $query = "SELECT * FROM
transaksi WHERE qr_codes = '$qr_codes'";
                                $hasil = $this->db-
>query($query);
                                return $hasil->result();
                }
```

Kode Sumber 4.20 Mengambil Durasi Transaski

Selanjutnya, data yang telah diambil di database akan di olah di *controller backend*. Kode Sumber 4.21 menjelaskan sintaks mengambil durasi transaksi dari data yang telah dikirimkan oleh *model*. Pertama *controller* menjalankan file *model TransaksiDurasi_model* pada sintaks *$this->load->model('TransaksiDurasi_model').* Setelah itu, dengan parameter *QR Codes* yang diambil dari aplikasi pada Andorid Studio dan dipanggil dengan menggunakan sintaks *isset($_POST['qr_codes']),* sistem menjalankan fungsi *UpdateStatus* pada fungsi di *model* dengan sintaks *$this->TransaksiDurasi_model->updateStatus($_POST['qr_codes']).* Setelah berhasil, Sistem akan menjalankan sintaks *getDurasiByTransaksi* dengan parameter *QR Codes*. Kemudian, hasil yang didapat dari *model*, akan dimasukkan kedalam variabel *$response* dan pada kasus ini data yang diambil berupa durasi dan

status transaksi dengan menjalankan sintaks *$response['durasi'] = $row->durasi_transaksi* untuk mengambil durasi transaksi dan sintaks $response['status'] = $row->status_transaksi untuk mengambil status transaksi.

```
class TransaksiDurasi_controller extends
CI_Controller
        {
                function __construct()
                {
                        parent::__construct();
                        $this->load-
>model('TransaksiDurasi_model');
                }
                public function index()
                {
                        $this->load->database();

        if(isset($_POST['qr_codes']) )
                        {


$this->TransaksiDurasi_model-
>updateStatus($_POST['qr_codes']);


$user = $this->TransaksiDurasi_model-
>getDurasiByTransaksi($_POST['qr_codes']);
                        if($user){
                        foreach ($user as $row )
{


$response['durasi'] = $row->durasi_transaksi;

$response['status'] = $row->status_transaksi;

            echo json_encode($response);
                }
        }
```

Kode Sumber 4.21 Controller Pengolahan Data Transaksi dari Model

Setelah didapatkan nilainya, maka kedua nilai tersebut akan dikirimkan ke dalam aplikasi pada Android Studio untuk diolah. Untuk memanggil *controller* yang sesuai, ditambahkan sintaks konfigurasi di Android Studio yang dijelaskan pada Kode Sumber 4.22. Variabel *GETDATA* berisikan sintaks dalam memanggil *controller* yang telah dibuat. Pada Kode Sumber 4.22 Variabel *GETDATA* memanggil *file* TransaksiDurasi_controller yang telah disimpan dan dibuat sebelumnya.

```
public static final String GETDATA =
ROOT_URL+"TransaksiDurasi_controller";
```

Kode Sumber 4.22 Konfigurasi Memanggil Controller pada Android Studio

Selanjutnya, dengan memang tambahan library *Android Volley*, dan memanggil konfigurasi pada variabel *GETDATA* dan dimasukkan ke dalam *stringRequest* pada sintaks *StringRequest stringRequest = **new** StringRequest(Request.Method.**POST**, Connect.**GETDATA**, **new** Response.Listener<String>()* yang dijelaskan pada Kode Sumber 4-23. Kemudian didapatkan *respone* berupa *jsonObject* yang berisi nilai dalam variabel *response*. Untuk memanggil data yang dibutuhkan yaitu status, status dan durasi, digunakan sintaks *jsonObject.getString("**status**")* untuk mendapatkan nilai status dan sintaks *jsonObject.getString("**durasi**")* untuk mendapatkan nilai durasi. Akan tetapi, sebelum mendapatkan nilai status dan durasi, terlebih dahulu mengirimkan nilai *QR Codes* yang aktif dengan menggunakan sintaks *params.put("**qr_codes**", contents).*

```
StringRequest stringRequest = new
StringRequest(Request.Method.POST,
Connect.GETDATA, new Response.Listener<String>()
{
            @Override
            public void onResponse(String
response) {
```

```
                    try {
                        JSONObject jsonObject = new
JSONObject(response);

if(jsonObject.getString("status").equals("1")) {
aIntent.putExtra("durasi",
jsonObject.getString("durasi"));
                        } else
if(jsonObject.getString("status").equals("2")){
                            finish();
                        }
                    } catch (JSONException e) {
                        e.printStackTrace();
                    }
                }
            }, new Response.ErrorListener() {
                @Override
                public void
onErrorResponse(VolleyError error) {

Toast.makeText(getApplicationContext(),
error.getMessage(), Toast.LENGTH_LONG).show();
                }
            }){
                @Override
                protected Map<String, String>
getParams() throws AuthFailureError {
                    Map<String, String> params = new
HashMap<>();
                    params.put("qr_codes", contents);
                    return params;
                }
            };
            RequestQueue requestQueue =
Volley.newRequestQueue(this);
            requestQueue.add(stringRequest);
```

Kode Sumber 4.23 Implementasi Pemakaian Backend pada Android Studio untuk Mendapatkan Nilai Status dan Durasi

C. Implementasi Tampilan Antarmuka Fitur *Scan QR Codes*

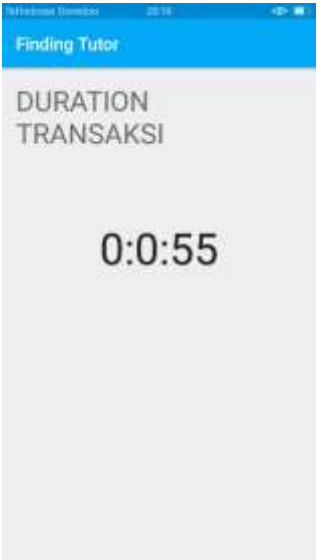
Subbab ini berisi tampilan – tampilan pada aplikasi *Finding Tutor* dalam hal melakukan scan *QR Codes* milik penyedia jasa tutor hingga menampilkan *countdown* durasi transaksi. Gambar 4.6 menjelaskan halaman detil transaksi pada halaman murid yang berisi data penyedia jasa tutor yang mengambil transaksi, beserta tombol *scan.*

Gambar 4.6 Detil Penyedia Jasa Tutor

Setelah murid menekan tombol *scan*, maka aplikasi *Finding Tutor* akan secara otomatis menggunakan kamera handphone milik tutor untuk melakukan *scan.* Gambar 4.7 menjelaskan tampilan ketika murid melakukan *scan QR Codes.* Sistem akan membuka kamera pada handphone milik murid untuk melakukan *scan QR Codes.* Setelah itu sistem akan membaca *QR Codes* milik penyedia jasa tutor seperti yang dijelaskan pada Gambar 4.7

Gambar 4.7 Proses Scan QR Codes

Lalu, setelah *QR Codes* berhasil di – *scan*, sistem secara otomatis akan berpindah ke halaman hitung *countdown* durasi transaksi yang sedang berlangsung. Gambar 4.8 menjelaskan tampilan halaman *countdown* durasi. Durasi *countdown* pada Gambar 4.8 ini sesuai dengan durasi transaksi pada *database*.



Gambar 4.8 Countdown Durasi Transaksi

4.2.3 Implementasi Pembatalan Transaksi

Fitur terakhir pada Tugas Akhir ini adalah pembatalan transaksi. Pada fitur ini baik murid ataupun penyedia jasa tutor dapat melakukan pembatalan transaksi. Murid dapat melakukan pembatalan transaksi apabila tutor telah mengambil transaksi oleh murid tersebut. Namun, setelah murid setuju melakukan pembatalan transaksi, murid akan dikenakan denda tambahan sebesar Rp20.000,00 pada transaksi selanjutnya. Begitu juga sebaliknya pada penyedia jasa tutor. Penyedia jasa tutor dapat membatalkan transaksi meskipun tutor telah mengambil sebuah transaksi. Penyedia jasa tutor pun akan mendapatkan sanksi. Sanksi yang diberikan kepada penyedia jasa tutor adalah penyedia jasa tutor tidak dapat mengambil transaksi selama satu menit.

Pada subbab ini terdapat tiga subbab baru. Subbab tersebut adalah implementasi kode program pembatalan transaksi pada Android Studio, implementasi kode program *backend* antara database dengan fitur pembatalan transaksi pada Android Studio, dan implementasi tampilan antarmuka.

A. Implementasi Kode Program Pembatalan Transaksi pada Android Studio

Pembatalan transaksi sendiri dibedakan menjadi dua jenis berdasarkan pengguna aplikasi. Pengguna pertama yaitu, penyedia jasa tutor. Penyedia jasa tutor dapat melakukan pembatalan transaksi setelah memilih transaksi yang ada. Kode Sumber 4.24 menjelaskan sintaks dalam melakukan pembatalan transaksi. Pembatalan transaksi dimulai ketika penyedia jasa tutor menekan tombol *transaksi dibatalkan.* Setelah itu sistem akan menjalankan sintaks *batal_btn.setOnClickListener*. Setelah itu sistem akan menjalankan sintaks membuat *AlertDialog* baru dengan nama variabel *altdial* yang berisikan notifikasi berupa “Apakah Anda Yakin?”.

```
batal_btn.setOnClickListener(new
View.OnClickListener() {
    @Override
    public void onClick(View view) {
        AlertDialog.Builder altdial = new
AlertDialog.Builder(GeneratorActivity.this);
        altdial.setMessage("Apakah Anda
Yakin?").setCancelable(false)
                .setPositiveButton("Ya", new
DialogInterface.OnClickListener()
```

Kode Sumber 4.24 Pembatalan Transaksi

Pada layar penyedia jasa tutor kemudian terdapat dua pilihan berupa “Ya” atau “Tidak”. Kode Sumber 4.25 menjelaskan sintaks proses pilihan tombol tersebut. Apabila penyedia jasa tutor memilih tombol “Ya”, maka sistem akan mengatur sintaks menjadi *.setPositiveButton("Ya", new DialogInterface.OnClickListener()* dan memberhentikan fungsi *countdown* yang berupa masa aktif *QR Codes* dengan sintaks *countDownTimer.cancel()*. Setelah itu, sistem akan memanggil fungsi pada *backend* dan akan melakukan perubahan data pada database lokal di Android Studiod dengan mengirimkan nilai berupa *“punish”*. Sintaks yang dijalankan adalah *db.updateFlag("punish")*. Selain itu, sistem juga akan mengirimkan nilai berupa *QR Codes*, dan jenis user menuju ke *backend*. Sintaks yang dijalankan adalah *params.put("qr_codes", s)* untuk mengirimkan nilai berupa *QR Codes* dan sintaks *params.put("jenis_user",db.getJenis())* untuk mengirimkan nilai berupa jenis user. Apabila, penyedia jasa tutor menekan tombol “Tidak”, maka sistem akan mengatur sintaks menjadi *.SetNegativeButton(“Tidak”)* dan akan mematikan notifikasi pilihan pembatalan transaksi. Sintaksnya berupa *dialogInterface.cancel()*.

```
.setPositiveButton("Ya", new
DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface
dialogInterface, int i) {
```

```
                StringRequest stringRequest = new
StringRequest(Request.Method.POST,
Connect.CANCELTRANSAKSI, new
Response.Listener<String>() {
                    @Override
                    public void onResponse(String
response) {
                        try {
                            final JSONObject
jsonObject = new JSONObject(response)
db.updateFlag("punish");
Toast.makeText(GeneratorActivity.this,
jsonObject.getString("message"),
Toast.LENGTH_SHORT).show();
                        } catch (JSONException e)
{
                            e.printStackTrace();
                        }
                    }
                },new Response.ErrorListener() {
                    @Override
                    public void
onErrorResponse(VolleyError error) {
Toast.makeText(getApplicationContext(),
error.getMessage(), Toast.LENGTH_LONG).show();
                    }
                }) {
                    @Override
                    protected Map<String, String>
getParams() throws AuthFailureError {
                        Map<String, String>
params = new HashMap<>();
                        params.put("qr_codes",
s);

params.put("jenis_user",db.getJenis());
                        return params;
                    }
                };
                RequestQueue requestQueue =
Volley.newRequestQueue(GeneratorActivity.this);
                requestQueue.add(stringRequest);

                countDownTimer.cancel();
```

```
                    finish();
                }
            })
            .setNegativeButton("Tidak", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface
dialogInterface, int i) {
                    dialogInterface.cancel();
                }
            });
AlertDialog alert = altdial.create();
alert.show();
```

Kode Sumber 4.25 Autentifikasi Pembatalan Transaksi

Setelah penyedia jasa tutor menekan tombol pembatalan transaksi, secara otomatis melalui layar android murid akan muncul notifikasi bahwa penyedia jasa tutor membatalkan transaksi. Kode Sumber 4.26 menjelaskan notifikasi yang muncul pada halaman murid. Pertama sintaks, mengambil nilai pada variabel *status* pada database dan nilai variabel pada *temp*. Apabila nilai *status* adalah *cancel* yang menandakan transaski di cancel dan *temp* bernilai 3, akan muncul notifikasi bahwa transaksi telah dibatalkan. Setelah itu sistem akan menjalankan sintaks *hapusTransaksi(qrcodes)* yang digunakan untuk memanggil *backend* untuk menghapus mengembalikan transaksi menjadi belum diambil oleh penyedia jasa tutor dan menghapus *QR Codes* yang ada.

```
if(status.equals("cancel") && temp==3)
{

AlertDialog.Builder altd = new
AlertDialog.Builder(TransaksiMuridActivity.this);
    altd.setMessage("Apakah Memberatkan
Anda?").setCancelable(false)
            .setPositiveButton("Ya", new
DialogInterface.OnClickListener() {
                @Override
                public void
```

```
onClick(DialogInterface dialogInterface, int i) {
                    hapusTransaksi(qrcode);

Toast.makeText(TransaksiMuridActivity.this,
"Pesanan anda sedang di proses ulang",
Toast.LENGTH_SHORT).show();
                }
            }) .setNegativeButton("Tidak", new
DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface
dialogInterface, int i) {
            hapusTransaksi(qrcode);

Toast.makeText(TransaksiMuridActivity.this,
"Pesanan anda sedang di proses ulang",
Toast.LENGTH_SHORT).show();
        }
    });

    AlertDialog alert = altd.create();
    alert.setTitle("Transaksi Dibatalkan Oleh
Pentutor");
    alert.show();
}
```

Kode Sumber 4.26 Notifikasi pada Halaman Murid

Kemudian, murid juga dapat membatalkan transaksi sama seperti peyedia jasa tutor. Pada halaman detil transaksi yang berisikan detil data penyedia jasa tutor dan tombol scan, juga terdapat tombol pembatalan transaksi. Apabila murid menekan tombol pembatalan transaksi, maka transaksi akan dibatalkan dan dihilangkan dari database transaksi. Serta penyedia jasa tutor akan mendapatkan notifikasi bahwa transaksi dibatalkan oleh murid. Kode Sumber 4.27 menjelaskan sintaks murid dalam membatalkan transaksi. Setelah murid menekan tombol pembatalan transaksi, sistem akan membuat notifikasi baru berupa *alertdialog* yang berisi pertanyaan “Apakah anda yakin?”. Apabila murid menekan tombol ya, maka sistem akan menerima sintaks dan menjalankan sintaks *.setpositivebutton*. Sintaks ini berisi bebarapa sintaks

lainnya seperti *db.updateflag(punish)*. Sintaks ini digunakan untuk mengubah status pada database menjadi *punish* sehingga murid akan terkena biaya tambahan apabila ingin memesan jasa tutor lagi. Apabila murid menekan tombol tidak maka sistem akan menjalankan sintaks *.setNegativeButton* dan transaksi tidak dibatalkan.

```
AlertDialog.Builder altdial = new
AlertDialog.Builder(TransaksiMuridActivity.this);
        altdial.setMessage("Apakah Anda
Yakin?").setCancelable(false)
                .setPositiveButton("Ya", new
DialogInterface.OnClickListener() {
                    @Override
                    public void
onClick(DialogInterface dialogInterface, int i) {
                        StringRequest
stringRequest = new
StringRequest(Request.Method.POST,
Connect.CANCELTRANSAKSI, new
Response.Listener<String>() {
                            @Override
                            public void
onResponse(String response) {
                                try {
                                    final
JSONObject jsonObject = new JSONObject(response);
                                    Log.d("masuk
sini", response);

Toast.makeText(TransaksiMuridActivity.this,
jsonObject.getString("message"),
Toast.LENGTH_SHORT).show();

db.updateFlag("punish");

                                } catch
(JSONException e) {

e.printStackTrace();
                                }
                            }
```

```
.setNegativeButton("Tidak", new
DialogInterface.OnClickListener() {
                    @Override
                    public void
onClick(DialogInterface dialogInterface, int i) {
                        dialogInterface.cancel();
                    }
                });
        AlertDialog alert = altdial.create();
        alert.show();
    }
});
```

Kode Sumber 4.27 Murid Membatalkan Transaksi

Setelah murid menekan tombol pembatalan transaksi, secara otomatis akan muncul notifikasi bahwa transaksi dibatalkan. Kode Sumber 4.28 menjelaskan sintaks dalam memunculkan notifikasi secara otomatis. Setelah sistem menerima bahwa status transaksi pada database berupa "cancel" maka akan muncul secara otomatis notifikasi berupa *alertdialog*. Setelah itu penyedia jasa tutor akan memilih apakah pembatalan memberatkan penyedia jasa tutor atau tidak. Setelah itu sistem akan menjalankan sintaks *hapustransaksi(s)* yang digunakan untuk menghapus transaksi dari sistem.

```
if(status.equals("cancel")){
    AlertDialog.Builder altd = new
AlertDialog.Builder(GeneratorActivity.this);
    altd.setMessage("Apakah Memberatkan
Anda?").setCancelable(false)
            .setPositiveButton("Ya", new
DialogInterface.OnClickListener() {
                @Override
                public void
onClick(DialogInterface dialogInterface, int i) {
                    hapusTransaksi(s);
                    finish();
                }
            }) .setNegativeButton("Tidak", new
DialogInterface.OnClickListener() {
```

```
        @Override
        public void onClick(DialogInterface
dialogInterface, int i) {
            hapusTransaksi(s);
            finish();
        }
    });
    AlertDialog alert = altd.create();
    alert.setTitle("Transaksi Dibatalkan Oleh
Murid");
    alert.show();
}
```

Kode Sumber 4.28 Notifikasi pada Halaman Penyedia Jasa Tutor

B. Implementasi Kode Program *Backend* pada Fitur Pembatalan Transaksi

Backend yang digunakan untuk fitur pembatalan transaksi ini meliputi Kode Sumber 4.29 di mana menjelaskan sintaks pada *backend* untuk merubah status pada transaksi menjadi "cancel." Status transaksi yang akan diubah disesuaikan dengan parameter *QR Codes* yang dikirim dari aplikasi pada Android Studio ke *backend.*

```
public function cancelByTransaksi($qr_codes)
                {
                        $query = "UPDATE
transaksi SET status_transaksi = 'cancel' WHERE
qr_codes = '$qr_codes'";
                        $hasil = $this->db-
>query($query);
                        if($hasil)
                        {
                                return true;
                        }
                        else
                        {
                                return false;
                        }
                }
```

```
$user = $this->CancelTransaksi_model-
>cancelByTransaksi($_POST['qr_codes']);
```

Kode Sumber 4.29 Merubah Status Menjadi Cancel pada Database

Setelah itu apabila pengguna yang melakukan pembatalan transaksi adalah penyedia jasa tutor, maka transaksi pada database akan di set menjadi nol agar transaksi yang di batalkan oleh penyedia jasa tutor dapat muncul di halaman transaksi kembali agar bisa diambil oleh penyedia jasa tutor lainnya. Kode Sumber 4.30 menjelaskan *backend* pada *controller* dalam mengambil kondisi penyedia jasa tutor. Setelah *$_POST['jenis_user']=="Pentutor"* adalah benar, maka sistem akan mengambil id transaksi pada transaksi sesuai dengan *QR Codes* yang di kirimkan dengan menjalakan sintaks *$getId = $this->UpdatePencarianTutor_model->getId($_POST['qr_codes']).* Sintaks tersebut menjalankan fungsi pada *model* untuk mengambil id transaksi. Setelah didapat id transaksi, sistem akan menjalankan sintaks *$aa = $this->UpdatePencarianTutor_model->updatePencarian($id_pencarian)* yang digunakan untuk membalikkan nilai transaksi menjadi nol dengan menjalankan fungsi *updatePencarianTutor* pada *model.*

```
                            $response = array();
                            $user = $this-
>CancelTransaksi_model-
>cancelByTransaksi($_POST['qr_codes']);

                            if
($_POST['jenis_user']=="Pentutor") {
                                        $getId = $this-
>UpdatePencarianTutor_model-
>getId($_POST['qr_codes']);
                                        foreach ($getId
as $key) {

         $id_pencarian = $key->id_pencariantutor;
                                        }
```

```
                                        $aa = $this-
>UpdatePencarianTutor_model-
>updatePencarian($id_pencarian);
                        }

                        $response['error'] =
false;
                        $response['message'] =
"Data has been updated";

                        echo
json_encode($response);
```

Kode Sumber 4.30 Kondisi Penyedia Jasa Tutor Membatalkan Transaksi

Kode Sumber 4.31 menjelaskan *model* yang digunakan untuk mengambil id transaksi dan mengembalikan nilai menjadi nol.

```
public function getId($qr_codes)
                {
                        $query = "SELECT
id_pencariantutor FROM transaksi WHERE
qr_codes='$qr_codes'";
                        $id = $this->db-
>query($query);
                        return $id->result();
                }


public function
updatePencarian($id_pencariantutor)
                {
                        $query = "UPDATE
`pencarian_tutor` SET `status_pencarian` =
0,`biayatutor_pencarian`= 0 WHERE `id_pencarian`
= '$id_pencariantutor'";
                        $hasil = $this->db-
>query($query);
                        if($hasil)
                        {
                                return true;
```

```
                                }
                                else
                                {
                                        return false;
                                }

                        }
```

Kode Sumber 4.31 Mengembalikan Transaksi ke dalam List

Apabila pengguna yang membatalkan transaksi adalah murid, maka transaksi tersebut setelah nilai status diubah menjadi “cancel”, maka transaksi tersebut akan langsung dihapus dari database. Kode Sumber 4.32 menjelaskan proses penghapusan data transaksi dari database. Parameter yang digunakan pada Kode Sumber 4.32 adalah *QR Codes* yang di kirimkan oleh sistem pada Android Studio.

```
public function deleteByTransaksi($qr_codes)
                        {
                                $query = "DELETE FROM
`transaksi` WHERE `qr_codes` = '$qr_codes'";
                                $hasil = $this->db-
>query($query);
                                if($hasil)
                                {
                                        return true;
                                }
                                else
                                {
                                        return false;
                                }
                        }

$user = $this->DeleteTransaksi_model-
>deleteByTransaksi($_POST['qr_codes']);
```

Kode Sumber 4.32 Menghapus Transaksi dari Database

Setelah itu, pada aplikasi android studio, karena pengguna baik murid ataupun penyedia jasa tutor dapat membatalkan transaksi setiap saat, maka sistem harus terus mengecek status

transaksi pada database apakah statusnya *cancel* atau tidak. Kode Sumber 4.33 menjelaskan sintaks yang digunakan oleh aplikasi pada android studio untuk mengambil data. Dengan menggunakan library *Volley* dan memanggil *controller* yang telah dibuat, maka fungsi ambilData ini digunakan untuk mengambil data transaksi pada database.

```
    private void ambilData(final String qrcode,
final int temp){
        Log.d("qrcode", qrcode);
        StringRequest stringRequest = new
StringRequest(Request.Method.POST,
Connect.TIMESERVER, new
Response.Listener<String>() {
new Response.ErrorListener() {
            @Override
            public void
onErrorResponse(VolleyError error) {

Toast.makeText(getApplicationContext(),
error.getMessage(), Toast.LENGTH_LONG).show();
            }
        }) {
            @Override
            protected Map<String, String>
getParams() throws AuthFailureError {
                Map<String, String> params = new
HashMap<>();
                params.put("qr_codes", qrcode);
                return params;
            }
        };
        RequestQueue requestQueue =
Volley.newRequestQueue(this);
        requestQueue.add(stringRequest);

    }
```

Kode Sumber 4.33 Mengambil Data Transaksi dari Database

Kemudian, Kode Sumber 4.34 menjelaskan bahwa fungsi *ambilData* tersebut terus dipanggil setiap 5 detik sekali selama durasi *QR Codes* belum hangus. Hal ini dikarenakan penyedia jasa tutor dan murid dapat menekan tombol pembatalan transaksi kapan saja. Pada Kode Sumber 4.34 fungsi ambilData terus dipanggil setiap 5 menit sekali dengan menjalankan sintaks ***if(second%5==0) { ambilData(qrcode,3) }*** . Parameter yang digunakan pada fungsi ini adalah *QR Codes* yang aktif.

```
private void waktu(final String qrcode) {
    final CountDownTimer countDownTimer = new
CountDownTimer(120 * 60 * 1000, 1000) {
        @Override
        public void onTick(long
millisUntilFinished) {
            long hour = millisUntilFinished /
3600000;
            long sisaHour = millisUntilFinished %
3600000;

            long minute = sisaHour / 60000;
            long sisaMinute = sisaHour % 60000;

            long second = sisaMinute / 1000;

            if(second%5 == 0) {
                Log.d("sini","lala");
                ambilData(qrcode, 3);
            }
        }
```

Kode Sumber 4.34 Pengambilan Data Setiap 5 Detik Sekali

C. Implementasi Tampilan Antarmuka Fitur Pembatalan Transaksi

Pada Subbab ini, terdapat tampilan – tampilan yang digunakan dalam mengimplementasikan pembatalan transaksi. Baik penyedia jasa tutor maupun murid dapat melakukan pembatalan transaksi. Gambar 4.9 menjelaskan tampilan ketika

penyedia jasa tutor menekan tombol pembatalan transaksi. Pertama, penyedia jasa tutor harus menekan tombol *transaksi dibatalkan* apabila ingin membatalkan transaksi yang sedang diambil oleh penyedia jasa tutor tersebut. Setelah itu notifikasi akan muncul seperti pada Gambar 4.9 yang memberikan konfirmasi apabila penyedia jasa tutor ingin membatalkan transaksi. Apabila penyedia jasa tutor menekan tombol "TIDAK", maka transaksi tidak jadi dibatalkan dan masa aktif *QR Codes* milik penyedia jasa tutor tetap berjalan. Namun, apabila penyedia jasa tutor tersebut menekan tombol "Ya", maka transaksi akan dibatalkan dan sistem akan memberikan data ke dalam database bahwa transaksi dibatalkan oleh penyedia jasa tutor.

Gambar 4.9 Penyedia Jasa Tutor Menekan Tombol Pembatalan Transaksi

Setelah penyedia jasa tutor menekan tombol "Ya" maka secara otomatis transaksi yang diambil penyedia jasa tutor dibatalkan dan kemudian secara otomatis akan muncul notifikasi

pada murid bahwa transaksi dibatalkan oleh penyedia jasa tutor. Gambar 4.10 menjelaskan tampilan notifikasi pada halaman murid. Notifikasi tersebut muncul secara otomatis setelah penyedia jasa tutor melakukan konfirmasi bahwa transaksi dibatalkan. Notifikasi yang muncul pada halaman murid berupa pertanyaan yang berisi apakah pembatalan transaksi tersebut memberatkan pihak murid. Baik jawaban ya atau tidak yang dikirimkan oleh pihak murid akan tetap membuat penyedia jasa tutor mendapatkan *punish* dari sistem. Sistem secara otomatis akan memasukkan data ke dalam database bahwa akun penyedia jasa tutor tersebut telah membatalkan transaksi dan dikenakan *punish. Punish* yang diberikan oleh sistem yaitu pihak penyedia jasa tutor tersebut tidak dapat mengambil transaksi selama satu menit.

Gambar 4.10 Notifikasi Penyedia Jasa Tutor Membatalkan Transaksi

Setelah penyedia murid menekan salah satu dari tombol pada Gambar 4.10, transaksi akan di kembalikan ke dalam list

permintaan tutor. Data dalam list permintaan tutor sama dengan data sebelum penyedia jasa tutor mengambil transaksi. Transaksi yang dibatalkan oleh penyedia jasa tutor pun juga kembali muncul di dalam list permintaan tutor. Setelah itu, penyedia jasa tutor akan terkena *punish* dimana penyedia jasa tutor tidak dapat mengambil transaksi dalam satu menit. Setelah satu menit, penyedia jasa tutor baru dapat mengambil transaksi lainnya. Gambar 4.11 menampilkan notifikasi *punish* dari penyedia jasa tutor. Setelah penyedia jasa tutor memilih transaksi dalam list permohonan transaksi, maka sistem seperti sebelumnya akan memunculkan halaman detail transaksi. Namun apabila penyedia jasa tutor masih menjalani status *punish*, maka notifikasi bahwa penyedia jasa tutor tersebut terkena punish akan muncul hingga waktu satu menit berakhir. Setelah satu menit, penyedia jasa tutor baru dapat mengambil transaksi yang lainnya.

Gambar 4.11 Notifikasi Punish

Selanjutnya, apabila murid yang melakukan pembatalan transaksi, maka murid akan menekan tombol seperti yang

dijelaskan pada Gambar 4.12. Tombol yang ditekan adalah tombol *cancel* yang letaknya bersebelahan dengan tombol *scan.* Tombol *cancel* pada Gambar 4.12 ini terletak pada halaman detail biodata dari penyedia jasa tutor yang mengambil transaksi tersebut. Setelah menekan tombol pembatalan transaksi, notifikasi oleh sistem akan muncul di halaman murid untuk memastikan transaksi ingin dibatalkan atau tidak. Apabila murid menekan tombol “TIDAK” maka transaksi tidak jadi dibatalkan dan notifikasi bahwa transaksi dibatalkan tidak muncul di halaman penyedia jasa tutor. Namun, ketika murid menekan tombol “YA”, hal ini berarti murid telah setuju membatalkan transaksi sehinga sistem memasukkan data bahwa transaksi dibatalkan oleh pihak murid.

Gambar 4.12 Murid Membatalkan Transaksi

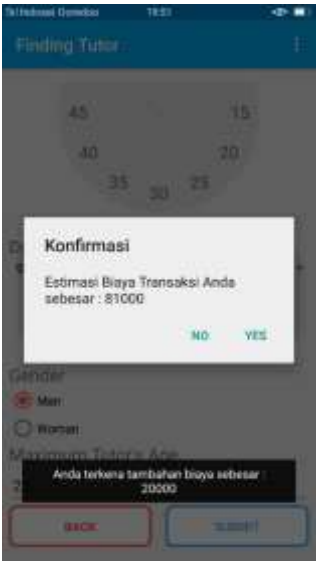
Kemudian, secara otomatis akan muncul di halaman penyedia jasa tutor bahwa murid telah membatalkan transaksi. Gambar 4.13 menunjukkan notifikasi pada halaman penyedia jasa tutor. Notifikasi ini muncul pada halaman penyedia jasa tutor

dimana terdapat *QR Codes.* Setelah transaksi dibatalkan, maka *QR Codes* milik penyedia jasa tutor akan hangus dan tidak berlaku. Kemudian pada halaman aplikasi milik penyedia jasa tutor akan muncul notifikasi yang berisi pertanyaan apakah pembatalan transaksi yang dilakukan oleh pihak murid memberatkan pihak penyedia jasa tutor atau tidak. Baik jawaban ya maupun tidak, murid akan tetap terkena *punish* dari sistem. *Punish* yang diberikan berupa penambahan harga untuk murid ketika murid tersebut ingin membuat transaksi yang baru.

Gambar 4.13 Notifikasi pada Halaman Penyedia Jasa Tutor

Apabila murid yang membatalkan transaksi ingin membuat transaksi yang baru setelah itu, maka murid akan terkena *punish* dimana biaya transaksi yang dikenakan bertambah sebesar Rp20.000. Biaya ini dikenakan oleh sistem setelah sistem mengecek status dari murid tersebut di dalam database lokal milik Android Studio. Apabila status tersebut adalah *punish*, maka sistem akan memberikan *punish* berupa tambahan biaya. Gambar

4.14 menjelaskan tampilan murid yang mengalami *punish*. Setelah murid menuliskan kriteria transaksi yang baru berupa mata pelajaran yang ingin ditutorkan, alamat, tanggal, jam, beserta durasi dari transaksi, ketika murid menekan tombol *submit*, sistem secara otomatis akan memberikan notifikasi bahwa murid tersebut telah membatalkan transaksi pada transaksi sebelumnya. Sistem juga memunculkan notifikasi berupa penambahan biaya sebesar RP20.000. Setelah itu biaya tambahan tersebut akan ditambahkan secara otomatis ke dalam estimasi biaya yang juga ditunjukkan oleh Gambar 4.14. Setelah itu, murid dapat mengonfirmasi persetujuan transaksi.



Gambar 4.14 Murid Terkena Punish

BAB V
UJI COBA DAN EVALUASI

Bab ini akan menjelaskan pengujian dan evaluasi dari fitur pembetukan *QR Codes, Scan QR Codes,* dan pembatalan transaksi pada aplikasi *Finding Tutor*. Pengujian yang dilakukan adalah pengujian dalam bentuk *control test* yang dilakukan pada waktu tertentu. Pengujian memerhatikan terpenuhinya kebuthan fungsionalitas pada setiap fitur yang dibuat serta konsistensi pada setiap kondisi yang dibuat dalam hal melakukan *scan QR Codes.*

5.1 Lingkungan Pengujian

Lingkungan pengujian adalah lingkungan baik perangkat keras maupun perangkat lunak tempat pengujian sistem dilakukan. Lingkungan pengujian pada uji coba fitur *QR Codes* pada penggunaan qr codes untuk mendeteksi *fraud* pada proses transaksi antara penyedia jasa tutor dengan konsumen jasa tutor pada aplikasi *finding – tutor* berbasis android menggunakan spesifikasi keras dan perangkat lunak seperti yang ditunjukkan pada Tabel 5.1.

Tabel 5.1 Spesifikasi Lingkungan Uji Coba

Perangkat	**Jenis Perangkat**	**Spesifikasi**
Perangkat Keras	Tipe Android	Oppo Smartphone F1s
	Memori	3GB
Perangkat Lunak	Sistem Operasi	Android Lolipop 5.1
	Perangkat Pengembang	Android Studio

Pengujian dilakukan dengan proses pertama pembuatan *QR Codes*. Pengujian berhasil apabila *QR Codes* telah berhasil

dibuat dari data berupa id transaksi dan username penyedia jasa tutor. Setelah itu pengujian kedua dilakukan dengan proses *scan QR Codes* milik penyedia jasa tutor. Pengujian berhasil apabila *QR Codes* berhasil di scan dan durasi transaksi dimulai. Terakhir, pengujian yang dilakukan berupa pembatalan transaksi. Pengujian berhasil apabila transaksi berhasil dibatalkan dan pihak yang membatalkan mendapatkan hukuman atau *punishment.*

5.2 Skenario Pengujian Fungsionalitas

Subbab ini menjelaskan mengenai scenario pengujian fungsionalitas yang digunakan. Seperti yang sudah dijelaskan pada Bab 5, metode yang digunakan dalam pengujian ini adalah metode *blackbox.* Metode *blackbox* merupakan metode dimana pengujian ditekankan pada pola *input* dan *output* yang sesuai dalam skenario. Pengujian dilakukan dengan mengacu kepada kasus penggunaan yang dijelaskan pada Bab 3, analisis dan perancangan sistem. Selanjutnya, terdapat pengujian kehandalan *QR Codes* dengan melakukan pengujian sebanyak 5 kali dengan ditambahkan beberapa kondisi.

5.2.1 Pengujian Pembuatan *QR Codes*

Pada pengujian ini penyedia jasa tutor setelah mengambil transaksi yang ada pada list transaksi yang tersedia akan secara otomatis mendapatkan *QR Codes* transaksi yang telah diubah secara otomatis dari sistem. Data yang diubah berupa id transaksi dan username dari penyedia jasa tutor tersebut. Skenario pengujian dalam pembuatan *QR Codes* terdapat pada Tabel 5.2.

Tabel 5.2 Skenario Uji Coba Pembuatan *QR Codes*

ID	UJ-001
Kasus Penggunaan	Menampilkan QR Codes yang telah di – generate

Nama	Menampilkan *QR Codes* yang telah di – generate
Tujuan Pengujian	Menguji fitur pembuatan *QR Codes*
Skenario	Penyedia jasa tutor mendapatkan *QR Codes* dari transaksi yang telah diambil
Kondisi Awal	Penyedia jasa tutor memilih transaksi
Langkah Pengujian	1. Penyedia jasa tutor mengambil transaksi. 2. Sistem mengubah id transaksi dan username penyedia jasa tutor menjadi *QR Codes*.
Kondisi yang diharapkan	*QR Codes* ditampilkan di halaman aplikasi penyedia jasa tutor
Kondisi yang didapat	Sistem menampilkan *QR Codes* yang telah diubah pada halaman penyedia jasa tutor
Hasil Pengujian	Berhasil

Pada skenario uji coba pada Tabel 5.2, sistem berhasil membuat *QR Codes* secara otomatis dan ditampilkan di halaman penyedia jasa tutor. Hasil pengujian dapat dilihat pada Gambar 4.3 dimana *QR Codes* telah berhasil ditampilkan di halaman penyedia jasa tutor.

5.2.2 Pengujian *Scan QR Codes*

Pada pengujian ini, setelah penyedia jasa tutor sampai di lokasi transaksi, murid harus melakukan *scan QR Codes* milik penyedia jasa tutor untuk memulai transaksi dan memberitahukan ke sistem bahwa penyedia jasa tutor sudah berada di tempat transaksi. Mulanya, murid akan melakukan *scan*. Lalu setelah *scan* berhasil, secara otomatis sistem akan memunculkan halaman *countdown* durasi dari transaksi tersebut. Skenario pengujian *scan QR Codes* terdapat pada Tabel 5.3. Tabel 5.3 menjelaskan tahapan ketika penyedia jasa tutor menunjukkan *QR Codes* hingga halaman

countdown durasi transaksi muncul di halaman aplikasi murid secara otomatis.

Tabel 5.3 Skenario Uji Coba *Scan QR Codes*

ID	UJ-002
Kasus Penggunaan	Melakukan *scan QR Codes*
Nama	Melakukan *scan QR Codes*
Tujuan Pengujian	Menguji fitur *scan QR Codes*
Skenario	Murid mendapatkan *countdown* durasi transaksi setelah melakukan *scan QR Codes*
Kondisi Awal	Penyedia jasa tutor datang sebelum masa berlaku *QR Codes* telah habis
Langkah Pengujian	1. Penyedia jasa tutor menunjukan *QR Codes* miliknya. 2. Murid melakukan *scan.* 3. Sistem membaca *QR Codes* yang di – *scan.*
Kondisi yang diharapkan	*Countdown* durasi transaksi muncul di halaman aplikasi murid
Kondisi yang didapat	Setelah murid melakukan *scan,* secara otomatis halaman berpindah ke halaman *countdown* durasi transaksi
Hasil Pengujian	Berhasil

Pada skenario uji coba pada Tabel 5.3, sistem berhasil memunculkan *countdown* durasi transaksi dari transaksi yang mewakili *QR Codes* tersebut. Hasil pengujian dapat dilihat pada Gambar 4.7 dan Gambar 4.8. Gambar 4.7 mewakili proses ketika melakukan *scan QR Codes* milik penyedia jasa tutor. Setelah sistem berhasil membaca *QR Codes* tersebut, sistem akan berpindah ke halaman *countdown* durasi transaksi yang sesuai dengan durasi pada *QR Codes* tersebut. Gambar 4.8 menjelaskan tampilan pada halaman *countdown* durasi transaksi.

5.2.3 Pengujian Pembatalan Transaksi

Pada pengujian ini, baik penyedia jasa tutor maupun murid dapat membatalkan transaksi. Penyedia jasa tutor dapat membatalkan transaksi setelah transaksi diambil oleh penyedia jasa tutor dan dimunculkan pada halaman *QR Codes*. Sedangkan murid dapat membatalkan transaksi pada halaman detil transaksi yang sedang berjalan. Baik murid maupun penyedia jasa tutor setelah melakukan pembatalan transaksi mendapatkan hukuman. Hukuman yang didapatkan oleh penyedia jasa tutor adalah penyedia jasa tutor tidak dapat mengambil transaksi lainnya selama satu menit. Sedangkan murid akan dikenakan biaya tambahan sebesar Rp20.000,00 pada transaksi selanjutnya. Skenario pengujian pembatalan transaksi dijelaskan pada Tabel 5.4.

Tabel 5.4 Skenario Uji Coba Pembatalan Transaksi

ID	UJ-003
Kasus Penggunaan	Membatalkan transaksi
Nama	Membatalkan transaksi
Tujuan Pengujian	Menguji fitur pembatalan transaksi
Skenario	Baik murid maupun penyedia jasa tutor dapat membatalkan transaksi yang sedang berjalan
Kondisi Awal	Transaksi sedang berlangsung
Langkah Pengujian	1. Baik murid ataupun penyedia jasa tutor menekan tombol pembatalan transaksi 2. Transaksi dibatalkan 3. Sistem memunculkan notifikasi bahwa transaksi dibatalkan pada halaman murid apabila penyedia jasa tutor membatalkan transaksi ataupun sebaliknya

	4. Baik murid ataupun penyedia jasa tutor yang membatalkan transaksi mendapatkan hukuman
Kondisi yang diharapkan	Transaksi dibatalkan, pengguna yang membatalkan transaksi mendapatkan hukuman
Kondisi yang didapat	Setelah murid membatalkan transaksi, murid mendapatkan denda. Begitu pula sebaliknya ketika penyedia jasa tutor membatalkan transaksi, penyedia jasa tutor tidak dapat mengambil transaksi selama satu menit
Hasil Pengujian	Berhasil

Pada skenario uji coba pada Tabel 5.4, transaski berhasil dibatalkan dan sistem memberikan hukuman pada pengguna yang membatalkan transaksi. Gambar 4.9, Gambar 4.10, dan Gambar 4.11 menjelaskan hasil uji coba ketika penyedia jasa tutor membatalkan transaksi. Pada Gambar 4.9, setelah penyedia jasa tutor menekan tombol pembatalan transaksi, maka akan muncul notifikasi berupa konfirmasi apakah penyedia jasa tutor akan membatalkan transaksi. Lalu, setelah penyedia jasa tutor membatalkan transaksi, maka notifikasi pembatalan transaksi akan muncul di halaman murid yang dapat dilihat pada Gambar 4.10. Setelah itu, penyedia jasa tutor akan mendapatkan hukuman berupa tidak dapat mengambil transaksi selama satu menit dan dapat dilihat pada Gambar 4.11. Selain itu, hasil pengujian apabila murid membatalkan transaksi dapat dilihat pada Gambar 4.12, Gambar 4.13, dan Gambar 4.14. Sama seperti langkah pada penyedia jasa tutor, setelah murid menekan tombol membatalkan transaksi, sistem akan memunculkan notifikasi konfirmasi pembatalan transaski yang dapat dilihat pada Gambar 4.12. Setelah itu notifikasi pada halaman penyedia jasa tutor bahwa transaksi dibatalkan oleh murid akan muncul dan dijelaskan pada Gambar 4.13. Kemudian, murid akan mendapatkan hukuman berupa denda

tambahan sebesar Rp20.000 pada transaksi selanjutnya yang dijelaskan pada Gambar 4.14.

5.3 Skenario Pengujian Kehandalan *QR Codes*

Pengujian kehandalan *QR Codes* dilakukan terhadap masing – masing kondisi. Kriteria yang diberikan dibagi menjadi tiga jenis. Kriteria yang pertama berupa kondisi rotasi terhadap *QR Codes*. Pada kondisi ini, proses *scan* akan dilakukan dengan merotasi *QR Codes* sebesar 0 derajat, 45 derajat, 90 derajat, dan 180 derajat. Kriteria kedua yang dilakukan berupa pencahayaan ketika melakukan proses *scan.* Pencahayaan ini dibagi menjadi dua jenis, yaitu pencahayaan terang dimana proses *scan* dilakukan pada ruangan yang disinari lampu dan *brightness* pada layar handphone adalah pada kondisi paling terang. Juga pencahayaan remang, dimana kondisi pencahayaan pada saat melakukan *scan* dilakukan pada ruangan yang gelap tidak disiniari lampu, dan *brightness* pada layar handphone adalah pada kondisi setengah terang. Kemudian, kriteria yang terakhir adalah kriteria posisi kamera handphone terhadap *QR Codes.* Kriteria ketiga ini dibedakan menjadi posisi pertama di mana kamera handphone sejajar dengan *QR Codes.* Posisi kedua adalah posisi di mana kamera handphone miring dengan kemiringan 30 derajat terhadap posisi *QR Codes.* Selain itu, ketiga parameter ini akan diuji coba dengan menggunakan tiga jarak yang berbeda. Ketiga parameter ini akan diuji coba dengan jarak 10cm, 20 cm, dan 30 cm untuk mendapatkan jarak yang cocok untuk melakukan *scan QR Codes.* Ketiga kriteria ini akan diukur tingkat keberhasilan serta waktu yang dibutuhkan dari setiap kondisi untuk melakukan *scan.* Hasil pengujian dapat dilihat pada Tabel 5.5.

Tabel 5.5 Hasil Uji Coba Kehandalan dalam Scan QR Codes

Kondisi	Hasil 5 Pengujian	
	Status	Waktu
Rotasi 0 Derajat dengan Jarak 10 cm	Berhasil	6.2525 detik

Rotasi 45 Derajat dengan Jarak 10 cm	Berhasil	8.308 detik
Rotasi 90 Derajat dengan Jarak 10 cm	Berhasil	4.098 detik
Rotasi 180 Derajat dengan Jarak 10 cm	Berhasil	4.356 detik
Pencahayaan Terang dengan Jarak 10 cm	Berhasil	3.214 detik
Pencahayaan Remang dengan Jarak 10 cm	Berhasil	3.092 detik
Posisi dengan QR Code Sejajar dengan jarak 10 cm	Berhasil	3.354 detik
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kanan dengan Jarak 10 cm	Gagal	60 detik
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kiri dengan Jarak 10 cm	Gagal	60 detik
Posisi dengan QR Code Kemiringan 30 derajat dari atas dengan Jarak 10 cm	Gagal	60 detik
Posisi dengan QR Code Kemiringan 30 derajat dari bawah dengan Jarak 10 cm	Gagal	60 detik
Rotasi 0 Derajat dengan Jarak 20 cm	Berhasil	1.91 detik
Rotasi 45 Derajat dengan Jarak 20 cm	Berhasil	2.382 detik
Rotasi 90 Derajat dengan Jarak 20 cm	Berhasil	2.292 detik
Rotasi 180 Derajat dengan Jarak 20 cm	Berhasil	1.918 detik
Pencahayaan Terang dengan Jarak 20 cm	Berhasil	2 detik
Pencahayaan Remang dengan Jarak 20 cm	Berhasil	2.318 detik
Posisi dengan QR Code Sejajar dengan jarak 20 cm	Berhasil	1.314 detik
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kanan dengan Jarak 20 cm	Gagal	60 detik
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kiri dengan Jarak 20 cm	Gagal	60 detik
Posisi dengan QR Code Kemiringan 30 derajat dari atas dengan Jarak 20 cm	Gagal	60 detik

Posisi dengan QR Code Kemiringan 30 derajat dari bawah dengan Jarak 20 cm	Gagal	60 detik
Rotasi 0 Derajat dengan Jarak 30 cm	Berhasil	1.43 detik
Rotasi 45 Derajat dengan Jarak 30 cm	Berhasil	1.848 detik
Rotasi 90 Derajat dengan Jarak 30 cm	Berhasil	1.528 detik
Rotasi 180 Derajat dengan Jarak 30 cm	Berhasil	2.016 detik
Pencahayaan Terang dengan Jarak 30 cm	Berhasil	2.766 detik
Pencahayaan Remang dengan Jarak 30 cm	Berhasil	1.458 detik
Posisi dengan QR Code Sejajar dengan jarak 30 cm	Berhasil	1.506 detik
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kanan dengan Jarak 30 cm	Gagal	60 detik
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kiri dengan Jarak 30 cm	Gagal	60 detik
Posisi dengan QR Code Kemiringan 30 derajat dari atas dengan Jarak 30 cm	Gagal	60 detik
Posisi dengan QR Code Kemiringan 30 derajat dari bawah dengan Jarak 30 cm	Gagal	60 detik

(Halaman ini sengaja dikosongkan)

BAB VI
KESIMPULAN DAN SARAN

Bab ini membahas tentang kesimpulan yang didasari oleh hasil uji coba yang telah dilakukan pada bab sebelumnya. Kesimpulan nantinya sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut di masa depan.

6.1 Kesimpulan

Dalam proses pengerjaan Tugas Akhir ini, setelah melalui tahap perancangan, implementasi, serta uji coba pada fitur *QR Codes* pada Aplikasi *Finding Tutor*, didapatkan kesimpulan sebagai berikut:

1. Aplikasi *Finding Tutor* pada Tugas Akhir ini telah dapat melakukan *generate QR Codes* dari data berupa id transaksi dan username penyedia jasa tutor dengan menggunakan library ZXING yang dapat di implementasikan pada Android Studio.
2. Aplikasi *Finding Tutor* dapat melakukan *scan QR Codes* dari *QR Codes* yang memang sesuai dengan transaksi tersebut dengan mencocokkan data *QR Codes* yang di – *scan* dengan data yang berada di *database*, dengan ditambahkan library ZXING pada Android Studio.
3. *QR Codes* yang telah dibuat oleh sistem memiliki masa aktif satu jam dengan mengimplementasikan fungsi *countdown* di Android Studio dan apabila *countdown* waktu telah selesai dan *QR Codes* belum di – *scan* maka transaksi secara otomatis akan gagal.
4. Aplikasi dapat memanggil halaman lain yang memulai fungsi *countdown* durasi transaksi dengan sebelumnya

melakukan *scan QR Codes* dan kembali melakukan *scan* setelah durasi pada fungsi *countdown* durasi telah selesai.

6.2 Saran

Adapun saran - saran yang diberikan terkait pengembangan pada Tugas Akhir ini adalah:

1. Mengoptimalkan fitur, seperti fitur pembatalan transaksi apabila murid setuju dengan pembatalan transaksi atau sebaliknya apabila penyedia jasa tutor setuju dengan pembatalan transaksi.
2. Dari segi kehandalan *QR Codes*, *QR Codes* masih belum mampu menangani kondisi apabila di – *scan* dengan kemiringan minimal 30^0 sehingga perlu adanya solusi penggunaan fitur lain selain *QR Codes*.
3. Menambahkan fitur notifikasi kepada penyedia jasa tutor apabila masa aktif *QR Codes* hanya tersisa lima menit atau kurang.
4. Menambah fitur lainnya, seperti fitur memperpanjang transaksi apabila transaksi telah selesai dan fitur memastikan bahwa penyedia jasa tutor dan murid hanya melakukan transaksi melalui aplikasi.

DAFTAR PUSTAKA

[1] Pasca Nugraha, Dr. Ir. Rinaldi Munir M.T,2011, "Pengembangan QR Code Generator dan QR Code Reader dari Data Berbentuk Image," Konferensi Nasional Informatika - KNIF 2011.

[2] Luiz F. F. Bellusi, Nina S. T. Hirata,"Fast QR Code Detection in Arbitrarily Acquired Images," *24th SIBGRAPI Conference on Graphics, Patterns and Images, Sibgrapi 2011, Alagoas, Maceió, Brazil, August 28-31, 2011, doi:* 10.1109/SIBGRAPI.2011.16.

[3] Gareth R.T. White, Georgina Gardiner, Guru Prabhakar, Azley Abd Razak,"A Comparison of Barcoding and RFID Technologies in Practice," *Journal of Information, Information Technology and Organizations*, 2. pp. 119-132. ISSN 1557-1319.

[4] Michael Grace, Yajin Zhou, Zhi Wang, Xuxian Jiang,"Systematic Detection of Capability Leaks in Stock Android Smartphones."

[5] Gianluca Segato, "An Introduction to Volley,"13 May 2015. [Online].Available: https://code.tutsplus.com/tutorials/an-introduction-to-volley--cms-23800.[Diakses 7 Mei 2017].

[6] Ficus Kirkpatrick, 2013, "Volley, Easy, Fast Networking for Android."

[7] Micjahn,"Zxing.Net Project Description."[Online].Available: https://zxingnet.codeplex.com.[Diakses 7 Mei 2017].

[8] CodeIgniter Team, "CodeIgniter." [Online]. Available: https://codeigniter.com. [Diakses 6 Mei 2017].

[9] Jayaprakash Garaga, 2015, “Personalized Movie Datatabase System.”

[10] Ken Arnold, James Gosling, David Holmes,"THE Java™ Programming Language, Fourth Edition,"2015, ISBN: 0-321-34980-6.

[11] Business Dictionary Team, "Startup Definition." [Online]. Available: http://www.businessdictionary.com/definition/startup.html.[Diakses 6 Maret 2017].

(Halaman ini sengaja dikosongkan)

LAMPIRAN

Bagian ini merupakan lampiran sebagai pelengkap dari buku Tugas Akhir. Pada bagian ini, terdapat informasi berupa lima tabel pengujian pada uji coba kehandalan *QR Codes*.

A. Percobaan yang digunakan Untuk Uji Coba Kehandalan *QR Codes*

Tabel A.1 Uji Coba 1

Kondisi	Pengujian ke 1	
	Status	Waktu
Rotasi 0 Derajat dengan Jarak 10 cm	Berhasil	9,48 detik
Rotasi 45 Derajat dengan Jarak 10 cm	Berhasil	16.45 detik
Rotasi 90 Derajat dengan Jarak 10 cm	Berhasil	3.17 detik
Rotasi 180 Derajat dengan Jarak 10 cm	Berhasil	3.18 detik
Pencahayaan Terang dengan Jarak 10 cm	Berhasil	4.22 detik
Pencahayaan Remang dengan Jarak 10 cm	Berhasil	3.93 detik
Posisi dengan QR Code Sejajar dengan jarak 10 cm	Berhasil	5.71 detik
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kanan dengan Jarak 10 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kiri dengan Jarak 10 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari atas dengan Jarak 10 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari bawah dengan Jarak 10 cm	Gagal	1 menit
Rotasi 0 Derajat dengan Jarak 20 cm	Berhasil	2.85 detik
Rotasi 45 Derajat dengan Jarak 20 cm	Berhasil	3.20 detik
Rotasi 90 Derajat dengan Jarak 20 cm	Berhasil	1.60 detik
Rotasi 180 Derajat dengan Jarak 20 cm	Berhasil	1.77 detik

Pencahayaan Terang dengan Jarak 20 cm	Berhasil	2.44 detik
Pencahayaan Remang dengan Jarak 20 cm	Berhasil	2.34 detik
Posisi dengan QR Code Sejajar dengan jarak 20 cm	Berhasil	1.31 detik
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kanan dengan Jarak 20 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kiri dengan Jarak 20 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari atas dengan Jarak 20 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari bawah dengan Jarak 20 cm	Gagal	1 menit
Rotasi 0 Derajat dengan Jarak 30 cm	Berhasil	1.27 detik
Rotasi 45 Derajat dengan Jarak 30 cm	Berhasil	2.23 detik
Rotasi 90 Derajat dengan Jarak 30 cm	Berhasil	1.64 detik
Rotasi 180 Derajat dengan Jarak 30 cm	Berhasil	2.67 detik
Pencahayaan Terang dengan Jarak 30 cm	Berhasil	2.51 detik
Pencahayaan Remang dengan Jarak 30 cm	Berhasil	1.47 detik
Posisi dengan QR Code Sejajar dengan jarak 30 cm	Berhasil	1.55 detik
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kanan dengan Jarak 30 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kiri dengan Jarak 30 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari atas dengan Jarak 30 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari bawah dengan Jarak 30 cm	Gagal	1 menit

Tabel A.2 Uji Coba 2

Kondisi	Pengujian ke 2	
	Status	Waktu
Rotasi 0 Derajat dengan Jarak 10 cm	Berhasil	5.8 detik
Rotasi 45 Derajat dengan Jarak 10 cm	Berhasil	4.58 detik
Rotasi 90 Derajat dengan Jarak 10 cm	Berhasil	5.44 detik
Rotasi 180 Derajat dengan Jarak 10 cm	Berhasil	7.04 detik
Pencahayaan Terang dengan Jarak 10 cm	Berhasil	2.48 detik
Pencahayaan Remang dengan Jarak 10 cm	Berhasil	2.12 detik
Posisi dengan QR Code Sejajar dengan jarak 10 cm	Berhasil	2.28 detik
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kanan dengan Jarak 10 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kiri dengan Jarak 10 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari atas dengan Jarak 10 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari bawah dengan Jarak 10 cm	Gagal	1 menit
Rotasi 0 Derajat dengan Jarak 20 cm	Berhasil	1.89 detik
Rotasi 45 Derajat dengan Jarak 20 cm	Berhasil	2.40 detik
Rotasi 90 Derajat dengan Jarak 20 cm	Berhasil	1.41 detik
Rotasi 180 Derajat dengan Jarak 20 cm	Berhasil	2.44 detik
Pencahayaan Terang dengan Jarak 20 cm	Berhasil	1.63 detik
Pencahayaan Remang dengan Jarak 20 cm	Berhasil	1.46 detik
Posisi dengan QR Code Sejajar dengan jarak 20 cm	Berhasil	1.44 detik
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kanan dengan Jarak 20 cm	Gagal	1 menit

Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kiri dengan Jarak 20 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari atas dengan Jarak 20 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari bawah dengan Jarak 20 cm	Gagal	1 menit
Rotasi 0 Derajat dengan Jarak 30 cm	Berhasil	1.41 detik
Rotasi 45 Derajat dengan Jarak 30 cm	Berhasil	1.73 detik
Rotasi 90 Derajat dengan Jarak 30 cm	Berhasil	1.79 detik
Rotasi 180 Derajat dengan Jarak 30 cm	Berhasil	1.45 detik
Pencahayaan Terang dengan Jarak 30 cm	Berhasil	3.33 detik
Pencahayaan Remang dengan Jarak 30 cm	Berhasil	1.29 detik
Posisi dengan QR Code Sejajar dengan jarak 30 cm	Berhasil	1.44 detik
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kanan dengan Jarak 30 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kiri dengan Jarak 30 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari atas dengan Jarak 30 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari bawah dengan Jarak 30 cm	Gagal	1 menit

Tabel A.3 Uji Coba 3

Kondisi	Pengujian ke 3	
	Status	Waktu
Rotasi 0 Derajat dengan Jarak 10 cm	Berhasil	6.71 detik
Rotasi 45 Derajat dengan Jarak 10 cm	Berhasil	9.40 detik
Rotasi 90 Derajat dengan Jarak 10 cm	Berhasil	4.20 detik
Rotasi 180 Derajat dengan Jarak 10 cm	Berhasil	4.20 detik

Pencahayaan Terang dengan Jarak 10 cm	Berhasil	3.11 detik
Pencahayaan Remang dengan Jarak 10 cm	Berhasil	4.23 detik
Posisi dengan QR Code Sejajar dengan jarak 10 cm	Berhasil	3.02 detik
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kanan dengan Jarak 10 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kiri dengan Jarak 10 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari atas dengan Jarak 10 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari bawah dengan Jarak 10 cm	Gagal	1 menit
Rotasi 0 Derajat dengan Jarak 20 cm	Berhasil	1.50 detik
Rotasi 45 Derajat dengan Jarak 20 cm	Berhasil	1.90 detik
Rotasi 90 Derajat dengan Jarak 20 cm	Berhasil	3.31 detik
Rotasi 180 Derajat dengan Jarak 20 cm	Berhasil	1.58 detik
Pencahayaan Terang dengan Jarak 20 cm	Berhasil	1.68 detik
Pencahayaan Remang dengan Jarak 20 cm	Berhasil	1.37 detik
Posisi dengan QR Code Sejajar dengan jarak 20 cm	Berhasil	1.30 detik
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kanan dengan Jarak 20 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kiri dengan Jarak 20 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari atas dengan Jarak 20 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari bawah dengan Jarak 20 cm	Gagal	1 menit
Rotasi 0 Derajat dengan Jarak 30 cm	Berhasil	1.56 detik

Rotasi 45 Derajat dengan Jarak 30 cm	Berhasil	1.67 detik
Rotasi 90 Derajat dengan Jarak 30 cm	Berhasil	1.40 detik
Rotasi 180 Derajat dengan Jarak 30 cm	Berhasil	2.71 detik
Pencahayaan Terang dengan Jarak 30 cm	Berhasil	2.67 detik
Pencahayaan Remang dengan Jarak 30 cm	Berhasil	1.4 detik
Posisi dengan QR Code Sejajar dengan jarak 30 cm	Berhasil	1.31 detik
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kanan dengan Jarak 30 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kiri dengan Jarak 30 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari atas dengan Jarak 30 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari bawah dengan Jarak 30 cm	Gagal	1 menit

Tabel A.4 Uji Coba 4

Kondisi	Pengujian ke 4	
	Status	Waktu
Rotasi 0 Derajat dengan Jarak 10 cm	Berhasil	3.30 detik
Rotasi 45 Derajat dengan Jarak 10 cm	Berhasil	7.81 detik
Rotasi 90 Derajat dengan Jarak 10 cm	Berhasil	2.48 detik
Rotasi 180 Derajat dengan Jarak 10 cm	Berhasil	3.22 detik
Pencahayaan Terang dengan Jarak 10 cm	Berhasil	3.68 detik
Pencahayaan Remang dengan Jarak 10 cm	Berhasil	1.77 detik
Posisi dengan QR Code Sejajar dengan jarak 10 cm	Berhasil	2.98 detik
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kanan dengan Jarak 10 cm	Gagal	1 menit

Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kiri dengan Jarak 10 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari atas dengan Jarak 10 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari bawah dengan Jarak 10 cm	Gagal	1 menit
Rotasi 0 Derajat dengan Jarak 20 cm	Berhasil	1.61 detik
Rotasi 45 Derajat dengan Jarak 20 cm	Berhasil	2.23 detik
Rotasi 90 Derajat dengan Jarak 20 cm	Berhasil	2.06 detik
Rotasi 180 Derajat dengan Jarak 20 cm	Berhasil	1.60 detik
Pencahayaan Terang dengan Jarak 20 cm	Berhasil	2.91 detik
Pencahayaan Remang dengan Jarak 20 cm	Berhasil	3.13 detik
Posisi dengan QR Code Sejajar dengan jarak 20 cm	Berhasil	1.41 detik
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kanan dengan Jarak 20 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kiri dengan Jarak 20 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari atas dengan Jarak 20 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari bawah dengan Jarak 20 cm	Gagal	1 menit
Rotasi 0 Derajat dengan Jarak 30 cm	Berhasil	1.41 detik
Rotasi 45 Derajat dengan Jarak 30 cm	Berhasil	2.25 detik
Rotasi 90 Derajat dengan Jarak 30 cm	Berhasil	1.33 detik
Rotasi 180 Derajat dengan Jarak 30 cm	Berhasil	1.81 detik
Pencahayaan Terang dengan Jarak 30 cm	Berhasil	3.11 detik
Pencahayaan Remang dengan Jarak 30 cm	Berhasil	1.5 detik
Posisi dengan QR Code Sejajar dengan jarak 30 cm	Berhasil	1.73 detik

Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kanan dengan Jarak 30 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kiri dengan Jarak 30 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari atas dengan Jarak 30 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari bawah dengan Jarak 30 cm	Gagal	1 menit

Tabel A.5 Uji Coba 5

Kondisi	Pengujian ke 5	
	Status	Waktu
Rotasi 0 Derajat dengan Jarak 10 cm	Berhasil	9.20 detik
Rotasi 45 Derajat dengan Jarak 10 cm	Berhasil	3.30 detik
Rotasi 90 Derajat dengan Jarak 10 cm	Berhasil	5.20 detik
Rotasi 180 Derajat dengan Jarak 10 cm	Berhasil	04.14 detik
Pencahayaan Terang dengan Jarak 10 cm	Berhasil	2.58 detik
Pencahayaan Remang dengan Jarak 10 cm	Berhasil	3.41 detik
Posisi dengan QR Code Sejajar dengan jarak 10 cm	Berhasil	2.78 detik
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kanan dengan Jarak 10 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kiri dengan Jarak 10 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari atas dengan Jarak 10 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari bawah dengan Jarak 10 cm	Gagal	1 menit
Rotasi 0 Derajat dengan Jarak 20 cm	Berhasil	1.70 detik
Rotasi 45 Derajat dengan Jarak 20 cm	Berhasil	2.18 detik

Rotasi 90 Derajat dengan Jarak 20 cm	Berhasil	3.08 detik
Rotasi 180 Derajat dengan Jarak 20 cm	Berhasil	2.20 detik
Pencahayaan Terang dengan Jarak 20 cm	Berhasil	1.34 detik
Pencahayaan Remang dengan Jarak 20 cm	Berhasil	3.29 detik
Posisi dengan QR Code Sejajar dengan jarak 20 cm	Berhasil	1.11 detik
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kanan dengan Jarak 20 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kiri dengan Jarak 20 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari atas dengan Jarak 20 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari bawah dengan Jarak 20 cm	Gagal	1 menit
Rotasi 0 Derajat dengan Jarak 30 cm	Berhasil	1.50 detik
Rotasi 45 Derajat dengan Jarak 30 cm	Berhasil	1.36 detik
Rotasi 90 Derajat dengan Jarak 30 cm	Berhasil	1.48 detik
Rotasi 180 Derajat dengan Jarak 30 cm	Berhasil	1.44 detik
Pencahayaan Terang dengan Jarak 30 cm	Berhasil	2.21 detik
Pencahayaan Remang dengan Jarak 30 cm	Berhasil	1.63 detik
Posisi dengan QR Code Sejajar dengan jarak 30 cm	Berhasil	1.50 detik
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kanan dengan Jarak 30 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari sebelah kiri dengan Jarak 30 cm	Gagal	1 menit
Posisi dengan QR Code Kemiringan 30 derajat dari atas dengan Jarak 30 cm	Gagal	1 menit

Posisi dengan QR Code Kemiringan 30 derajat dari bawah dengan Jarak 30 cm	Gagal	1 menit

BIODATA PENULIS

I Nyoman Pande Wahyu Dharmawan, lahir di Bekasi pada tanggal 29 Mei 1995. Penulis merupakan anak ketiga dari tiga bersaudara pasangan Bapak I Nyoman Astawa dan Ibu Nyoman Ayu. Penulis menempuh Pendidikan formal dimulai dari TK Permata Nusantara Bekasi (1999-2001), SD Tunas Jakasampurna Bekasi (2001-2004), SMP Global Prestasi School Bekasi (2007-2010), SMA Negeri 81 Jakarta Timur (2010-2013) dan S1 Teknik Informatika ITS (2013-2017). Bidang Studi yang diambil oleh penulis pada saat berkuliah di Teknik Informatika ITS adalah Manajemen Informasi (MI). Selama masa kuliah, penulis aktif dalam organisasi kemahasiswaan yaitu staff Departemen Dalam Negeri Himpunan Mahasiswa Teknik Computer – Informatika (2014-2015), Kepala Divisi Kekeluargaan Departemen Dalam Negeri Himpunan Mahasiswa Teknik Computer – Informatika (2015-2016). Penulis juga aktif dalam berbagai kegiatan kepantiaian yaitu SCHEMATICS 2014-2016 dan ITS EXPO 2014-2016. Penulis juga sempat menjadi asisten dosen selama menjalani perkuliahan sebagai asisten dosen mata pada kuliah Analisis Perancangan Sistem Informasi, Manajemen Proyek Perangkat Lunak, dan Interaksi Manusia dan Komputer. Penulis memiliki ketertarikan dalam hal bidang *data scientist* dan menyukai hal – hal yang baru. Komunikasi dengan penulis dapat melalui email: **nyomanwahyu.d@gmail.com**