

**RANCANG BANGUN PROTOTIPE SISTEM KONFIRMASI
PENGIRIMAN BARANG MENGGUNAKAN QR CODE PADA
PERANGKAT BERGERAK**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
MUHAMMAD MAHFUD
NIM: 105060801111059



PROGRAM STUDI INFORMATIKA / ILMU KOMPUTER
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2015

PENGESAHAN

RANCANG BANGUN PROTOTIPE SISTEM KONFIRMASI PENGIRIMAN BARANG
MENGUNAKAN QR CODE PADA PERANGKAT BERGERAK

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh:
MUHAMMAD MAHFUD
NIM. 105060801111059

Skripsi ini telah diuji dan dinyatakan lulus pada
29 Oktober 2015

Telah diperiksa dan disetujui oleh:

Pembimbing I

Pembimbing II

Agi Putra Kharisma, S.T., M.T
NIK. 2013048604301001

Edy Santoso, S.Si., M.Kom
NIP. 19740414 200312 1 004

Mengetahui

Ketua Program Studi Informatika/Ilmu Komputer

Drs. Marji, M.T
NIP. 19670801 199203 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 29 Oktober 2015



Muhammad Mahfud
NIM: 105060801111059

KATA PENGANTAR

Puji dan syukur Penulis panjatkan kehadirat Allah SWT, karena hanya dengan rahmat dan bimbingannya Penulis dapat menyelesaikan tugas akhir dengan judul “Rancang Bangun Prototipe Sistem Konfirmasi Pengiriman Barang Menggunakan QR Code pada Perangkat Bergerak” dengan baik. Tanpa rahmat dan bimbingan dari Tuhan Yang Maha Esa, maka niscaya Penulis tidak akan dapat menyelesaikan tugas akhir ini dengan baik dan tepat waktu.

Terima kasih pula Penulis sampaikan kepada pihak-pihak yang telah membantu Penulis dalam penyelesaian tugas akhir ini. Pihak-pihak tersebut antara lain:

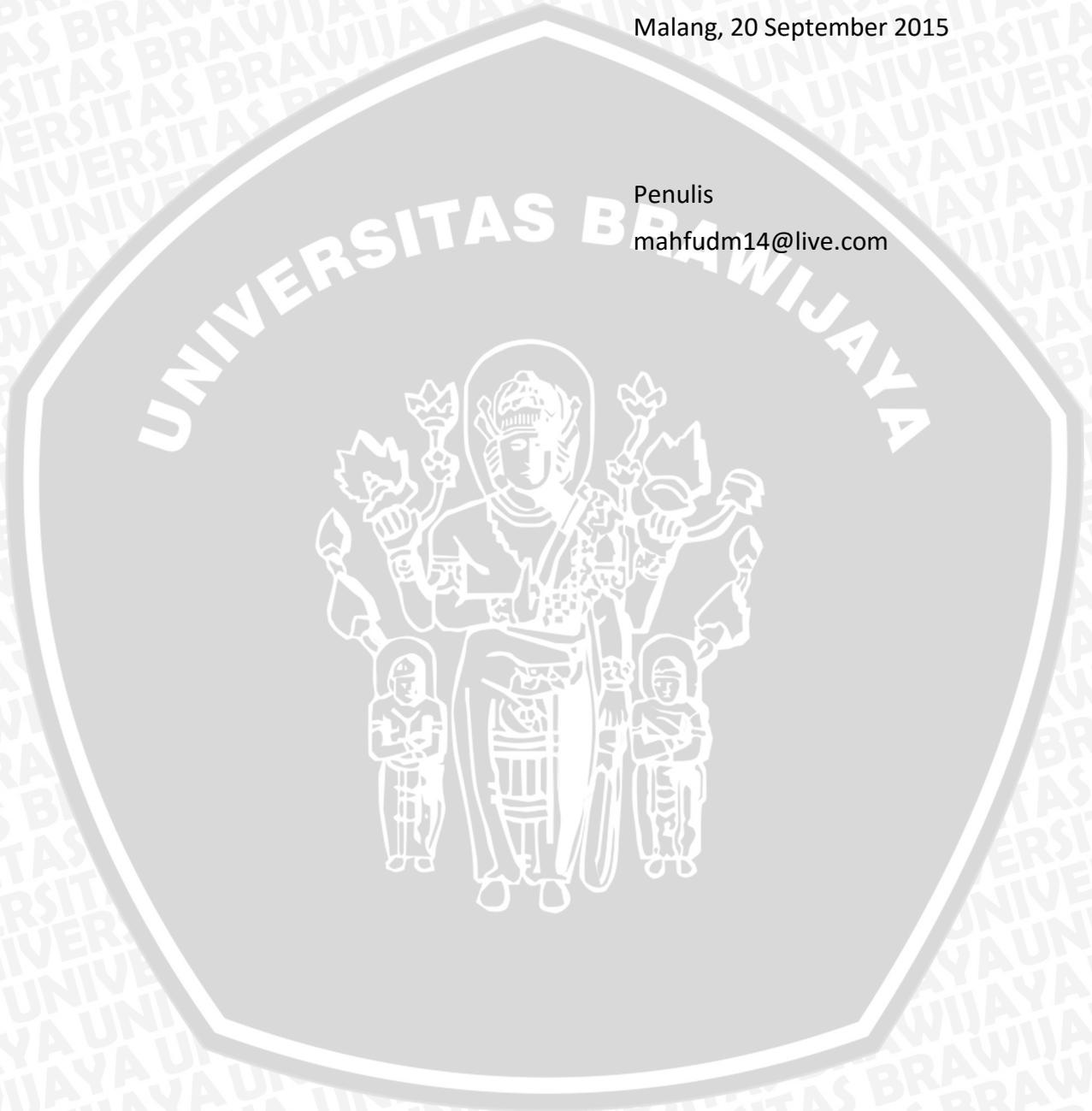
1. Agi Putra Kharisma, S.T., M.T., dan Edy Santoso, S.Si., M.Kom., selaku dosen pembimbing I dan dosen pembimbing II yang telah banyak memberikan ide, pemikiran, bimbingan, ilmu, dan saran dalam penyusunan tugas akhir ini.
2. Issa Arwani, S.T., M.T., selaku dosen pembimbing akademik yang telah memberikan bimbingan, ilmu, dan saran selama Penulis menuntut ilmu di Fakultas Ilmu Komputer, Universitas Brawijaya.
3. Drs. Mardji, M.Si., dan Issa Arwani, S.T., M.T., selaku Ketua dan Sekretaris Program Studi Informatika serta segenap Bapak / Ibu Dosen, Staff Administrasi dan Perpustakaan Fakultas Ilmu Komputer Universitas Brawijaya.
4. Orang tua Penulis, Bapak Nawari dan Ibu Amadah yang telah memberi motivasi, kasih sayang serta dukungan moril dan materil kepada Penulis. Saudari penulis Mbak Siti Romlah dan Mbak Siti Nur Halimah yang telah memberikan semangat dari awal sampai akhir pengerjaan tugas akhir ini.
5. Teman-teman Teknik Informatika angkatan 2010 yang telah senantiasa menghibur dan menemani serta memberikan dukungan, masukan, dan inspirasi kepada penulis selama menempuh studi dan menyelesaikan skripsi ini.
6. Seluruh pihak yang telah membantu kelancaran penulisan tugas akhir yang tidak dapat disebutkan satu persatu.

Semoga jasa dan amal baik mendapatkan balasan dari Allah SWT, tidak ada satu orang pun sempurna, begitu juga dengan penulis dalam menulis skripsi ini. Dengan segala kerendahan hati, penulis mengharapkan masukan dan kritik

dari semua pihak demi tercapainya hasil yang lebih baik atau kesempurnaan dari skripsi ini. Akhir kata semoga skripsi ini bermanfaat bagi semua pihak.

Malang, 20 September 2015

Penulis
mahfudm14@live.com



ABSTRAK

Muhammad Mahfud, 2015, Rancang Bangun Prototipe Sistem Konfirmasi Pengiriman Barang Menggunakan QR Code pada Perangkat Bergerak. Fakultas Ilmu Komputer, Universitas Brawijaya, Malang. Dosen Pembimbing: Agi Putra Kharisma, S.T., M.T. dan Edy Santoso, S.Si., M.Kom.

Pada sistem konfirmasi pengiriman menggunakan tanda tangan dalam jual beli *online* sering dijumpai masalah dalam pengiriman barang, salah satunya adalah tanda tangan mudah ditiru sehingga konfirmasi pengiriman bias dimanipulasi oleh orang lain. Untuk menjamin barang sampai pada pembeli sebenarnya dibutuhkan sebuah tanda pengenal dari penerima barang yang bisa dikenali kurir dan dapat digunakan untuk konfirmasi pengiriman barang, misalnya QR Code berisikan kode pengiriman barang yang diberikan oleh pihak penjual atau pengirim barang. Dan untuk mengkonfirmasi pengiriman barang dibutuhkan prototipe sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak.

Prototipe sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak mempunyai beberapa fitur di dalamnya, yaitu *login*, konfirmasi pengiriman dengan dua metode yaitu metode utama adalah *scan* QR Code dan metode alternatifnya memasukkan kode pengiriman, yang terakhir mengirim hasil konfirmasi pengiriman pada *server*. Prototipe sistem konfirmasi pengiriman barang menggunakan QR Code berupa aplikasi *hybrid* dengan menggunakan *framework Cordova* yang dijalankan pada perangkat bergerak dengan sistem operasi *Android*. Data konfirmasi pengiriman diterima dari *server* setelah kurir berhasil *login* pada aplikasi. Kemudian enkripsi MD5 digunakan untuk mengamankan kode pengiriman barang.

Dari hasil pengujian disimpulkan bahwa fungsi dari sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak telah berjalan dengan benar, dan kode pengiriman yang diterima oleh aplikasi klien telah berhasil dienkripsi.

Kata kunci: konfirmasi pengiriman, QR Code, perangkat bergerak

ABSTRACT

Muhammad Mahfud, 2015, *Design Prototype Delivery Confirmation System Using QR Code on Mobile Devices*. Faculty of Computer Science, Brawijaya University, Malang. Advisor: Agi Putra Kharisma, S.T., M.T. and Edy Santoso, S.Si., M.Kom.

Delivery confirmation system using signature for online shopping has many problem for a good delivery process. One of them is signature is easy to be imitated. In the result, the delivery confirmation can be manipulated by other person. To ensure the good is delivered to the right person need a sign that can be recognize by courier and can be used for confirm a good delivery, like a QR Code contains a delivery confirmation code provided by online shop or delivery service party. Then delivery confirmation system using QR Code on mobile devices is needed for confirm a good delivery.

The prototype of delivery confirmation system using QR Code on mobile devices has some features, they are login, confirm for a delivered item with two method, scan QR Code as primary method and insert the delivery code for alternative method, and the last feature is application send delivery confirmation result to server. Prototype of delivery confirmation system using QR Code is a hybrid application use Cordova framework and built for mobile devices with Android Operating system. Applicaton receives delivery confirmation code data from server when courier signed in successfully. Then delivery confirmation code is secured by using a MD5 encryption method.

The result of testing concludes that prototype of delivery confirmation system using QR Code on mobile devices's functions are running properly and the delivery confirmation code can be encrypted successfully.

Keyword: delivery confirmation, QR Code, mobile devices

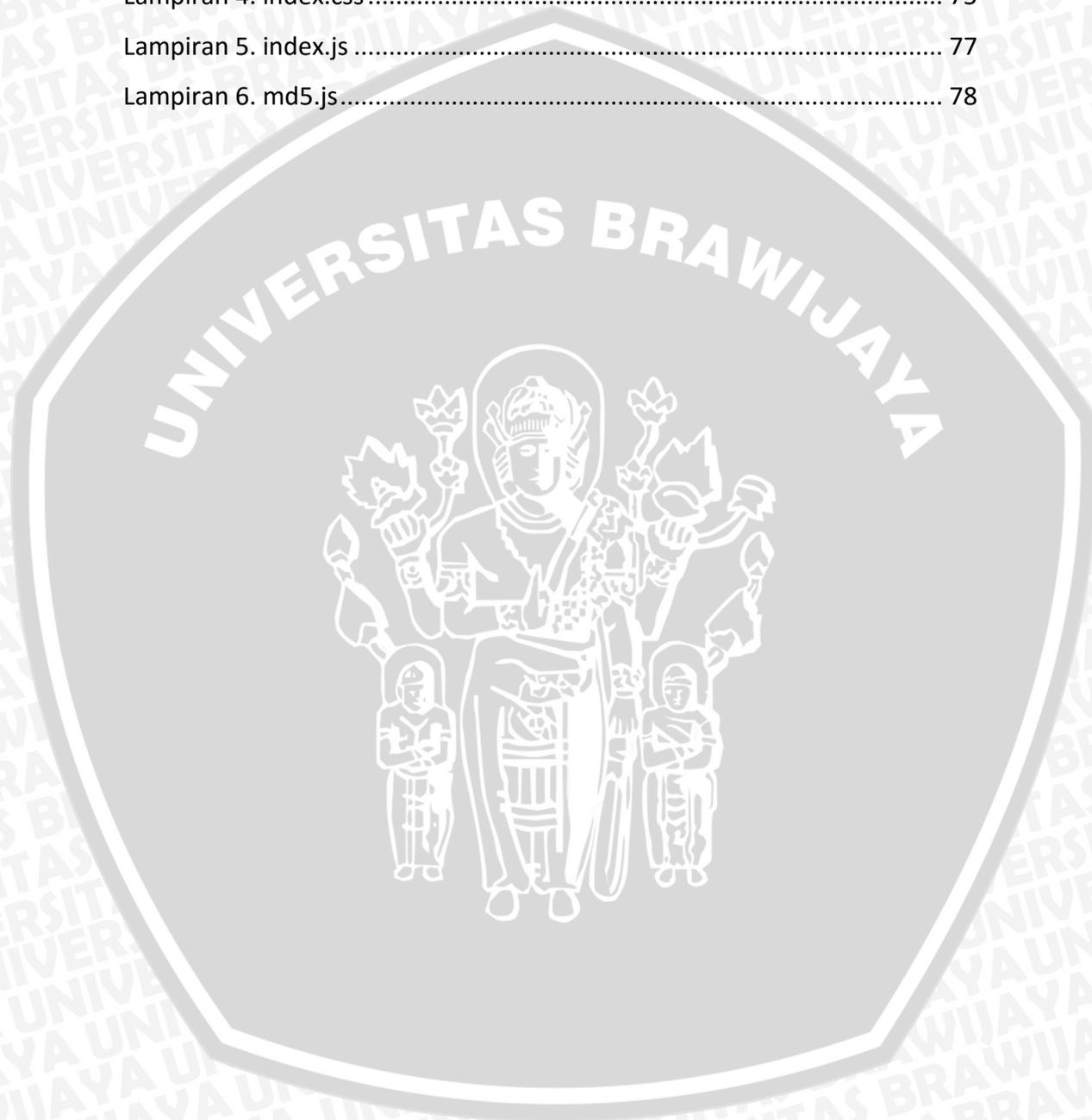
DAFTAR ISI

PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
<i>ABSTRACT</i>	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiii
DAFTAR KODE.....	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Manfaat.....	3
BAB II KAJIAN PUSTAKA.....	4
2.1 Sistem pengiriman barang.....	4
2.2 QR Code.....	4
2.3 <i>Android</i>	5
2.4 <i>Phonegap</i>	6
2.5 <i>JQuery Mobile</i>	7
2.6 Metode Pengembangan Perangkat Lunak.....	8
BAB III METODOLOGI PENELITIAN.....	11
3.1. Studi Literatur.....	11
3.2. Analisis Kebutuhan Sistem.....	12
3.3. Perancangan.....	12
3.4. Implementasi.....	12
3.5. Pengujian dan Analisis.....	13
3.6. Pengambilan Kesimpulan dan Saran.....	13
BAB IV ANALISIS DAN PERANCANGAN.....	14

4.1. Analisis Kebutuhan.....	14
4.1.1. Gambaran Umum.....	14
4.1.2. Daftar Kebutuhan.....	15
4.1.3. Skenario <i>Use Case</i>	17
4.2. Perancangan Sistem.....	20
4.2.1. Perancangan Arsitektural.....	20
4.2.2. Perancangan Aktifitas Sistem.....	24
4.2.3. Perancangan Alur Proses Dalam Sistem	27
4.2.4. Perancangan Konseptual.....	30
4.2.5. Perancangan Basis Data	31
4.2.6. Perancangan Antar Muka.....	32
BAB V IMPLEMENTASI.....	35
5.1 Spesifikasi Lingkungan Pengembangan Sistem.....	35
5.2 Batasan Implementasi.....	36
5.3 Implementasi Basis Data.....	36
5.4 Implementasi <i>Web API</i>	37
5.5 Implementasi Aplikasi Klien	39
BAB VI PENGUJIAN	49
6.1. Pengujian.....	49
6.1.1. Pengujian user login.....	50
6.1.2. Pengujian konfirmasi kode pengiriman barang	54
6.1.3. Pengujian update data pengiriman barang	58
6.1.4. Pengujian enkripsi data ketika berhasil login	63
6.1.5. Pengujian enkripsi data ketika update data pengiriman barang.....	65
6.2. Analisis Pengujian.....	67
BAB VII KESIMPULAN DAN SARAN	68
7.1. Kesimpulan.....	68
7.2. Saran.....	68
DAFTAR PUSTAKA.....	70
LAMPIRAN	72



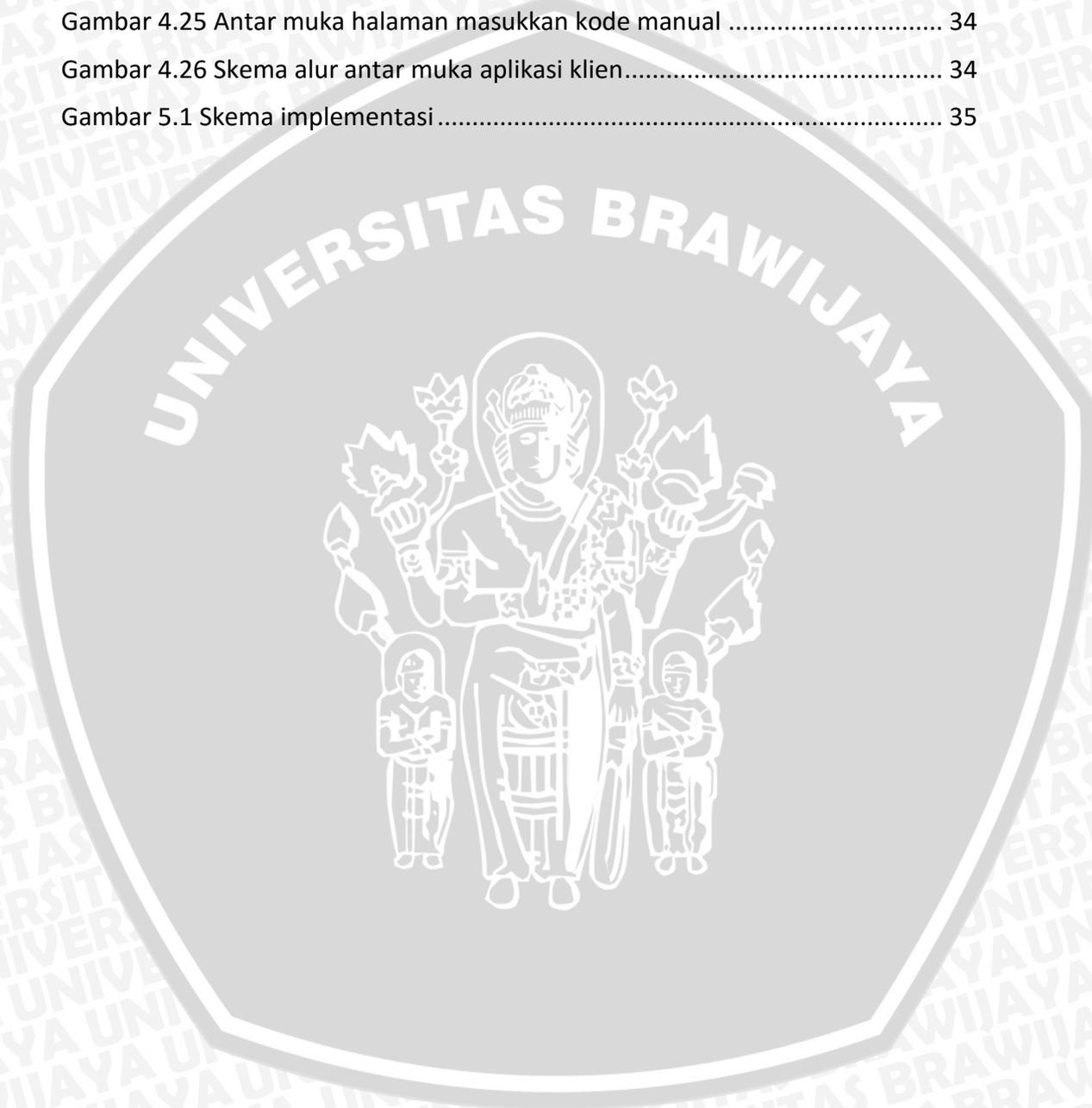
Lampiran 1. Kode DDL <i>Database Server</i>	72
Lampiran 2. connect.php	72
Lampiran 3. index.html	72
Lampiran 4. index.css	75
Lampiran 5. index.js	77
Lampiran 6. md5.js	78



DAFTAR GAMBAR

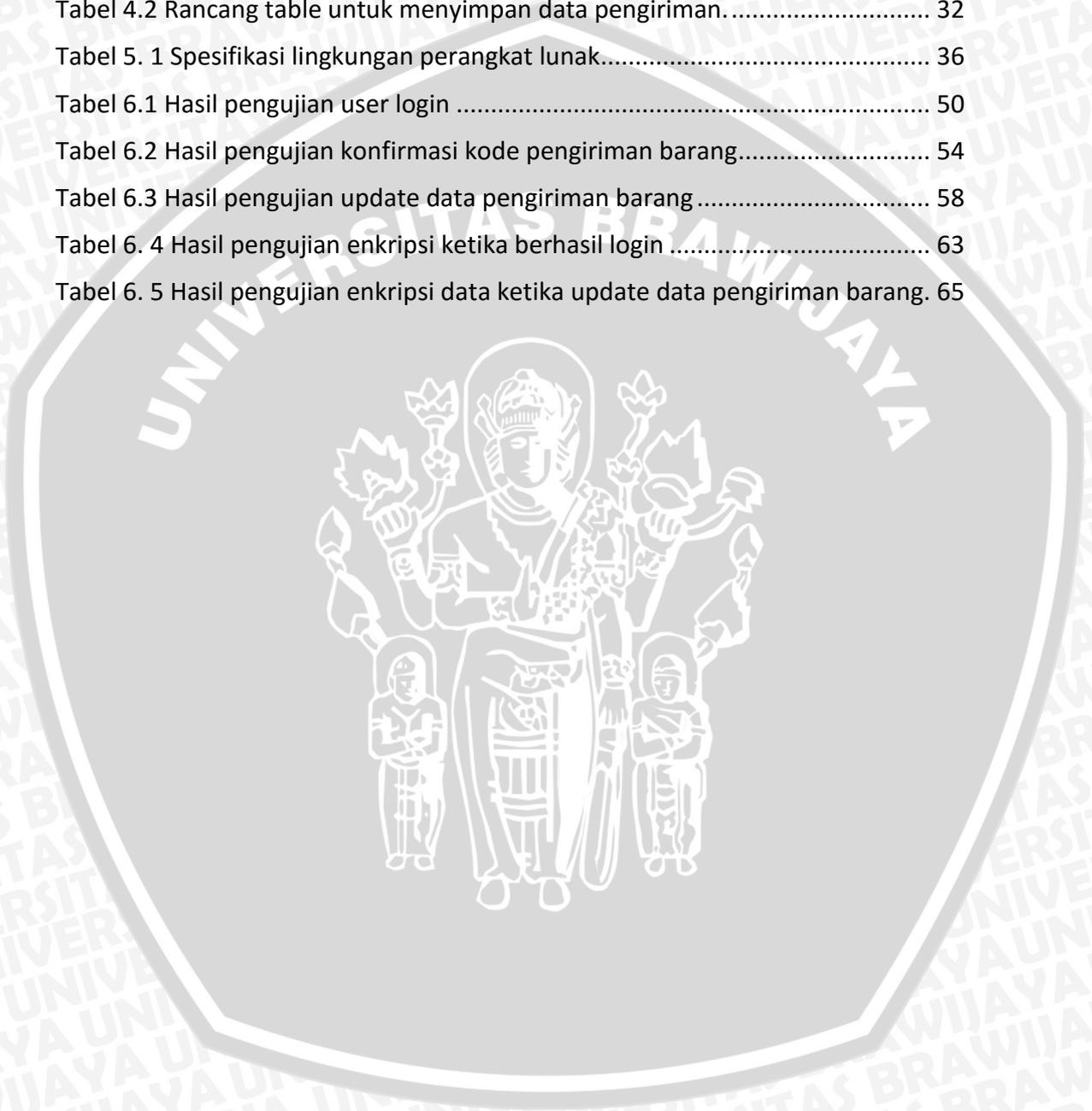
Gambar 2.1 Contoh bentuk QR Code.....	4
Gambar 2.2 Arsitektural Phonegap.....	7
Gambar 2.3 Fase-fase model Waterfall menurut Pressman	9
Gambar 2.4 Fase-fase model Waterfall menurut Sommerville	10
Gambar 3.1 Tahapan penelitian.....	11
Gambar 4.1 Proses berjalannya sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak.....	15
Gambar 4.2 Contoh QR Code yang dimiliki penerima barang.....	16
Gambar 4.3 Use case diagram sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak	16
Gambar 4.4 Arsitektural aplikasi klien pada perangkat bergerak dengan sistem operasi <i>Android</i>	21
Gambar 4.5 Skema JSON <i>form login</i>	22
Gambar 4.6 Skema JSON login berhasil.....	22
Gambar 4.7 Skema JSON <i>login gagal</i>	22
Gambar 4.8 Alur pengiriman JSON pada proses melakukan <i>login</i>	23
Gambar 4.9 Skema JSON daftar kode pengiriman.....	23
Gambar 4.10 Skema JSON menerima daftar data pengiriman baru	24
Gambar 4.11 Alur transaksi JSON pada proses mengirim hasil konfirmasi pengiriman	24
Gambar 4.12 Diagram aktifitas proses <i>login</i>	25
Gambar 4.13 Diagram aktifitas mengkonfirmasi pengiriman barang	26
Gambar 4.14 Diagram aktifitas proses mengirim hasil konfirmasi pengiriman ..	27
Gambar 4.15 <i>Sequence diagram</i> untuk alur proses <i>Login</i> berhasil	28
Gambar 4.16 <i>Sequence diagram</i> untuk alur proses <i>Login</i> gagal.....	28
Gambar 4.17 <i>Sequence diagram</i> dari proses konfirmasi pengiriman dengan cara <i>scan QR Code</i>	29
Gambar 4.18 <i>Sequence diagram</i> dari proses konfirmasi pengiriman dengan cara manual.....	29
Gambar 4.19 <i>Sequence diagram</i> Mengirim hasil konfirmasi pengiriman	30
Gambar 4.20 <i>Class diagram</i> aplikasi klien	30

Gambar 4.21 Basis data pada aplikasi klien	31
Gambar 4.22 Antar muka tampilan awal aplikasi klien	32
Gambar 4.23 Antar muka <i>form login</i>	33
Gambar 4.24 Antar muka halaman utama aplikasi klien	33
Gambar 4.25 Antar muka halaman masukkan kode manual	34
Gambar 4.26 Skema alur antar muka aplikasi klien	34
Gambar 5.1 Skema implementasi	35



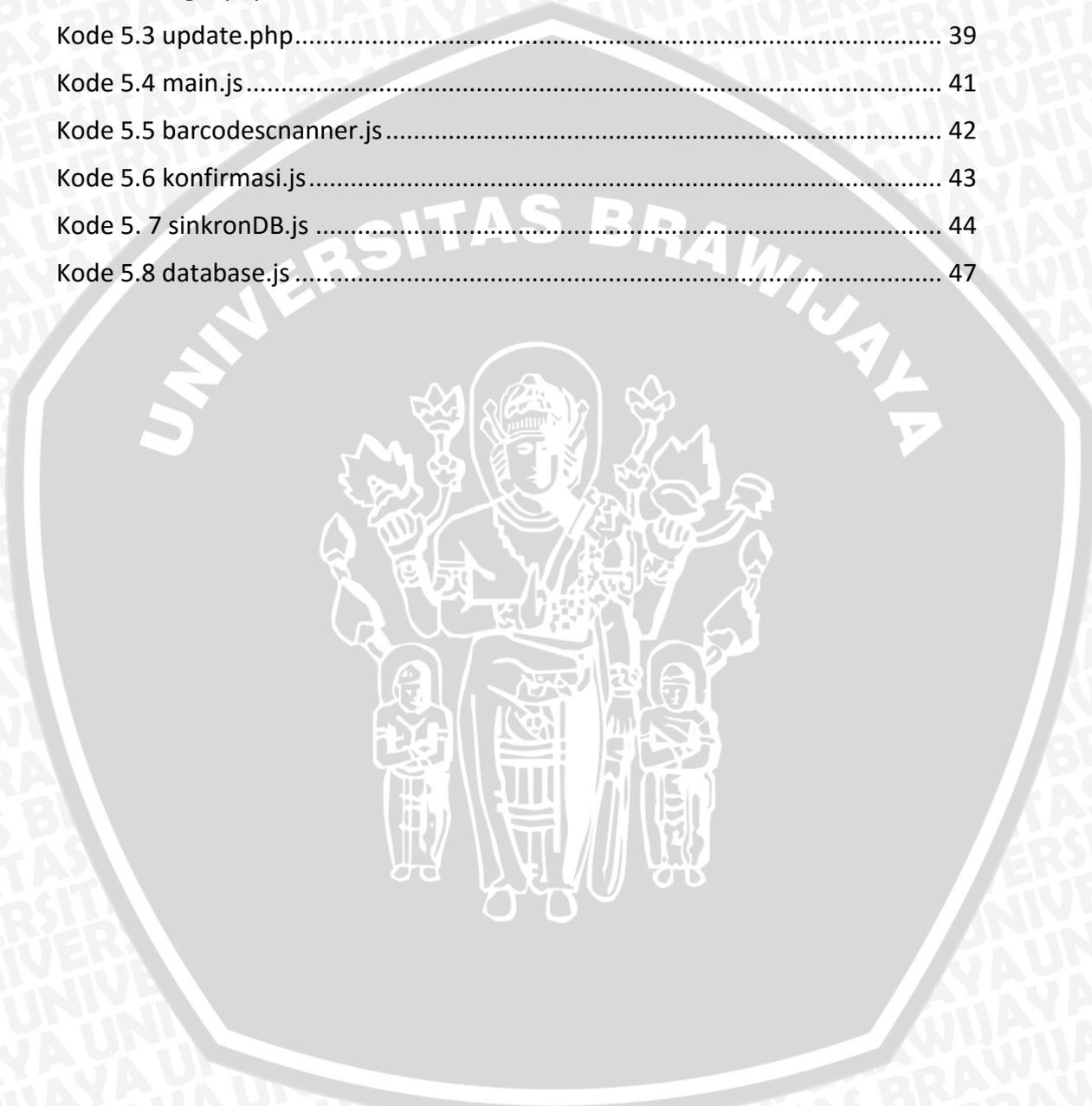
DAFTAR TABEL

Tabel 4.1 Daftar kebutuhan non fungsional	17
Tabel 4.2 Rancang table untuk menyimpan data pengiriman.....	32
Tabel 5. 1 Spesifikasi lingkungan perangkat lunak.....	36
Tabel 6.1 Hasil pengujian user login	50
Tabel 6.2 Hasil pengujian konfirmasi kode pengiriman barang.....	54
Tabel 6.3 Hasil pengujian update data pengiriman barang.....	58
Tabel 6. 4 Hasil pengujian enkripsi ketika berhasil login	63
Tabel 6. 5 Hasil pengujian enkripsi data ketika update data pengiriman barang.	65



DAFTAR KODE

Kode 5.1 DDL Basis data pada aplikasi klien	37
Kode 5.2 login.php	38
Kode 5.3 update.php.....	39
Kode 5.4 main.js.....	41
Kode 5.5 barcodescanner.js.....	42
Kode 5.6 konfirmasi.js.....	43
Kode 5.7 sinkronDB.js	44
Kode 5.8 database.js	47



BAB I PENDAHULUAN

1.1 Latar Belakang

Jual beli *online* atau *e-commerce* sudah bukan hal yang asing lagi di dalam kehidupan masyarakat Indonesia. Dimulai dari berdagang di media sosial sampai dengan bisnis yang lebih serius yaitu dengan membuat aplikasi *online shop* sendiri. Teknologi ini sangat menguntungkan baik bagi pembeli dan penjual. Selain menghemat waktu dan tenaga bagi pembeli, teknologi ini juga terbukti meningkatkan pendapatan perusahaan, karena pembeli hanya perlu mengaksesnya dari telepon genggam atau *mobile device* yang selalu pembeli bawa, biasa disebut dengan *m-commerce* [ASK-13]. Bukan hanya itu kebiasaan jual beli *online* ini sendiri semakin lama semakin banyak peminatnya seiring meningkatnya pengguna *internet* di Indonesia. Hal ini disebabkan membaiknya pertumbuhan perekonomian di Indonesia. Dari data lembaga riset *International Data Corporation (IDC)*, nilai perdagangan lewat internet di Indonesia tahun 2011 mencapai 3,4 miliar dollar AS atau sekitar Rp 30 triliun. Hasil *survei Master Card Worldwide* pada Februari 2012 juga menunjukkan tren peningkatan belanja *online* sebesar 15 persen di Indonesia dalam enam bulan ke depan. Meningkatnya transaksi *e-commerce* di Indonesia antara lain disebabkan membaiknya pertumbuhan perekonomian, di samping tumbuhnya kelas menengah. Bank Dunia menyebutkan bahwa 56,5 persen populasi Indonesia atau sekitar 134 juta jiwa masuk kategori kelas menengah dengan nilai belanja 2-20 dollar AS per hari [TKP-14].

Adapun hal yang cukup berpengaruh dalam jual beli *online* adalah kualitas keamanan pengiriman barang. Semakin baik tanggung jawab penjual terhadap pengiriman barang semakin meningkatkan kepercayaan pelanggan pada penjual *online* tersebut. Pada pengiriman barang yang ada saat ini digunakan sistem konfirmasi pengiriman barang menggunakan tanda tangan. Terdapat beberapa kesalahan-kesalahan dalam pengiriman barang yang terjadi pada sistem konfirmasi menggunakan tanda tangan sehingga membuat pembeli kecewa salah satunya adalah disaat barang tiba di alamat tujuan namun pembeli tidak sedang berada di lokasi. Tindakan dari pengirim barang adalah menyerahkan barang kiriman kepada tetangga terdekat dari pembeli. Namun belum tentu tetangga tersebut akan menyerahkannya kepada penerima yang sebenarnya. Hal ini yang belum bisa diantisipasi oleh sistem pengiriman barang saat ini yang dimana sampainya barang dikonfirmasi menggunakan tanda tangan penerima.

Untuk mendukung terjaminnya pengiriman barang sampai pada tujuan dan pada tangan yang tepat dibutuhkan sebuah sistem yang dapat mendukung konfirmasi yang benar-benar dari pembelinya atau pembeli dapat mewakilkannya pada orang lain ketika tidak sedang berada di lokasi pengiriman barang. Dalam hal ini konfirmasi pengiriman barang yang hanya dimiliki oleh pembeli dan tidak dapat direkayasa oleh orang lain seperti menggunakan *QR Code* berisikan kode

pengiriman yang dikirim oleh pihak penjual atau pengirim barang kepada pembeli dan hanya dimiliki oleh pembeli saja. Dan QR Code juga sangatlah mudah untuk di-scan (pindai) oleh kamera smartphone atau sejenisnya. Disamping itu perkembangan teknologi informasi pada perangkat bergerak seperti *smartphone*, *tablet*, *notebook*, dan sebagainya. Terutama pada perangkat dengan sistem operasi *Android*. Sistem operasi *Android* yang berbasis *open source* sangat mendominasi perkembangan teknologi perangkat bergerak, membuat perangkat *mobile* seperti *smartphone* dan *tablet* sangat banyak digunakan [KPS-14]. Selain itu QR Code juga tidak dapat di manipulasi oleh sembarang orang. Dengan mengkonfirmasi penerima barang dengan informasi yang ada pada QR Code maka konfirmasi pengiriman barang lebih valid dan juga lebih cepat karena hasil scan langsung dikirim ke *server*.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan di atas, dihasilkan rumusan masalah pada penelitian ini adalah bagaimana membangun prototipe sistem konfirmasi pengiriman barang yang dapat digunakan untuk menjamin barang kiriman sampai pada tangan yang tepat dengan memanfaatkan QR Code sebagai alat untuk mengkonfirmasi pengiriman barang yang dipindai menggunakan kamera dari perangkat bergerak.

1.3 Batasan Masalah

Berdasarkan latar belakang dan rumusan masalah yang telah dikemukakan, penelitian ini mempunyai batasan-batasan masalah sebagai berikut:

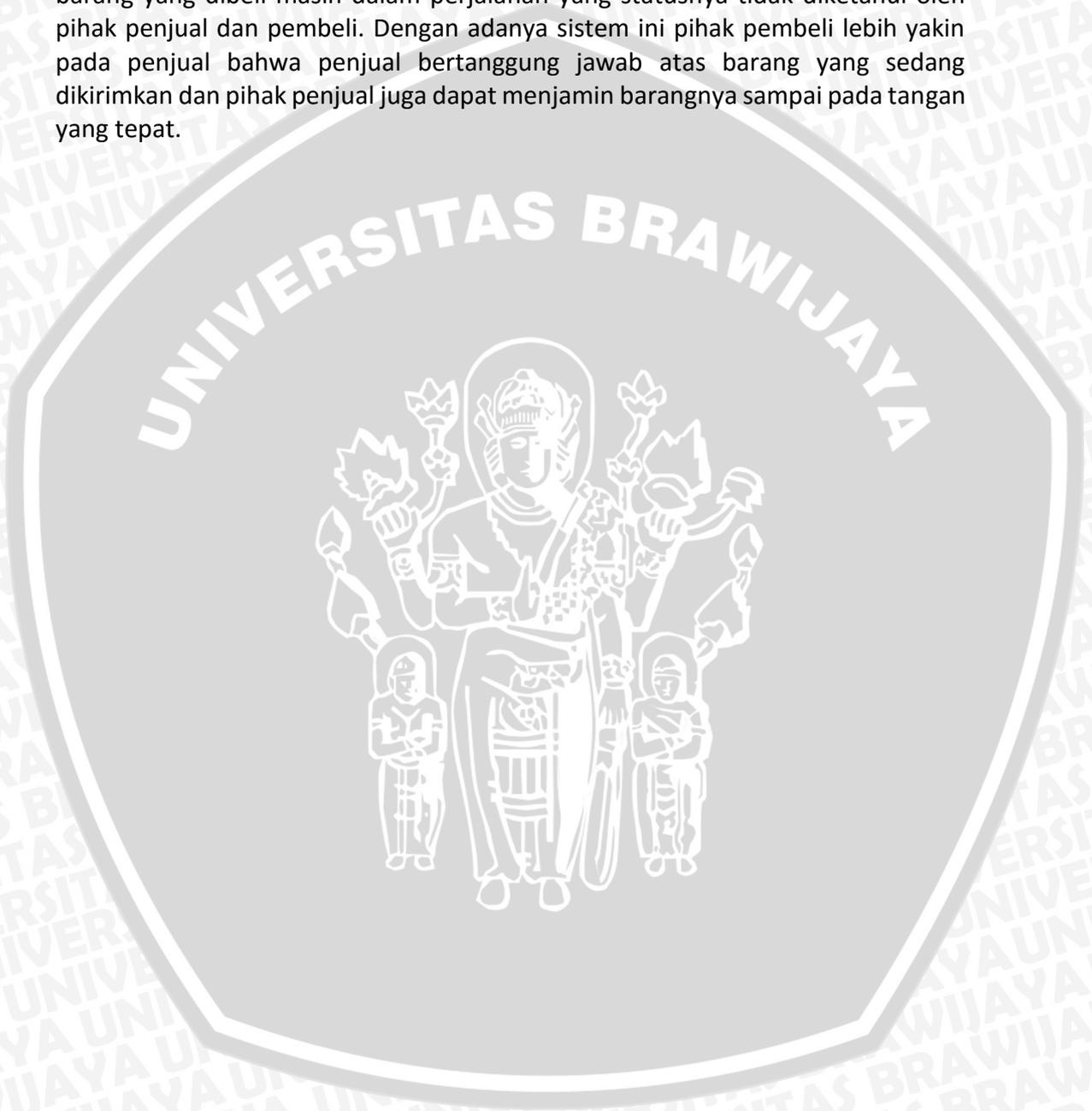
1. Sistem pengiriman barang yang akan digantikan adalah sistem pengiriman barang yang menggunakan tanda tangan yang dilakukan oleh penerima barang.
2. Platform perangkat bergerak yang digunakan untuk aplikasi ini adalah Android.
3. Pengirim barang berasal dari pihak ketiga yang bekerja sama dengan penjual barang (Online Shop)

1.4 Tujuan

Tujuan dari penelitian ini adalah untuk membuat sebuah sistem konfirmasi pengiriman barang pada *online shop* yang lebih dapat diakui validasinya dengan memastikan barang sampai pada tangan yang tepat menggunakan *scan QR Code*, dimana yang sebelumnya menggunakan tanda tangan dari penerima barang yang dimana pengirim barang tidak bisa mengenali tanda tangan pembeli barang sehingga bisa dengan mudah direkayasa oleh sembarang orang.

1.5 Manfaat

Manfaat yang diharapkan dari sistem ini adalah dapat meningkatkan keamanan dan kenyamanan dalam jual beli *online* dengan menjamin barang sampai pada tangan orang yang tepat. Terutama dari pihak pembeli yang dimana saat setelah membeli barang dan melakukan transaksi pembayaran pembelian barang yang dibeli masih dalam perjalanan yang statusnya tidak diketahui oleh pihak penjual dan pembeli. Dengan adanya sistem ini pihak pembeli lebih yakin pada penjual bahwa penjual bertanggung jawab atas barang yang sedang dikirimkan dan pihak penjual juga dapat menjamin barangnya sampai pada tangan yang tepat.



BAB II

KAJIAN PUSTAKA

Bab ini berisi pembahasan tentang teori dasar yang berhubungan dengan pengembangan perangkat lunak yang dilakukan. Teori dasar yang akan dibahas pada bab ini yaitu Sistem pengiriman barang, *Android*, *Phonegap*, *Jquery Mobile*, *QR Code*, *Web Service* dan Metode pengembangan perangkat lunak.

2.1 Sistem pengiriman barang

Sistem adalah kumpulan dari elemen-elemen yang saling berkaitan dan tersusun untuk mencapai suatu tujuan tertentu. Secara umum Pengiriman Barang adalah segala upaya yang diselenggarakan secara sendiri atau bersama-sama dalam suatu organisasi untuk memberikan pelayanan jasa berupa pengiriman barang [AMS-06]. Sistem pengiriman barang adalah sebuah sistem yang digunakan oleh seseorang atau organisasi untuk memudahkan upaya pelayanan pengiriman barang.

Layanan pengiriman barang sudah banyak kita temui seiring dengan berkembangnya teknologi informasi. Beberapa contoh jasa pengiriman barang yang ada di Indonesia diantaranya JNE, POS Indonesia, TIKI, dan lain-lain.

2.2 QR Code

QR Code merupakan singkatan dari *Quick Response Code*, atau dapat diterjemahkan menjadi kode respon cepat. *QR Code* dikembangkan oleh Denso Corporation, sebuah perusahaan Jepang yang banyak bergerak di bidang otomotif. *QR Code* ini dipublikasikan pada tahun 1994 dengan tujuan untuk pelacakan kendaraan di bagian manufaktur dengan cepat dan mendapatkan respon dengan cepat pula.

QR Code adalah *image* berupa matriks dua dimensi yang memiliki kemampuan untuk menyimpan data di dalamnya. *QR Code* merupakan evolusi dari kode batang (*barCode*). *BarCode* merupakan sebuah simbol penandaan objek nyata yang terbuat dari pola batang-batang berwarna hitam dan putih agar mudah untuk dikenali oleh komputer [RYD-06]. Contoh sebuah *QR Code* pada gambar 2.2:



Gambar 2.1 Contoh bentuk *QR Code*

Sumber: [RYD-06]

2.3 **Android**

Android merupakan sistem operasi *open source* yang dikembangkan oleh *Google* untuk perangkat *mobile*. Aplikasi *Android* dapat dikembangkan melalui *Android SDK (Android Standart Development Kit)* dengan menggunakan bahasa *java* dan *Google* menyediakan banyak *library* yang dapat membantu *developer* dalam mengembangkan program pada *Android*. Perangkat *Android* sendiri terdiri dari Sistem Operasi, *middleware*, dan *key application*. Pengembangan *software Android* menggunakan bahasa pemrograman *java* dan aplikasi *Android* tidak berjalan langsung diatas *kernel* sistem operasi namun aplikasi tersebut berjalan diatas *Dalvik, virtual machine* yang khusus dioptimalkan untuk perangkat *mobile*.

Adapun arsitektur *Android* terdiri dari 5 Layer, yaitu [HSP-12]:

1. *Layer Application*. *Layer Application*, layer dimana aplikasi tersebut berjalan seperti *email client*, kalender, peta, dan lain-lain.
2. *Layer Framework*, layer yang berisikan *framework API* yang digunakan oleh *developer* dalam membangun aplikasi untuk *platform Android*.
3. *Layer libraries*, layer dimana *library* yang digunakan oleh berbagai komponen dalam sistem *Android* tersimpan.
4. *Layer Runtime*, layer yang berisi berbagai *core libraries* yang menyediakan sebagian besar fungsionalitas yang serupa dengan *API* pada pemrograman *Java*.

Pada OS *Android* terdapat tiga sensor dasar, yaitu:

1. Sensor gerak, merupakan sensor yang mengukur setiap pergerakan perpindahan maupun perputaran yang bertumpu pada tiga sumbu (x, y, z). Beberapa yang termasuk sensor *accelerometer*, sensor gravitasi, sensor perputaran vektor.
2. Sensor *environmental*, yaitu sensor yang mendeteksi tentang hal-hal menyangkut kondisi disekitar seperti suhu udara, penerangan, dan kelembaban. Termasuk dalam kategori ini adalah termometer, *barometer*, dan *photometer*.
3. Sensor posisi, yaitu sensor yang berfungsi mengukur posisi secara fisik dari perangkat sendiri. Termasuk dalam sensor ini adalah sensor orientasi dan *magnetometer*.

Terdapat beberapa *hardware* yang didukung oleh *Android* yang dapat digunakan pada sebagian besar perangkat dengan OS *Android*. Diantaranya adalah [AND-15]:

1. Prosesor yang mendukung adalah *ARM Architecture (ARMv7)* atau setelahnya bahkan *android 5.0* mendukung *ARMv8-A* baik 32bit maupun 64bit. Selain itu sejak tahun 2012 *Android* juga mendukung *Intel* untuk digunakan sebagai prosesor dari perangkat yang digunakan.

2. RAM (*Random Access Memory*) yang mendukung berbagai jenis kapasitas. Namun pada tahun 2013 mulai dari *Android* versi 4.4 (Kitkat) RAM minimal yang digunakan adalah 512 MB.
3. GPU (*Graphic Processing Unit*) yang digunakan pada perangkat-perangkat *Android* adalah OpenGL ES 1.1, 2.0, 3.0, dan 3.1.
4. Kamera dengan berbagai jenis kemampuan pengambilan gambar dan juga fitur yang tersedia pada perangkat *android*
5. Layar sentuh dengan berbagai ukuran dan resolusi dan dapat me-render tampilan 2D dan 3D.
6. Berbagai jenis perangkat sensor tambahan yang dapat meningkatkan performa dan fungsi dari perangkat *android*.

Beberapa keunggulan *Android* adalah:

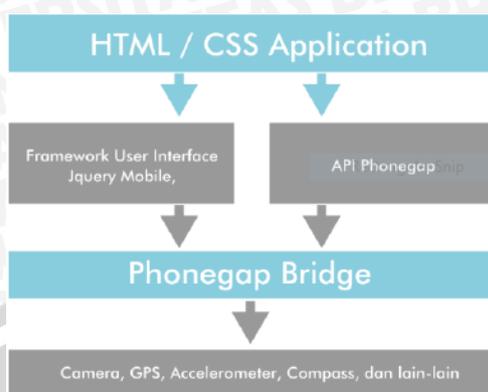
1. *Open Source*, sehingga siapa saja bebas untuk melakukan pengembangan aplikasi menggunakan *platform Android* tanpa harus membayar *lisence* untuk *developer*.
2. *Android* dikembangkan oleh *Google* yang merupakan salah satu perusahaan paling besar, paling kaya, dan paling inovatif di dunia teknologi saat ini, sehingga mempunyai banyak sponsor yang akan mendukung para pengembang aplikasi *Android*.
3. Aplikasi *Android* sangat mudah untuk dipasarkan dari pada aplikasi pada *platform* lain. Hal ini yang paling penting bagi para *developer* aplikasi *mobile* baik pemula maupun profesional.

2.4 *Phonegap*

Phonegap atau *Cordova* adalah kerangka pengembangan *mobile* diproduksi oleh Nitobi, dibeli oleh *Adobe Systems*. Dengan *framework* ini memungkinkan *programmer* perangkat lunak untuk membangun aplikasi untuk perangkat *mobile* menggunakan *JavaScript*, HTML5 dan CSS3, bukan perangkat bahasa spesifik seperti *Objective-C* atau biasa disebut aplikasi berbasis *hybrid*. Aplikasi yang dihasilkan *hybrid*, yang berarti bahwa tidak benar-benar asli (karena semua *render* tata letak dilakukan melalui pandangan *web* bukan kerangka asli UI *platform*) atau murni berbasis *web* (karena bukan aplikasi hanya *web*, tetapi dikemas sebagai aplikasi untuk distribusi dan memiliki akses ke API perangkat asli). *Cordova* berfungsi untuk menghubungkan aplikasi berbasis *web* dengan sistem operasi dalam perangkat bergerak agar bisa menggunakan layanan yang ada dalamnya, seperti sensor, *contact*, kamera, dan lain-lain.

Perangkat lunak yang mendasari *Phonegap* adalah *Apache Cordova*. Perangkat lunak ini sebelumnya hanya disebut "*Phonegap*", kemudian "*Apache Callback*", kemudian *Apache Cordova* adalah perangkat lunak *open source* [HNS-13].

Arsitektural dari *Phonegap* seperti pada gambar 2.2:



Gambar 2.2 Arsitektural *Phonegap*

Pada gambar 2.2 ditunjukkan terdapat layer *Phonegap Bridge* yang berfungsi untuk menghubungkan aplikasi dengan fungsi dari layanan-layanan pada perangkat bergerak.

Untuk menggunakan layanan *phonegap* bisa dengan cara CLI (Command Line Interface) atau dengan cara memasang *library* dari *phonegap* ke dalam *project*. *Phonegap* telah menyediakan berbagai *plugin* yang dapat digunakan dengan mudah. Untuk cara pemasangan *library* bisa dengan mencari *plugin* yang terdapat pada *web* Github dan mengunduhnya. Setelah *plugin* diunduh, *plugin* dimasukkan pada *folder plugin* yang terdapat dalam *project*. Sedangkan jika menggunakan CLI hanya dengan memasukkan perintah `cordova plugin add [nama plugin / lokasi direktori]`.

Pada penelitian ini penggunaan *phonegap* menggunakan CLI sehingga pengerjaan aplikasi klien tidak menggunakan *text editor* yang disertai fungsi *debug project*.

2.5 *JQuery Mobile*

JQuery Mobile adalah sebuah sistem *user interface* yang berdasarkan pada HTML. Di desain untuk membuat *website* dan *apps* yang bisa diakses pada semua *smartphone*, *tablet*, dan pengikat *desktop* [JQM-15]. *JQuery mobile* adalah sebuah *framework* untuk membangun desain *user interface* yang dioptimalkan pada perangkat bergerak (*mobile*). *Jquery mobile* dibangun berdasar *library* yang sangat banyak digunakan, yaitu *library JavaScript*, *JQuery*, dan *CSS*.

JQuery mobile sangat mudah untuk digunakan dan juga mempunyai arsitektur pemetaan yang sangat baik dalam membangun *mobile web* yang memfokuskan pada HTML5 [GGH-11]. Penggunaan *JQuery Mobile* hanya dengan memanggil *class* yang telah diidentifikasi oleh seperti *ui-header*, *ui-content*, *ui-body*, dan lain-lain. Selain itu juga bisa menggunakan api dari *JQuery Mobile* seperti *data-role*, *data-theme*, *data-icon*, dan lain-lain [JQM-15].

2.6 Metode Pengembangan Perangkat Lunak

Metodologi pengembangan perangkat lunak (atau disebut juga model proses atau paradigma rekayasa perangkat lunak) adalah suatu strategi pengembangan yang memadukan proses, metode, dan perangkat (*tools*). Metode-metode rekayasa perangkat lunak, memberikan teknik untuk membangun perangkat lunak. Berkaitan dengan serangkaian tugas yang luas yang menyangkut analisis kebutuhan, konstruksi program, desain, pengujian, dan pemeliharaan [PRS-02].

Beberapa metode-metode pengembangan perangkat lunak (model proses pengembangan perangkat lunak) yaitu [SPM-08]:

1. *Linear Sequential Model/ Waterfall Model*

Model ini adalah model klasik yang bersifat sistematis, berurutan dalam membangun software. Dalam pengerjaan menggunakan metode *waterfall* ini proses-proses yang sebelumnya harus sudah lengkap sebelum melanjutkan pada proses berikutnya, karena pada proses ini sangat sulit untuk mengakomodasi perubahan setelah proses berjalan.

2. *Evolutionary Software Process Models*

Model ini bersifat iterative atau mengandung perulangan. Hasil proses berupa produk yang makin lama makin lengkap sampai versi terlengkap dihasilkan sebagai produk akhir dari proses. Pada *Evolutionary* terdapat dua model, yaitu *Incremental* dan *Spiral*.

3. *RAD (Rapid Application Development)*

RAD adalah model proses pembangunan perangkat lunak yang *incremental*. RAD menekankan pada siklus pembangunan yang pendek/singkat. RAD mengadopsi model *waterfall* dan pembangunan dalam waktu singkat dicapai dengan menerapkan *component based construction*. Waktu yang singkat adalah batasan yang penting untuk model ini. Namun kelemahan dari model ini adalah tidak bisa diterapkan pada pembangunan *software* dengan skala besar.

4. *Prototyping Model*

Model pembangunan *software* ini sangat cocok untuk pembuatan *software* yang dimana *client* hanya memberi kebutuhan *software* secara umum, sehingga *developer* membuat *software* dengan spesifikasi prototipe dan kebanyakan tidak memandang efisiensi algoritma dan adaptasi sistem operasi dari *software*.

5. *Component-based Development Model*

Component-based development sangat berkaitan dengan teknologi berorientasi objek. Pada pemrograman berorientasi objek, banyak *class* yang dibangun dan menjadi komponen dalam suatu *software*. *Class-class*

tersebut bersifat *reusable*, artinya bisa digunakan kembali. Model ini bersifat iteratif atau berulang-ulang prosesnya.

6. *Extreme Programming (XP) Model*

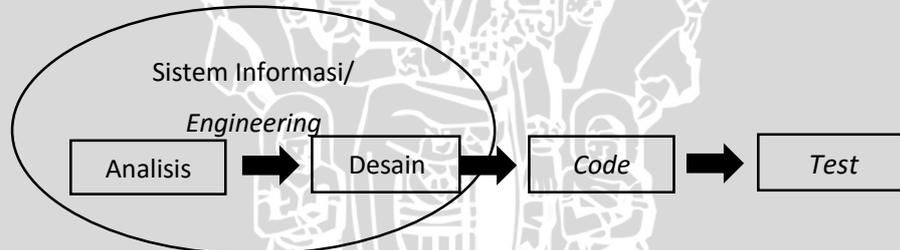
Model proses ini diciptakan dan dikembangkan oleh Kent Beck. Model ini adalah model proses yang terbaru dalam dunia rekayasa perangkat lunak dan mencoba menjawab kesulitan dalam pengembangan software yang rumit dan sulit dalam implementasi.

Menurut Kent Beck XP adalah: "*A lightweight, efficient, low-risk, flexible, predictable, scientific and fun way to develop software*". Suatu model yang menekankan pada:

- keterlibatan user secara langsung
- pengujian
- pay-as-you-go design*

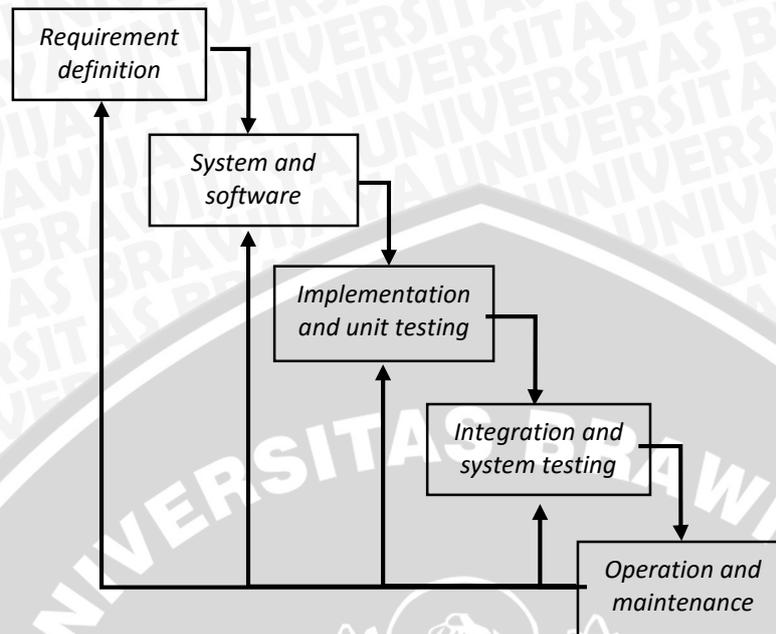
Pada penelitian ini metode pengembangan perangkat lunak yang digunakan untuk membangun sistem konfirmasi pengiriman barang dengan QR Code pada aplikasi bergerak adalah metode *Linear Sequential Model / Waterfall Model*. Pada model pengembangan yang dikenal dengan sebutan metode *Waterfall* ini terdapat beberapa fase sebagai berikut [SPM-08]:

1. Menurut Pressman:



Gambar 2.3 Fase-fase model *Waterfall* menurut *Pressman*

2. Menurut Sommerville:



Gambar 2.4 Fase-fase model *Waterfall* menurut *Sommerville*

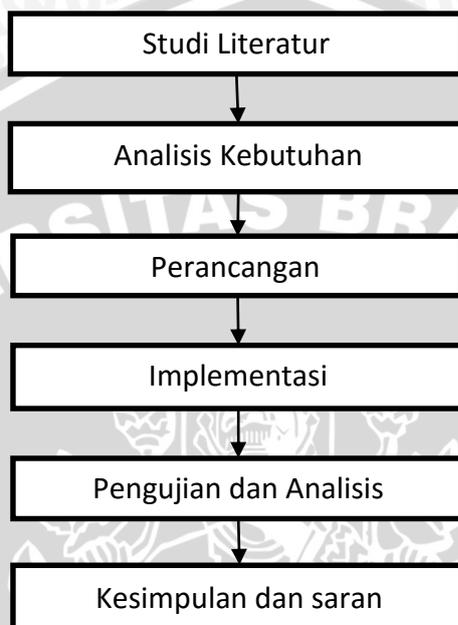
Penjelasan proses dari fase-fase pada model *waterfall*:

1. *Requirements analysis and definition*: Mengumpulkan kebutuhan secara lengkap kemudian dianalisis dan didefinisikan kebutuhan yang harus dipenuhi oleh program yang akan dibangun. Fase ini harus dikerjakan secara lengkap untuk bisa menghasilkan desain yang lengkap.
2. *System and software design*: Desain dikerjakan setelah kebutuhan selesai dikumpulkan secara lengkap.
3. *Implementation and unit testing*: desain program diterjemahkan ke dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah ditentukan. Program yang dibangun langsung diuji secara unit.
4. *Integration and system testing*: Penyatuan unit-unit program kemudian diuji secara keseluruhan (*system testing*).
5. *Operation and maintenance*: mengoperasikan program dilingkungannya dan melakukan pemeliharaan, seperti penyesuaian atau perubahan karena adaptasi dengan situasi sebenarnya.

BAB III

METODOLOGI PENELITIAN

Metode penelitian yang digunakan dalam pembuatan Rancang Bangun Prototipe Sistem Konfirmasi Pengiriman Barang Menggunakan QR Code Pada Perangkat Bergerak adalah sebagai berikut:



Gambar 3.1 Tahapan penelitian

Gambar 3.1 menunjukkan tahapan penelitian secara umum. Langkah pertama yang dilakukan yaitu mengumpulkan teori-teori yang mendukung dan mengemasnya ke dalam studi literature yang bisa dijadikan bahan pembuatan sistem konfirmasi pengiriman barang menggunakan QR Code pada aplikasi *mobile*. Tahap selanjutnya adalah proses analisis kebutuhan dan perancangan sistem. Kemudian dilanjutkan dengan melakukan implementasi ke dalam kode-kode bahasa pemrograman yang telah ditentukan sesuai dengan perancangan. Kemudian dilakukan pengujian dan analisis pada aplikasi yang telah dibuat dan evaluasi hasil uji coba. Kesimpulan dan saran disertakan sebagai catatan atas aplikasi dan kemungkinan arah pengembangan aplikasi selanjutnya.

3.1. Studi Literatur

Tahap ini adalah proses pengumpulan bahan, informasi, keterangan dan teori baik dari buku maupun dari jurnal terkait dengan pembuatan sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak. Adapun teori-teori pendukung tersebut meliputi:

1. Sistem pengiriman barang
2. *Android*

3. *Phonegap*
4. *Jquery Mobile*
5. *QR Code*
6. Metode pengemabangan Sistem.

3.2. Analisis Kebutuhan Sistem

Analisis kebutuhan bertujuan untuk mendapatkan semua kebutuhan yang diperlukan dari sistem yang akan dibangun. Analisis diperlukan untuk dapat membangun aplikasi yang sesuai kebutuhan dan terstruktur. Proses ini dilakukan dengan mendapatkan semua kebutuhan pembuatan aplikasi agar dapat menentukan sejauh mana aplikasi dapat mencapai tujuan dari pembuatan aplikasi.

Dalam pembuatan sebuah sistem atau perangkat lunak awalnya akan dijelaskan cara kerja dan fungsi-fungsi yang ada pada system yang akan dibuat dalam gambaran umum sistem. Dari gambaran umum akan didapatkan beberapa kebutuhan dari sistem yang akan dibuat digambarkan dalam *use case diagram*. Penggunaan *use case* sangat penting dalam mendeskripsikan kebutuhan-kebutuhan dan fungsionalitas perangkat lunak dari sudut pandang *user*. Analisis kebutuhan dilakukan dengan mengidentifikasi semua kebutuhan sistem konfirmasi pengiriman barang dimulai dari barang diterima oleh kurir, proses *user* (kurir) *login*, hingga konfirmasi pengiriman barang kepada server yang kemudian akan dimodelkan dalam bentuk diagram *use case*. Dari tiap-tiap *use case* yang ada akan dijelaskan lebih rinci dalam skenario *use case*.

3.3. Perancangan

Perancangan perangkat lunak dilakukan setelah semua kebutuhan sistem atau perangkat lunak telah didapatkan. Perancangan sistem atau perangkat lunak pada penelitian ini menggunakan bahasa pemodelan UML (*Unified Modeling Language*) yang dimulai dari perancangan arsitektural, perancangan diagram aktivitas yang digunakan untuk merancang seluruh alur aktifitas yang dilakukan *user* (kurir), perancangan *sequence diagram* yang menjelaskan hubungan antar *object*, juga perancangan basis data. Kemudian tahap perancangan dilanjutkan dengan perancangan *user interface*.

3.4. Implementasi

Pada tahap ini seluruh analisis kebutuhan dan perancangan sistem akan diimplementasikan ke dalam kode-kode bahasa pemrograman yang telah ditentukan. Tahap awal dari proses implementasi adalah melakukan penjabaran spesifikasi lingkungan penggunaan sistem atau perangkat lunak. Implementasi perangkat lunak pada penelitian ini dilakukan dengan menggunakan bahasa pemrograman PHP pada *server* dan bahasa pemrograman *Web* (HTML, CSS, *Jquery*, *Java Script*) dan *Java* pada aplikasi klien (kurir). Implementasi *user interface* berdasarkan perancangan yang telah dilakukan. Selanjutnya dilakukan

implementasi dan pemasangan pada *smartphone* android sebagai klien, dan pada *web API (host)* untuk *server*.

3.5. Pengujian dan Analisis

Pengujian sistem atau perangkat lunak dilakukan agar dapat menunjukkan bahwa sistem atau perangkat lunak telah mampu bekerja sesuai dengan spesifikasi dari kebutuhan yang diinginkan pada latar belakang pembuatan sistem atau perangkat lunak. Pengujian yang dilakukan meliputi pengujian validasi (*validation testing*), dan pengujian keamanan. Pengujian validasi (*validation testing*) meliputi pengujian apakah fungsi-fungsi pada aplikasi telah terpenuhi sesuai kebutuhan, pengujian keamana yakni untuk mengetahui apakah data kode pengiriman barang telah berada dalam kondisi aman baik dalam pengiriman maupun penyimpanan. Setelah tahap pengujian, dilakukan analisis untuk mengetahui hasil dari pengujian sistem atau perangkat lunak sehingga didapatkan kesimpulan dari aplikasi yang telah dibuat.

3.6. Pengambilan Kesimpulan dan Saran

Pembuatan kesimpulan dilakukan setelah semua perancangan, implementasi dan pengujian sistem aplikasi telah selesai dilakukan. Kesimpulan diambil berdasarkan dari hasil implementasi, pengujian dan analisis Rancang Bangun Prototipe Sistem Konfirmasi Pengiriman Barang Menggunakan QR Code Pada Perangkat Bergerak. Penulisan saran dimaksudkan untuk memberikan pertimbangan atas pengembangan aplikasi selanjutnya.

BAB IV ANALISIS DAN PERANCANGAN

4.1. Analisis Kebutuhan

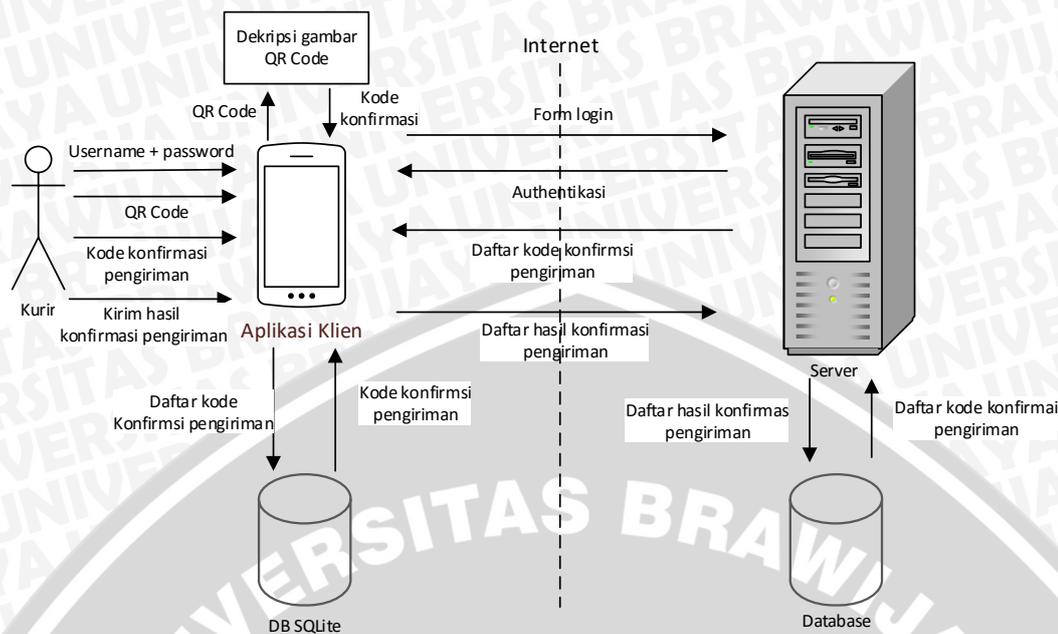
Analisis kebutuhan bertujuan untuk mendeskripsikan kebutuhan-kebutuhan yang ada pada system yang akan dibuat. Tahap-tahap pada pembuatan system konfirmasi pengiriman barang menggunakan QR Code ini adalah pertama akan dijelaskan cara kerja dari sistem dan fungsi-fungsi yang ada di dalamnya dengan gambaran umum sistem. Dari penjelasan gambaran umum sistem akan didapatkan kebutuhan fungsional dari sistem. Selanjutnya kebutuhan fungsional dan cara kerja sistem akan dibuat dalam bentuk *use case diagram* dan skenario *use case*. Dari skenario *use case* ini kebutuhan untuk memangun sistem akan didapatkan.

4.1.1. Gambaran Umum

Sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak ini adalah sebuah sistem konfirmasi pengiriman barang yang dibuat dengan tujuan untuk memastikan barang sampai pada tangan yang tepat dengan kata lain pembeli barang dari *online shop* yang sebenarnya. *User* atau pengguna perangkat lunak klien dari sistem ini adalah kurir atau pengirim barang. Dengan pihak pengirim barang adalah pihak ketiga yang dimana telah bekerja sama dengan pihak *online shop*.

Adapun sistem yang akan dikembangkan pada penelitian adalah aplikasi klien yang digunakan untuk mengkonfirmasi pengiriman dengan QR Code dan *web API* sebagai pengelola *transfer* data pada *server*. Sebelum penggunaan aplikasi klien, pada penelitian ini diasumsikan data pengiriman telah didapatkan dari pihak *online shop* dan QR Code dengan kode pengiriman telah dikirimkan pada *email* pembeli barang. Sehingga penelitian ini fokus pada proses konfirmasi pengiriman barang pada aplikasi klien.

Untuk bisa menggunakan fungsi dari aplikasi klien *user* atau kurir harus *login* terlebih dahulu menggunakan *username* dan *password* milik masing-masing kurir. Setelah berhasil *login*, kurir bisa mengkonfirmasi kode pengiriman yang telah diterima oleh pembeli barang. Metode dari konfirmasi pengiriman barang mempunyai dua metode, yaitu metode utama yang disarankan adalah *scan* gambar QR code dan metode alternatif, yaitu manual, yakni dengan cara memasukkan kode pengirimannya pada *form* kotak teks (*textbox*) kode pengiriman. Pada gambar 4.1 dijelaskan proses jalannya sistem yang terjadi.



Gambar 4.1 Proses berjalannya sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak

Dari jalanya proses yang ditunjukkan pada gambar 4.1 dikarenakan data kode pengiriman yang diproses pada penelitian ini bersifat rahasia yang hanya boleh diketahui oleh sang penerima barang, maka terdapat potensi-potensi resiko kebocoran data pada pengiriman daftar kode pengiriman dari server ke aplikasi klien, dan pada penyimpanan data dalam database di aplikasi klien. Sehingga dengan demikian pada faktor keamanan data menjadi kebutuhan non fungsional yang harus dipenuhi.

4.1.2. Daftar Kebutuhan

Berdasarkan penjabaran dari gambaran umum sistem bisa ditentukan kebutuhan dari sistem yang harus dipenuhi. Pada sistem konfirmasi pengiriman barang menggunakan QR Code yang akan dikembangkan ini terdapat dua jenis kebutuhan, yaitu kebutuhan fungsional dan kebutuhan non fungsional. Kebutuhan fungsional adalah kebutuhan yang berhubungan dengan fungsi-fungsi dalam sistem atau aplikasi klien yang akan direpresentasikan dalam use case. Sedangkan kebutuhan non fungsional merupakan kebutuhan yang diluar fungsi dari sistem, contohnya seperti keamanan kode pengiriman barang. Untuk setiap kebutuhan fungsional sistem akan dilakukan pengujian terlebih dahulu sebelum dipublikasikan. Pada penelitian ini yang menjadi kebutuhan fungsional adalah mengkonfirmasi kode pengiriman barang dengan menyesuaikan kode pengiriman yang dimasukkan oleh kurir dengan kode pengiriman pada database. Sedangkan kebutuhan non fungsionalnya adalah sistem dapat mengamankan kode pengiriman barang sehingga tidak dapat diketahui oleh pihak lain selain penerima barang.

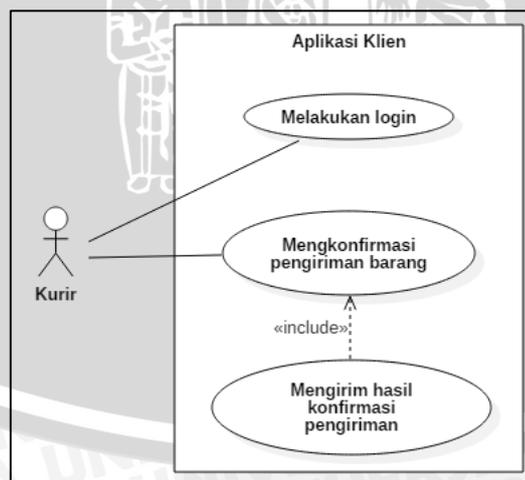
Sedangkan kode pengiriman yang diketahui oleh penerima barang adalah berupa QR Code yang dibutuhkan QR Code Reader untuk membaca isi dari QR

Code. Akan tetapi untuk mengantisipasi adanya kegagalan pembacaan QR Code maka akan disisipkan kode pengiriman yang berupa 7 alphanumeric diluar gambar QR Code, yakni berada dibagian bawah dari gambar QR Code. Sehingga contoh bentuk QR Code yang diterima oleh penerima barang seperti pada gambar 4.2.



Gambar 4.2 Contoh QR Code yang dimiliki penerima barang

Dari daftar kebutuhan sistem yang telah diperoleh, kebutuhan fungsional sistem akan direpresentasikan dalam bentuk *use case diagram*. *Use case diagram* adalah suatu susunan diagram yang menjelaskan penggunaan fungsionalitas dari sistem sehingga pengguna paham akan penggunaan dari sistem. Terdapat dua peran yang sering digunakan dalam *use case diagram*, yaitu *use case* dan aktor. *Use case* merupakan fungsi yang dapat digunakan atau dilakukan dalam sistem. Sedangkan aktor merupakan segala sesuatu diluar sistem yang digunakan, bisa berupa bagian sistem lain yang diluar sistem yang digunakan. Pada gambar 4.3 dijelaskan *use case diagram* dari kebutuhan sistem konfirmasi pengiriman barang menggunakan aplikasi klien dengan QR Code pada perangkat bergerak.



Gambar 4.3 Use case diagram sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak

Use case diagram pada gambar 4.3 menjelaskan tentang kebutuhan fungsional dari sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak. Aktor dari *use case diagram* adalah kurir dan sistemnya

adalah aplikasi klien. *Use case diagram* ini menjelaskan tentang kebutuhan dari kurir untuk mengkonfirmasi kode pengiriman. *Login* dibutuhkan sebagai syarat dari kurir untuk bisa mengkonfirmasi kode pengiriman. Setelah berhasil *login* kurir bisa mengkonfirmasi kode pengiriman dari penerima barang. Setelah kurir mengkonfirmasi kode pengiriman barang kurir bisa mengirim hasil konfirmasi pengiriman barang pada *server* untuk memberikan laporan pengiriman karena aplikasi klien hanya mengkonfirmasi kode pengiriman pada *database* yang ada pada aplikasi klien.

Selain kebutuhan fungsional yang telah dijelaskan dan direpresentasikan dalam *use case* dan skenario *use case* terdapat kebutuhan non fungsional yang tidak dimasukkan dalam *use case*. Kebutuhan ini didapatkan dari kebutuhan-kebutuhan diluar fungsi sistem. Kebutuhan non fungsional dari sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak dalam penelitian ini adalah keamanan data agar tidak bisa diketahui oleh orang lain selain penerima barang. Kebutuhan non fungsional ditunjukkan pada tabel 4.1.

Tabel 4.1 Daftar kebutuhan non fungsional

No	Nama Kebutuhan	Penjelasan	Tujuan
1	Keamanan data konfirmasi pengiriman	Data pengiriman terutama kode pengiriman sebisa mungkin tidak dapat diketahui oleh orang lain selain penerima barang, termasuk pengirim barang. Karena kode pengiriman merupakan satu-satunya tanda untuk mengetahui penerima barang yang sebenarnya. Sehingga jika orang lain mempunyai kodenya barang bisa diambil orang tersebut.	Kode pengiriman barang tidak boleh diketahui oleh orang lain selain penerima barang.

4.1.3. Skenario *Use Case*

Daftar kebutuhan telah direpresentasikan menggunakan *use case diagram* pada gambar 4.2. Dari *use case diagram* dibutuhkan skenario *use case* untuk menjelaskan tentang masing-masing *use case* yang ada pada *use case diagram*. Di dalam skenario *use case*, akan diberikan penjelasan tentang aktor, tujuan dari *use case*, deskripsi global tentang *use case*, dan alur proses dari *use case*.

1) Skenario *Use Case* Melakukan login

Sesuai *use case diagram* yang ditunjukkan pada gambar 4.2. Skenario *use case* ini menjelaskan tentang kebutuhan fungsional sistem yang harus dipenuhi yaitu *user* dapat melakukan *login* pada aplikasi dengan menggunakan ID dari *user*. Aktor dari kebutuhan ini adalah kurir sebagai *user* dari aplikasi klien.

Use case ini mempunyai tujuan kurir melakukan *login* dengan memasukkan *username* dan *password* kurir pada aplikasi klien. Melakukan *login* merupakan kebutuhan dari kurir agar bisa mengkonfirmasi kode pengiriman barang. Sehingga skenario *use case* ini menjelaskan proses dari melakukan *login*.

Alur utama skenario:

1. Kurir memasukkan *username* dan *password* pada halaman *login* dari aplikasi klien.
2. Aplikasi klien mengirim *form login* pada *server*.
3. Aplikasi klien berhasil mendapatkan autentikasi *login*.

Alur Alternatif:

- *Exception*
 - 2.a. Aplikasi klien gagal mengirim *form login* karena tidak terhubung dengan *internet*.
 - 2.a.1. Sistem menampilkan pesan koneksi *error*
 - 2.a.2. Kurir menghubungkan aplikasi klien dengan *internet* dan mencoba lagi
- *Exception*
 - 3.a. Aplikasi gagal mendapatkan autentikasi *login*.
 - 3.a.1. Sistem menampilkan pesan gagal *login*.
 - 3.a.2. Kurir memasukkan *username* dan *password* yang benar.

2) Skenario *Use Case* Mengkonfirmasi pengiriman barang

Kebutuhan kurir yang utama adalah mengkonfirmasi kode pengiriman. Kebutuhan ini bertujuan untuk dapat mengkonfirmasi kode pengiriman yang ditunjukkan penerima barang dengan menggunakan aplikasi klien.

Use case ini menjelaskan bahwa kurir melakukan konfirmasi kode pengiriman barang menggunakan aplikasi klien. Skenario *use case* ini menjelaskan dari proses konfirmasi pengiriman barang menggunakan aplikasi klien dengan kurir sebagai aktor.

Alur utama dari skenario *use case*:

1. Kurir memindai QR Code.
2. Sistem mengkonfirmasi kode pengiriman.

Alur alternative skenario:

- Varian:

- 1.a. Kurir memasukkan kode pengiriman
- *Exception*
 - 1.a. Kurir gagal memindai QR Code.
 - 1.a.1. Sistem tidak memberi respon
 - 1.a.2. Lanjut ke varian 1.a.
- *Exception*
 - 1.b. Kurir memindai QR Code yang sudah terkonfirmasi
 - 1.b.1. Sistem menampilkan pesan kode pengiriman telah terkonfirmasi
 - 1.b.2. Kurir memindai QR Code dari kode pengiriman barang yang lain.
- *Exception*
 - 2.a. Konfirmasi gagal
 - 2.a.1. Sistem menampilkan pesan konfirmasi gagal
 - 2.a.2. Kurir memasukkan kode pengiriman yang valid.

3) Skenario *Use Case* Mengirim hasil konfirmasi pengiriman barang.

Skenario *use case* ini menjelaskan tentang kebutuhan fungsional sistem yang harus dipenuhi yaitu aplikasi klien mengirim daftar hasil konfirmasi pengiriman barang pada *server*. Pada *use case* ini kurir bertindak sebagai aktor dan aplikasi klien sebagai sistem.

Tujuan dari pemenuhan kebutuhan ini adalah untuk bisa memperbarui data yang ada pada *server* sesuai dengan hasil konfirmasi pengiriman barang yang telah dilakukan kurir. Mengirim hasil konfirmasi pengiriman bisa dilakukan setelah kurir berhasil mengkonfirmasi pengiriman barang dan ketika terhubung dengan *web API* melalui *internet*. Pada skenario *use case* ini dijelaskan proses dari mengirim daftar hasil konfirmasi pengiriman kepada *server*.

Alur utama:

1. Kurir menekan tombol *Update* data konfirmasi pengiriman.
2. Aplikasi klien mengirim daftar hasil konfirmasi pengiriman barang pada *server*.
3. Aplikasi klien memperbarui daftar konfirmasi pengiriman pada *database* aplikasi klien sesuai dengan data pengiriman yang baru dari *server*.

Alur alternatif:

- *Exception*:
 - 2.a. Aplikasi klien tidak dapat terhubung dengan *server*.
 - 2.a.1. Sistem menampilkan koneksi *error*.
 - 2.a.2. Kurir menghubungkan aplikasi klien dengan *internet* dan menekan tombol *Update* data konfirmasi pengiriman lagi.

4.2. Perancangan Sistem

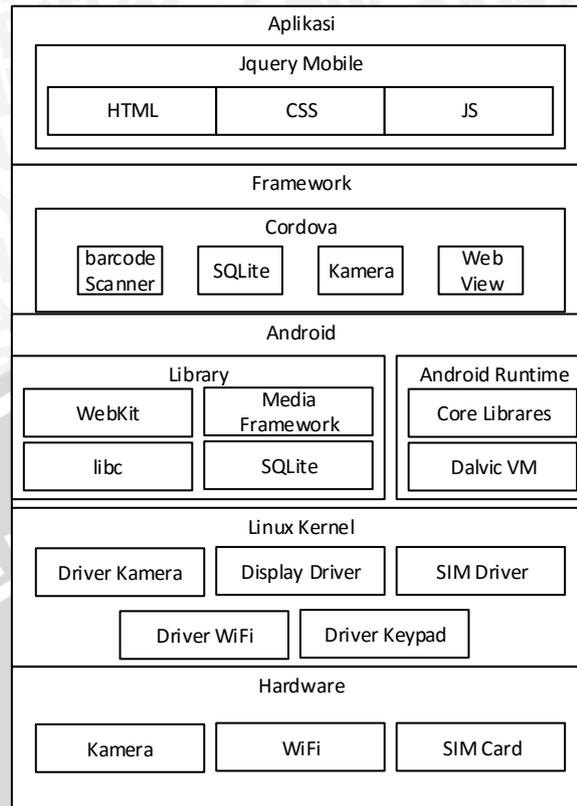
Perancangan sistem pada penelitian ini mempunyai enam tahap perancangan, yaitu perancangan arsitektural, perancangan aktifitas untuk menjelaskan aktifitas yang terjadi dalam sistem, perancangan alur proses sistem, perancangan konseptual, perancangan basis data, dan terakhir perancangan antar muka.

4.2.1. Perancangan Arsitektural

Perancangan arsitektural menjelaskan tentang arsitektural dari sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak yang terdiri dari aplikasi klien dan *web API*. Pada sub-bab ini akan dijelaskan perancangan arsitektural dari kedua elemen tersebut.

Aplikasi klien dari sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak berjalan diatas perangkat bergerak seperti *smartphone* atau *tablet* dengan sistem operasi *Android*. Selain itu aplikasi klien dibangun berbasis *hybrid*, yang berarti aplikasi yang berbasis *web* dibangun di atas *framework* untuk bisa menggunakan perangkat keras dan layanan dari perangkat yang digunakan dengan demikian aplikasi klien dari sistem bisa menggunakan kamera dari perangkat bergerak. *Framework* yang digunakan pada pengembangan aplikasi klien pada penelitian ini adalah *Phonegap* atau *Cordova*. *Cordova* adalah *framework* untuk pembangunan aplikasi perangkat berbasis *hybrid* yang sangat banyak digunakan oleh para pengembang-pengembang aplikasi perangkat bergerak. *Cordova* bertugas menghubungkan aplikasi berbasis *web* pada *Android* yang kemudian mengakses perangkat keras dari *device*. Gambar 4.4 menunjukkan susunan *layer* perangkat bergerak dari *layer* aplikasi hingga *layer hardware*.





Gambar 4.4 Arsitektural aplikasi klien pada perangkat bergerak dengan sistem operasi *Android*

Pada gambar 4.4 HTML, CSS, dan JS adalah *layer* aplikasi, yang berarti aplikasi terbuat dari HTML, CSS, dan Javascript. Hal ini karena aplikasi berupa aplikasi berbasis *web*. Dan dibawah *layer* aplikasi adalah *Cordova*, dan di bawahnya lagi adalah *Android*. Ini menunjukkan bahwa *Cordova* merupakan *framework* penghubung antara aplikasi berbasis *web* dengan *platform Android* yang kemudian *Android* menghubungkan *hardware* dari perangkat keras. Pada *Cordova* terdapat *plugin* yang dibutuhkan diantaranya *plugin BarcodeScanner*, *SQLite*, *Kamera*, dan *Web View*. Dengan menggunakan *plugin Cordova* aplikasi klien bisa menggunakan *hardware* dari perangkat bergerak dan mengambil informasi darinya seperti kamera yang digunakan untuk memindai QR Code pada penelitian ini.

Sedangkan untuk penerima data atau *web API* pada penelitian ini berjalan pada *server* berbasis Apache yang dibuat dengan PHP dan MySQL sebagai penyimpanan basis data. *Web API* memiliki fungsi sama seperti *web service*, hanya saja *web API* tidak terstandarisasi. Interaksi antara aplikasi klien dengan *web API* menggunakan protokol HTTP. Setelah itu data diproses pada *web API* dengan menggunakan bahasa program PHP.

Pada sistem ini interaksi pengiriman data memanfaatkan JSON dalam melakukan pertukaran data. *JavaScript Object Notation* atau JSON adalah standar berbasis teks untuk pertukaran data. Format JSON dikenal ringan (berukuran kecil), mudah dibaca, ditulis, dan dipahami manusia serta mudah untuk diuraikan



dan dibuat oleh mesin. Format ini dibuat berdasarkan bahasa pemrograman *JavaScript*.

Untuk data yang dikirim menggunakan JSON dijelaskan pada skema JSON:

a. Skema JSON *form login*

Pada saat melakukan *login* aplikasi klien mengirim *username* dan *password*. *Username* yang bisa dimasukkan oleh kurir adalah maksimal 20 karakter dan *password* 16 karakter dengan tipe *alphanumeric*. Sehingga bentuk skema JSON seperti pada gambar 4.5.

```
Skema JSON
username, password
```

Gambar 4.5 Skema JSON *form login*

Contoh data: {kurir1, qn3ld8dje}.

b. Skema JSON *login berhasil*

Pada saat *login berhasil server* mengirim data JSON dengan konten status autentikasi bernilai “*true*”, *id_pengiriman* dengan panjang maksimal 11 karakter dengan tipe *Integer* dan daftar kode pengiriman dengan jumlah maksimal 11 karakter *alphanumeric*. Sehingga bentuk skema JSON seperti pada gambar 4.6.

```
Skema JSON
True, {id_pengiriman, kode_konfirmasi}*
```

Gambar 4.6 Skema JSON *login berhasil*.

Keterangan: {} = data kompleks.

* = data majemuk.

Contoh data: true, [{2, jh39dn}, {3, me0d2n5}]

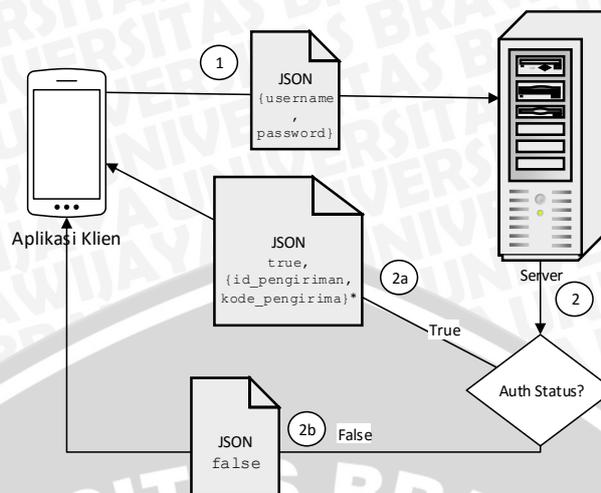
c. Skema JSON *login gagal*.

Pada saat *login gagal server* akan mengirim data JSON dengan konten status yang bernilai “*false*”. Skema JSON-nya ditunjukkan pada gambar 4.7.

```
Skema JSON
false
```

Gambar 4.7 Skema JSON *login gagal*

Alur pengiriman JSON dari proses ini ditunjukkan pada gambar 4.8.

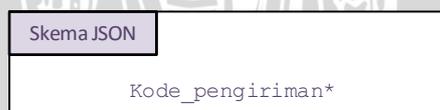


Gambar 4.8 Alur pengiriman JSON pada proses melakukan *login*.

Dari gambar 4.8 alur pengiriman JSON dapat dijelaskan seperti berikut:

- ①. Aplikasi klien mengirim *form login* berisikan *username*, dan *password* dari kurir ke *server*.
- ②. *Server* mengirim hasil autentikasi berupa JSON berisikan:
 - a. Jika berhasil status *true*, ID pengiriman, dan kode pengiriman.
 - b. Jika gagal status *false*.
- d. Skema JSON mengirim hasil konfirmasi

Pada saat aplikasi klien mengirim hasil konfirmasi data JSON yang dikirim berisikan daftar kode pengiriman yang sudah mempunyai status “Ter kirim” dengan tipe *alphanumeric*. Sehingga dengan demikian skema dari JSON untuk mengirim hasil konfirmasi bisa digambarkan seperti gambar 4.9.



Gambar 4.9 Skema JSON daftar kode pengiriman

Keterangan: * = data majemuk

Contoh data: jkjn4m2, kd92j43

- e. Skema JSON menerima daftar data pengiriman baru

Pada saat *server* menerima hasil konfirmasi dari aplikasi klien *server* mengirimkan daftar kode pengiriman barang yang baru dengan JSON yang isinya *id_pengiriman* dengan tipe *integer* panjang maksimal 11 digit, dan *kode_konfirmasi* dengan tipe *alphanumeric* dengan panjang karakter maksimal 11. Sehingga untuk skema JSON-nya seperti pada gambar 4.10.

```
Skema JSON
{id_pengiriman, kode_pengiriman}*

```

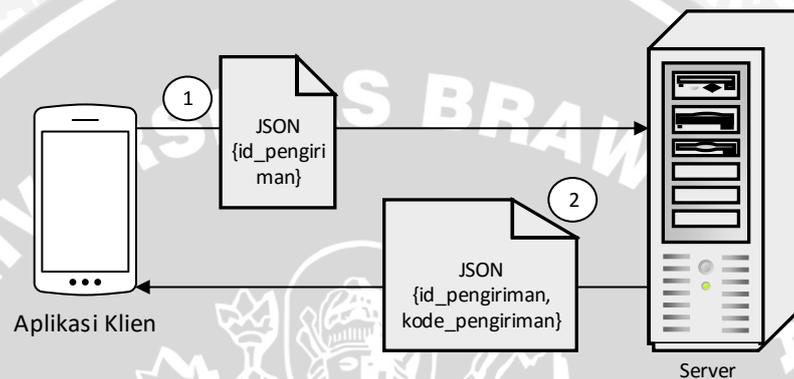
Gambar 4.10 Skema JSON menerima daftar data pengiriman baru

Keterangan: {} = data kompleks.

* = data majemuk.

Contoh data: [{4, j0d9tye}, {5, ml1d2n5}]

Alur pengiriman JSON pada proses ini ditunjukkan pada gambar 4.11.



Gambar 4.11 Alur transaksi JSON pada proses mengirim hasil konfirmasi pengiriman

Alur transaksi JSON pada Gambar 4.11 dapat dijelaskan seperti berikut:

- ①. Aplikasi klien mengirim JSON berisi ID pengiriman yang telah dikonfirmasi kepada server.
- ②. Server mengirim data JSON berisikan ID pengiriman dan kode pengiriman barang yang baru atau belum dikonfirmasi dengan ID kurir yang bersangkutan.

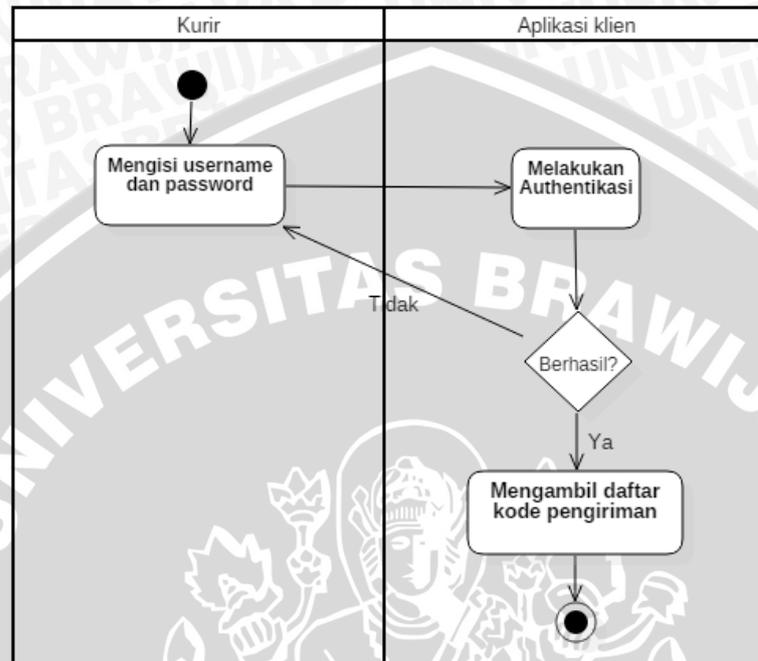
Dalam memenuhi kebutuhan non fungsional yang berupa keamanan maka data kode pengiriman barang yang hanya bisa diketahui oleh penerima barang akan dienkripsi, sehingga tidak dapat diketahui oleh orang lain. Proses enkripsi dilakukan pada sisi server dengan metode hash MD5, dan data kode pengiriman yang dikirim pada aplikasi klien dalam keadaan telah terenkripsi. Saat data kode pengiriman dalam kondisi terenkripsi dan ID pengiriman diterima oleh aplikasi klien data langsung dimasukkan ke dalam database dari aplikasi klien. Ketika melakukan konfirmasi pengiriman barang data kode pengiriman masukan yang didapatkan dari penerima barang akan di jadikan hash dengan MD5 terlebih dahulu untuk bisa disesuaikan dengan kode pengiriman yang ada dalam database aplikasi klien.

4.2.2. Perancangan Aktifitas Sistem

Dari perancangan arsitektual yang telah dibuat sebelumnya dan proses pemenuhan kebutuhan dalam use case diagram akan direpresentasikan dalam



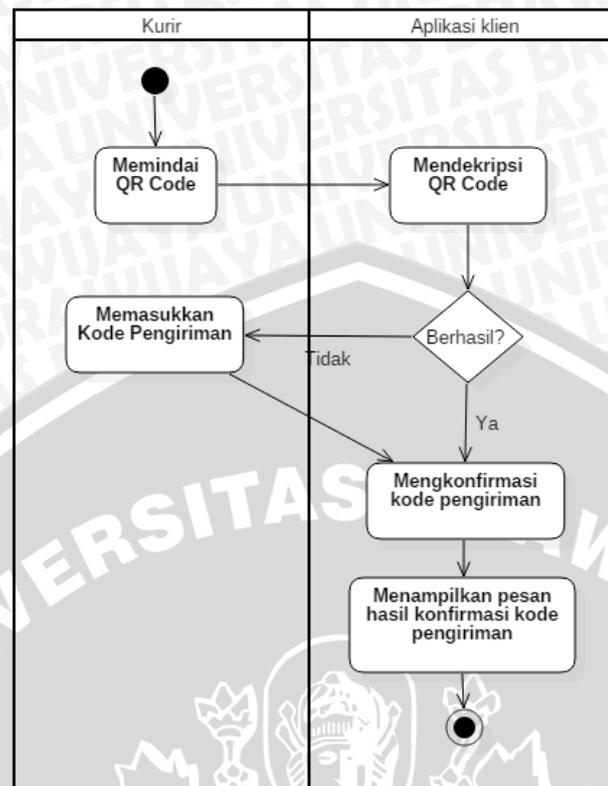
activity diagram atau diagram aktifitas dari pembuat sistem. Activity diagram adalah diagram yang menjelaskan pola aktifitas antara pengguna dan sistem yang berdasarkan pada skenario *use case*. Pada penelitian ini terdapat tiga *use case* yang telah ditunjukkan oleh gambar 4.3, yang pertama adalah *use case login*. Diagram aktifitas dari kebutuhan ini ditunjukkan pada gambar 4.12.



Gambar 4.12 Diagram aktifitas proses login.

Pada diagram aktifitas yang ditunjukkan gambar 4.12 dijelaskan alur aktifitas dari proses melakukan *login*. Proses diawali dengan kurir mengisi *username* dan *password* pada halaman *login* dari aplikasi klien. Kemudian aplikasi klien melakukan autentikasi *username* dan *password* pada *server* dari sistem konfirmasi pengiriman barang menggunakan *QR Code* pada perangkat bergerak. Akan diberikan hasil autentikasi dari *server*, jika hasil autentikasi benar aplikasi klien akan mengambil daftar kode pengiriman dan jika autentikasi gagal akan kembali ke halaman *login* dan kurir dapat memasukkan *username* dan *password* yang sesuai.

Use case yang kedua adalah mengkonfirmasi kode pengiriman barang. Untuk diagram aktifitas dari proses pemenuhan kebutuhan ini ditunjukkan pada gambar 4.13.

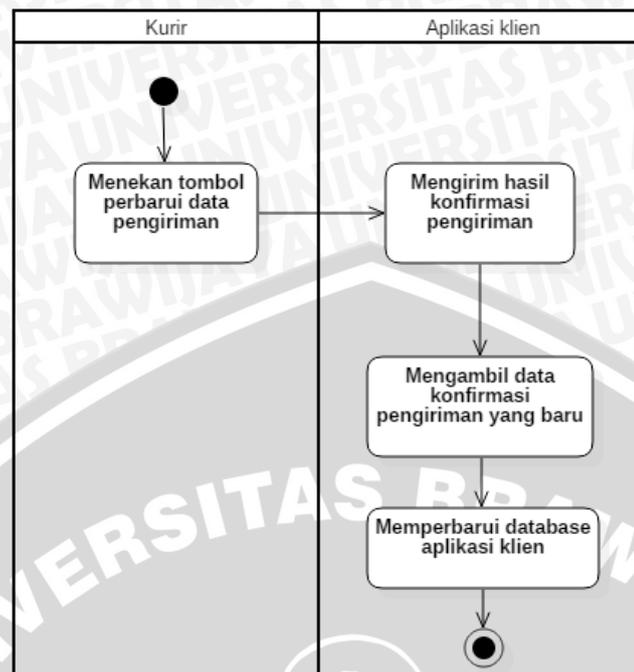


Gambar 4.13 Diagram aktifitas mengkonfirmasi pengiriman barang

Mengkonfirmasi pengiriman barang bisa dilakukan setelah kurir berhasil *login*. Setelah kurir berhasil *login* kurir akan bisa mengkonfirmasi pengiriman barang beberapa kali selama kurir belum *logout* meski telah keluar dari aplikasi aplikasi akan kembali pada halaman konfirmasi pengiriman begitu aplikasi dibuka kembali sehingga tidak perlu melakukan *login* ulang.

Gambar 4.13 menjelaskan tentang alur proses dari sistem untuk mengkonfirmasi pengiriman barang menggunakan sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak. Proses diawali dengan kurir memindai QR Code yang ditunjukkan oleh penerima barang. Kemudian QR Code didekripsi oleh aplikasi klien untuk menangkap kode pengiriman di dalamnya. Jika berhasil didapatkan kode pengirimannya kode tersebut akan dikonfirmasi dan jika gagal kurir bisa menggunakan cara manual dengan memasukkan kode pengiriman pada *textbox* yang disediakan. Setelah kode pengiriman dimasukkan akan dikonfirmasi dengan *database* dari aplikasi klien. Kemudian yang terakhir hasil konfirmasi akan dimunculkan oleh aplikasi klien dalam *alert*.

Use case yang ketiga adalah mengirim hasil konfirmasi pengiriman pada *server*. Pada proses ini kurir tidak banyak melakukan aktifitas. Penjelasan lengkapnya ditunjukkan oleh diagram aktifitas pada gambar 4.14.



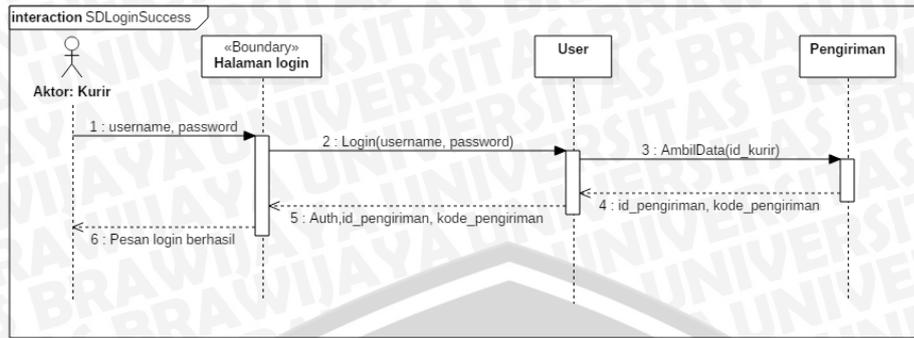
Gambar 4.14 Diagram aktifitas proses mengirim hasil konfirmasi pengiriman

Gambar 4.14 menunjukkan diagram aktifitas untuk proses mengirim hasil konfirmasi pengiriman barang pada *server*. Pada diagram aktifitas ini dijelaskan aktifitas-aktifitas yang terjadi dalam mengirim hasil konfirmasi pengiriman barang pada *server*.

Proses pada diagram aktifitas diawali dengan aktifitas kurir menekan tombol perbarui data pengiriman. Ketika tombol ditekan aplikasi klien akan mengirim kode pengiriman barang yang mempunyai status “Ter kirim” dari *database* aplikasi klien pada *server*. Kemudian *server* memperbarui data status untuk setiap kode pengiriman yang diterima dari aplikasi klien menjadi “Ter kirim” juga. Selanjutnya *server* membalasnya dengan mengirimkan kode pengiriman barang dengan *id_kurir* yang sama dengan kurir yang mengirim hasil konfirmasi pengiriman dan memiliki status “Belum Ter kirim”. Setelah itu aplikasi klien memperbarui *database* aplikasi klien.

4.2.3. Perancangan Alur Proses Dalam Sistem

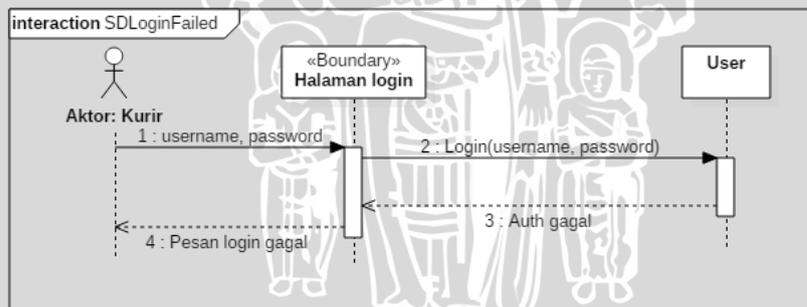
Setelah perancangan *activity diagram* selesai maka akan dibuat alur proses dari berjalan sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak yang direpresentasikan dalam bentuk *sequence diagram*. *Sequence diagram* adalah pemodelan alur proses interaksi antar objek yang disusun berdasarkan urutan waktu. *Sequence diagram* digunakan untuk menggambarkan rangkaian langkah-langkah yang dilakukan sebagai respon dari suatu aktifitas lain untuk menghasilkan sebuah *output*. Gambar 4.15 menunjukkan *sequence diagram* dari alur proses kurir melakukan *login* pada aplikasi klien.



Gambar 4.15 Sequence diagram untuk alur proses Login berhasil

Pada gambar 4.15 dijelaskan proses dari login ketika berhasil. Proses ini diawali dengan kurir mengisi form login dengan username dan password milik kurir pada halaman login. Setelah itu dari halaman login, username dan password diproses oleh method Login kepada object User. Selanjutnya method ambilData mengambil data ID pengiriman dan kode pengiriman dengan parameter id_kurir yang didapatkan kepada object Pengiriman. Setelah didapatkan hasilnya dari object Pengiriman, object User mengirim auth dengan status "true" dan data pada aplikasi klien dan aplikasi klien menampilkan pesan login berhasil.

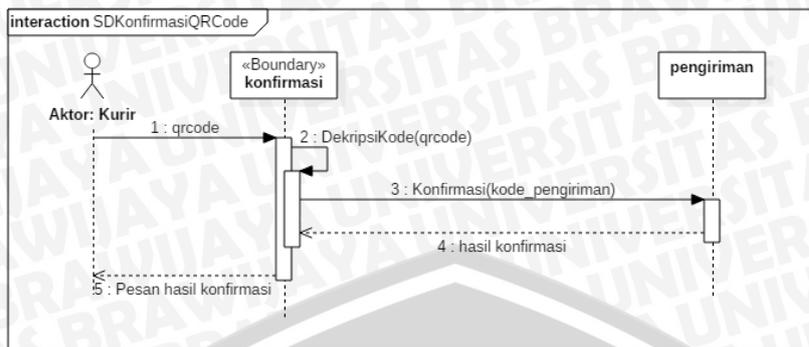
Berbeda dengan ketika proses autentikasi login berhasil, untuk autentikasi login yang gagal balasan untuk method Login adalah status auth "false" atau gagal sehingga tidak mengambil data pengiriman dan hanya mengirimkan hasil autentikasi pada aplikasi klien. Untuk sequence diagram dari alur proses ini ditunjukkan pada gambar 4.16.



Gambar 4.16 Sequence diagram untuk alur proses Login gagal

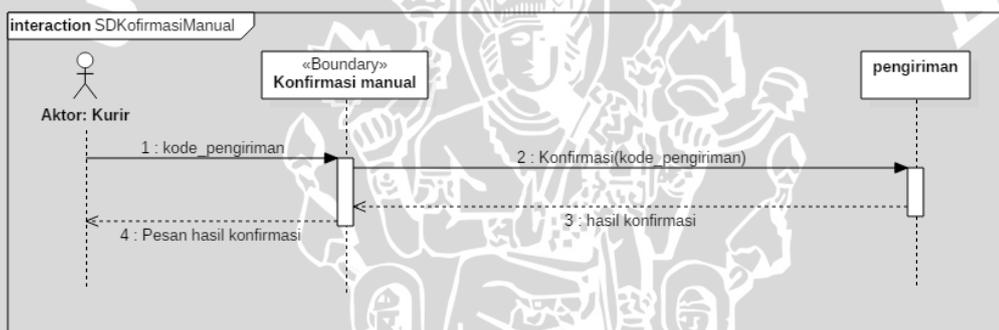
Setelah kurir berhasil login, kurir dapat melakukan konfirmasi kode pengiriman barang. Untuk konfirmasi pengiriman barang bisa dilakukan dengan dua cara scan QR Code dan memasukkan kode pengiriman. Untuk konfirmasi dengan cara scan QR Code ditunjukkan pada gambar 4.17.





Gambar 4.17 Sequence diagram dari proses konfirmasi pengiriman dengan cara scan QR Code

Pada awalnya kurir akan memindai QR Code dari penerima barang. Selanjutnya QR Code didekripsi. Lalu dari dekripsi QR Code dihasilkan kode pengiriman barang. Kode pengiriman barang kemudian dikonfirmasi dengan database pengiriman. Setelah dicari hasilnya akan dikirimkan ke boundary konfirmasi. Dan boundary akan menampilkan pesan hasil konfirmasi kode pengiriman barang.



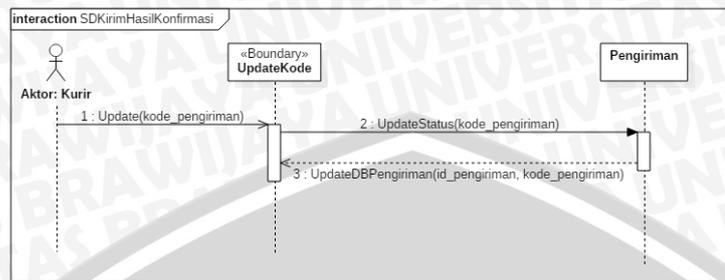
Gambar 4.18 Sequence diagram dari proses konfirmasi pengiriman dengan cara manual

Gambar 4.18 menjelaskan proses konfirmasi pengiriman barang dengan cara maual. Proses diawali dengan kurir memasukkan kode pengiriman pada halaman konfirmasi manual. Setelah itu kode pengiriman dikonfirmasi menggunakan *method* Konfirmasi dengan *object* pengiriman. Kemudian database memunculkan hasil pencarian. Berdasarkan hasil dari pencarian, hasil konfirmasi dikirim pada boundary Konfirmasi manual dan boundary menampilkan pesan hasil konfirmasi.

Setelah selesai dengan konfirmasi pengiriman barang kurir dapat mengirim hasil konfirmasi pengiriman pada server. Pengiriman hasil konfirmasi pada server ini adalah aplikasi klien mengambil data ID pengiriman dari database aplikasi klien yang mempunyai status "Ter kirim", kemudian mengirimnya ke server. Saat server menerima daftar ID pengiriman dari aplikasi klien, server memperbarui status dari daftar ID pengiriman yang sama dengan yang didapatkan dari aplikasi klien menjadi "Ter kirim" setelah itu kode pengiriman yang mempunyai status yang "Belum Ter kirim" dengan data kurir yang sama dikirim ke aplikasi klien. Sesaat



daftar data pengiriman diterima oleh aplikasi klien, *database* pada aplikasi klien diperbarui juga. Alur proses dari mengirim hasil konfirmasi pada aplikasi klien ini direpresentasikan dengan *sequence diagram* pada gambar 4.19.

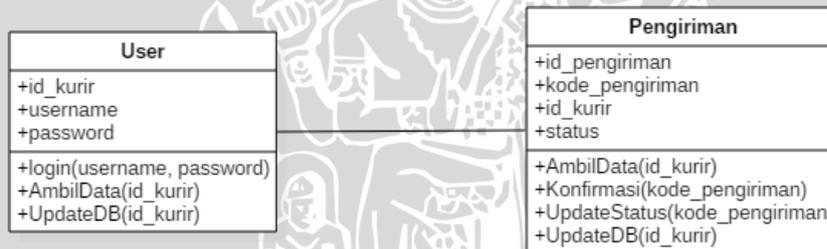


Gambar 4.19 *Sequence diagram* Mengirim hasil konfirmasi pengiriman

4.2.4. Perancangan Konseptual

Dari hasil *sequence diagram* yang telah dirancang akan didapatkan *class* yang dibutuhkan oleh sistem konfirmasi pengiriman barang menggunakan QR Code yang direpresentasikan dengan *class diagram*. *Class diagram* adalah gambaran pemodelan dari elemen-elemen *class* yang membentuk sebuah sistem.

Pada gambar 4.20 ditunjukkan perancangan *class diagram* untuk penelitian ini.



Gambar 4.20 *Class diagram* aplikasi klien

Pada gambar 4.20 ditunjukkan *class diagram* dari klien pada *class diagram* terdapat dua *class*, yaitu *class User* dan *class Pengiriman*. Pada *class User* terdapat *method login()*. *Method* ini beroperasi untuk melakukan *login* dengan menggunakan parameter *username* dan *password*. Kemudian pada *class User* terdapat atribut *id_kurir*, *id_kurir* ini digunakan untuk membedakan antara kurir yang satu dengan yang lainnya dan untuk menjadi parameter penghubung pada *class Pengiriman*. Selain itu terdapat *method AmbilData()* dengan parameter *id_kurir*. *Method* ini mengambil data *id_kurir* dari *class User* ketika ingin mengambil daftar kode transaksi pada *class Pengiriman*.

Pada *class Pengiriman* terdapat *method AmbilKode()* dengan parameter *id_kurir*. *Method* ini beroperasi untuk mengirim daftar kode pengiriman pada aplikasi klien dan disimpan ke dalam *database* klien. *Method* kedua adalah *konfirmasi()* yang memiliki parameter *kode_konfirmasi*. *Method* ini digunakan untuk mengkonfirmasi kode pengiriman barang yang diterima oleh aplikasi klien. Pada *class Pengiriman* juga terdapat atribut *id_kurir*. Atribut ini digunakan

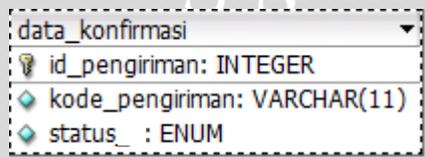
parameter `AmbilKode()`. Atribut ini diambil dari *class* `User` untuk memastikan bahwa kurir yang melakukan konfirmasi kode pengiriman barang adalah kurir yang terdaftar pada sistem. Kemudian ada atribut `status`, atribut ini digunakan untuk memuat status pengiriman.

4.2.5. Perancangan Basis Data

Berdasarkan *Class diagram* yang dirancang pada sub-bab sebelumnya akan dibuat sebuah basis data yang akan digunakan oleh sistem konfirmasi pengiriman barang menggunakan *QR Code* pada perangkat bergerak. Basis data pada sistem konfirmasi pengiriman barang berfungsi untuk menyimpan data-data pada aplikasi klien yang diperlukan untuk melakukan konfirmasi pengiriman barang. Perancangan basis data ini ditunjukkan dalam bentuk konseptual diagram. Konseptual diagram menunjukkan hubungan antara objek (entitas) yang terlibat dalam suatu basis data. Konseptual diagram berisi himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan beberapa atribut yang mewakili seluruh fakta yang ditinjau dari keadaan yang nyata.

Pada penelitian ini terdapat dua basis data yang digunakan. Pertama adalah basis data yang berada pada *server*. Basis data ini berfungsi untuk menyimpan keseluruhan data yang dibutuhkan oleh sistem. Namun pada penelitian ini *server* hanya diasumsikan karena perancangan berfokus pada proses berjalannya aplikasi klien. Basis data kedua adalah basis data yang digunakan pada aplikasi klien yang berfungsi menyimpan data yang diterima dari *server*. Basis data pada aplikasi klien bersifat sementara, karena basis data akan berubah setiap kali *user* yang melakukan *login* berubah.

Pada basis data dari aplikasi klien hanya memiliki satu entitas, yaitu `data_konfirmasi`. Hal ini karena data yang tersimpan hanya untuk kebutuhan konfirmasi kode pengiriman barang yang dimiliki oleh penerima barang. Pada entitas `data_konfirmasi` terdapat dua atribut, yaitu `kode_konfirmasi` dan `status`. Atribut `kode_konfirmasi` ini menyimpan data kode pengiriman yang didapatkan dari *database server*. Sedangkan atribut `status` berisikan status konfirmasi pengiriman barang. Untuk diagram konseptual basis data dari aplikasi klien dijelaskan pada gambar 4.21.



data_konfirmasi	
id_pengiriman	: INTEGER
kode_pengiriman	: VARCHAR(11)
status_	: ENUM

Gambar 4.21 Basis data pada aplikasi klien

Penjelasan untuk tabel yang akan dibuat pada basis data:

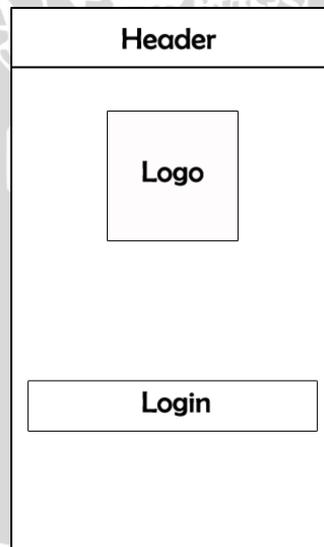
Tabel 4.2 Rancang table untuk menyimpan data pengiriman.

Tabel data_pengiriman			
Kolom	Type Data	Keterangan	Contoh Data
id_pengiriman	Int	primary key	1
Kode_pengiriman	Text	unique	J35n92d
status	Text		Belum Terkirim

Pada tabel 4.2 dijelaskan tentang susunan dari rancangan tabel untuk menyimpan data pengiriman pada *database* aplikasi klien. Pada tabel ini terdiri dari *id_pengiriman*, *kode_pengiriman*, dan *status*, berfungsi untuk menyimpan data pengiriman yang telah diterima dari *server*.

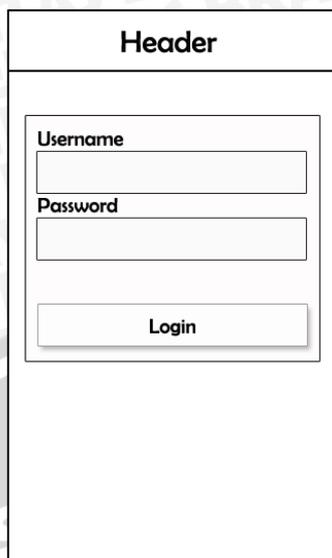
4.2.6. Perancangan Antar Muka

Perancangan. Antar muka merupakan implementasi antar muka dari sistem konfirmasi pengiriman barang menggunakan *QR Code* pada perangkat bergerak yang akan dibangun. Dalam sistem ini antar muka yang akan dirancang adalah aplikasi klien yang merupakan aplikasi yang berjalan pada perangkat bergerak yang dimana mempunyai ukuran layar yang kecil. Aplikasi pada perangkat bergerak mempunyai tampilan yang bisa menyesuaikan dengan perangkat yang digunakan.



Gambar 4.22 Antar muka tampilan awal aplikasi klien

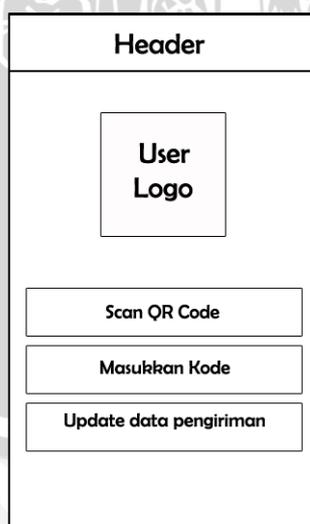
Gambar 4.22 adalah antar muka dari tampilan awal aplikasi klien. Halaman ini hanya memuat logo dan tombol *login*. Karena untuk menggunakan fungsi-fungsi dari aplikasi klien membutuhkan *login*. Jadi melakukan *login* harus dilakukan terlebih dahulu. Sehingga jika tombol *login* ditekan *user* akan diarahkan pada halaman *form login* yang ditunjukkan pada gambar 4.23.



The image shows a login form with a header. Below the header, there are two text input fields. The first is labeled 'Username' and the second is labeled 'Password'. Below these fields is a button labeled 'Login'.

Gambar 4.23 Antar muka form login

Pada gambar 4.23 terdapat dua *textbox* yang ber-label *Username* dan *Password*. Pada halaman ini *user* diminta untuk memasukkan *username* dan *password* miliknya. Tombol *Login* digunakan untuk menyudahi mengirim *form login* ke *server* untuk diotentikasi. Setelah melakukan *login*, jika berhasil maka *user* akan diarahkan ke halaman utama yang berisikan menu-menu utama dari aplikasi klien, yaitu *Scan QR Code*, *Masukkan kode*, dan *Update Database*. Menu *Scan QR Code* berfungsi untuk mengaktifkan kamera yang digunakan *scan QR Code*, Tombol *Masukkan kode* adalah untuk memasukkan kode pengiriman secara manual dan tombol *Update Database* berfungsi untuk meng-*update database* dari *server* dan aplikasi klien. Antar muka untuk halaman utama ditunjukkan pada gambar 4.24.

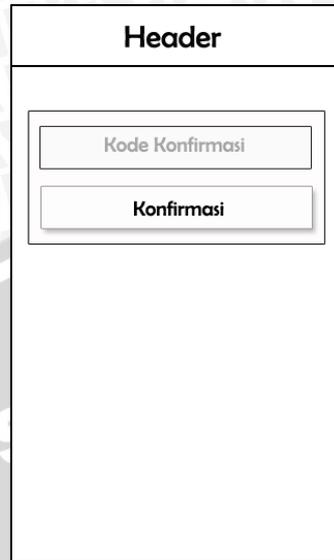


The image shows the main application interface. It has a header section. Below the header, there is a box labeled 'User Logo'. Below that, there are three buttons: 'Scan QR Code', 'Masukkan Kode', and 'Update data pengiriman'.

Gambar 4.24 Antar muka halaman utama aplikasi klien

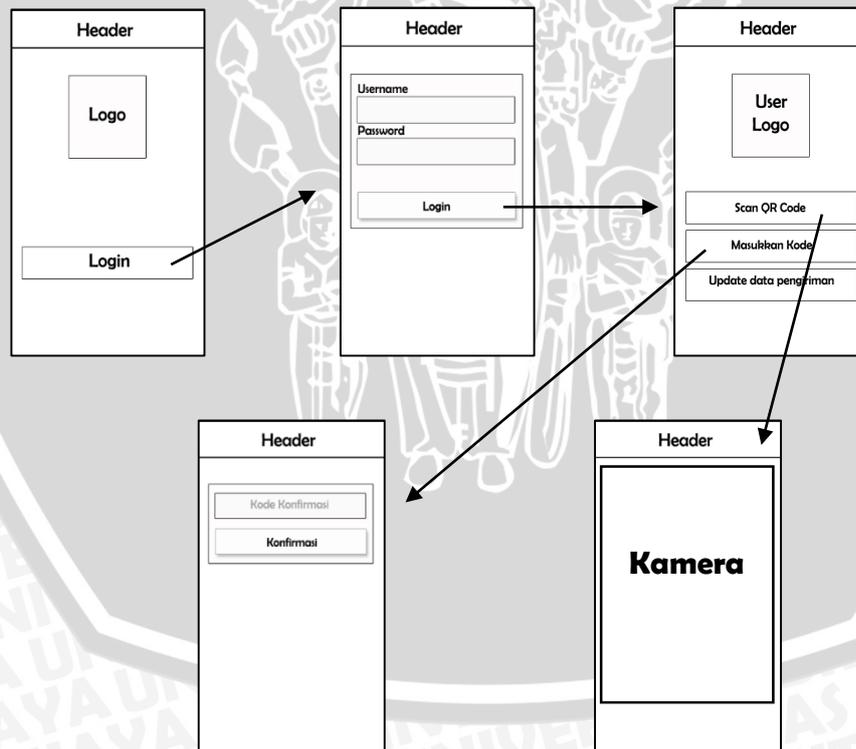
Saat tidak bisa menggunakan *scan QR Code* alternative yang digunakan adalah memasukkan kode secara manual. Gambar 4.25 menunjukkan antar muka dari tampilan aplikasi klien pada halaman masukkan kode. Halaman ini hanya

terdapat *textbox* untuk memasukkan kode pengiriman dan tombol Konfirmasi yang berfungsi untuk mengirim mengkonfirmasi kode yang telah dimasukkan.



Gambar 4.25 Antar muka halaman masukkan kode manual

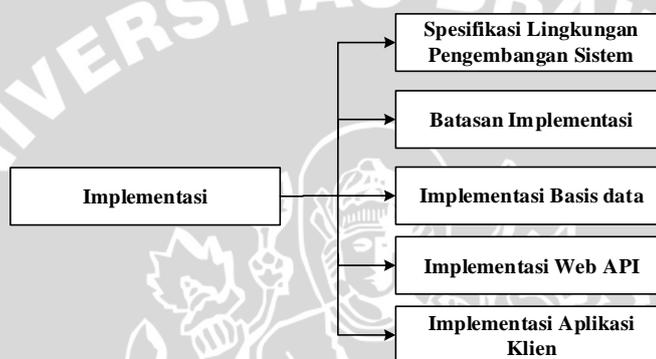
Skema alur dari antar muka aplikasi klien ini bisa digambarkan seperti pada gambar 4.26.



Gambar 4.26 Skema alur antar muka aplikasi klien

BAB V IMPLEMENTASI

Bab ini membahas tentang implementasi dari pembuatan aplikasi dari sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak. Pada tahap ini seluruh analisis kebutuhan dan perancangan sistem akan diimplementasikan ke dalam kode-kode bahasa pemrograman yang telah ditentukan. Sebelum melakukan implementasi terlebih dahulu dijelaskan spesifikasi lingkungan pengembangan dan batasan implementasi. Sehingga implementasi pada penelitian ini jika direpresentasikan pada diagram dapat dilihat pada gambar 5.1.



Gambar 5.1 Skema implementasi

Implementasi dari penelitian ini diawali dengan penjelasan tentang spesifikasi lingkungan pengembangan sistem, batasan-batasan implementasi, pelaksanaan implementasi basis data, implementasi web API dan implementasi aplikasi klien.

5.1 Spesifikasi Lingkungan Pengembangan Sistem

Sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak dikembangkan dalam lingkungan implementasi yang melibatkan perangkat keras dan perangkat lunak.

1. Spesifikasi lingkungan perangkat keras

Lingkungan perangkat keras yang digunakan untuk implementasi sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak terdiri dari komputer *notebook* dan perangkat bergerak *smartphone*. Untuk spesifikasi komputer *notebook* adalah:

- *processor Intel Core I3*
- RAM (*Random Akses Memory*) 4GB.

Sedangkan untuk spesifikasi perangkat bergerak adalah *smartphone* dengan *Operating System Android 4.2.2 (Jelly Bean)* dengan RAM 512MB dan *smartphone* lain yang menggunakan *Operating System Android*.

2. Spesifikasi lingkungan perangkat lunak

Lingkungan perangkat lunak yang digunakan untuk implementasi dari penelitian ini disebutkan pada tabel 5.1

Tabel 5. 1 Spesifikasi lingkungan perangkat lunak

<i>Operating System</i>	Microsoft Windows 8.1
<i>Programing Language</i>	<ul style="list-style-type: none"> • PHP (untuk <i>web API</i>) versi 5.5 • HTML, CSS, Javascript (untuk aplikasi klien / <i>Android Hybrid</i>)
<i>Text Editor / IDE</i>	<i>Sublime text 3</i>
<i>Server</i>	Apache HTTP <i>Server</i> versi 2.4.10
<i>Database</i>	MYSQL <i>Database</i> versi 4.1.61 SQLite (untuk <i>database</i> aplikasi klien)

5.2 Batasan Implementasi

Pada sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak terdapat batasan-batasan pada saat proses implementasi dilakukan, diantaranya:

1. Aplikasi membutuhkan koneksi ke *internet* untuk *login* sebelum melakukan konfirmasi pengiriman barang. Dan *update database* pengiriman.
2. Sistem ini diperuntukkan pada pengiriman barang dari situs jual beli *online*.
3. Kecepatan akses tergantung pada kecepatan dari koneksi *internet* yang digunakan. Dalam penelitian ini kecepatan proses aplikasi tidak diperhitungkan.
4. *Database* disimpan pada perangkat lunak MySQL pada *server* dan SQLite untuk penyimpanan *database* pada aplikasi klien.

5.3 Implementasi Basis Data

Pada sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak menggunakan penyimpanan basis data MySQL dan dimodelkan dengan *physical diagram*. Dari *physical diagram* akan dibuat basis data pada MySQL dengan menggunakan perintah-perintah *Query* (bahasa perintah dalam basis data) atau DDL (*Data Definition Language*). Pada penelitian

ini terdapat dua *database* yang dibuat, yaitu *database* pada aplikasi klien dan *database* pada *server*. Kode DDL untuk *database* pada aplikasi klien ditunjukkan pada kode 5.1. dan kode DDL untuk *database* pada *server* terdapat pada lampiran 1.

```
CREATE DATABASE konfirmasi
CREATE TABLE IF NOT EXISTS data_konfirmasi (id_pengiriman int
PRIMARY KEY, kode_pengiriman text UNIQUE, status text)
```

Kode 5.1 DDL Basis data pada aplikasi klien

5.4 Implementasi Web API

Web API dari sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak berjalan pada *server* yang menggunakan Apache HTTP Server dan dibuat menggunakan bahasa pemrograman PHP.

Pertama adalah fungsi untuk terhubung dengan basis data. Pada sistem ini terdapat *script* php yang bernama *connect.php*. Pada *connect.php* menggunakan fungsi *mysqli_connect()* untuk menghubungi basis data pada MySQL dalam *server*. Kode *connect.php* dijelaskan pada lampiran 2.

Selain fungsi menghubungkan *server* dengan *database* terdapat *script* yang berfungsi untuk memproses autentikasi *login* dan *script* yang berfungsi untuk memperbarui *database* ketika aplikasi klien melakukan *update* data pengiriman barang. *Script* untuk *login* terdapat pada *login.php* dan untuk *update* terdapat pada *script* PHP *update.php*. *Script* *login.php* ditunjukkan pada kode 5.2 dan *script* *update.php* ditunjukkan pada kode 5.3.

```
1 <?php
2 header("access-control-allow-origin: *") ;
3 include "connect.php";
4
5 $uname = $_POST['username'];
6 $passwd = $_POST['password'];
7 $que_data = "SELECT * FROM pengiriman WHERE id_kurir =
8 (SELECT id_kurir FROM user WHERE username = '$uname')
9 AND status = 'Belum Terkirim!";
10 $res_data = mysqli_query($dbc, $que_data);
11 $i = 1;
12 while ($data = mysqli_fetch_array($res_data)) {
13     $id[] = $data['id_pengiriman'];
14     $kode[] = md5($data['kode_pengiriman']);
15     $i = $i+1;
16 }
17
18 $query = "SELECT * FROM user WHERE username = '$uname'
19 AND password = '$passwd'";
20 $result = mysqli_query($dbc, $query);
21
22 if(mysqli_num_rows($result)>0){
23     $output = array('status' => true, 'message' => $uname,
24 'kode' => $kode, 'id' => $id);
25     echo json_encode($output);
```

```

26     }else{
27         $output = array('status' => false);
28         echo json_encode($output);
29     }
30     ?>

```

Kode 5.2 login.php

Pada kode 5.2 baris 7 sampai 15 berperan untuk mengambil data konfirmasi pengiriman barang yang belum terkirim dari *database server* dengan ID kurir yang sama dengan ID kurir dari *username* yang diterima dan kode pengiriman dienkripsi dengan metode MD5. Kemudian untuk autentikasi dari proses *login* terdapat pada baris 18 sampai 29. Pada baris 22 dan 25 merupakan proses jika autentikasi berhasil akan mengirim status *true* data yang telah didapat dari *database server* untuk dikirim pada aplikasi klien. Sedangkan pada baris 26 sampai 28 adalah jika autentikasi gagal akan mengirim status *false* pada aplikasi klien.

```

1  <?php
2      header("access-control-allow-origin: *") ;
3      include "connect.php";
4
5      $data_baru =
6      json_decode(stripslashes($_POST['data']));
7      $uname = $_POST['username'];
8      $info = count($data_baru);
9      $data = $_POST['data'];
10
11     if ($data_baru != 'null' || $data_baru != null ||
12     $data_baru != '' || $data_baru != 0 || $data_baru !=
13     'undefined') {
14         for ($i=0; $i<count($data_baru); $i++){
15             $upd_pengiriman = "UPDATE pengiriman SET status =
16             'Ter kirim' WHERE id_pengiriman =
17             '". $data_baru[$i]. "'";
18             $res_upd_p = mysqli_query($dbc, $upd_pengiriman);
19         };
20
21         $upd_data = "SELECT * FROM pengiriman WHERE id_kurir
22         = (SELECT id_kurir FROM user WHERE username =
23         '$uname') AND status = 'Belum Ter kirim'";
24         $res_upd = mysqli_query($dbc, $upd_data);
25         $count = 1;
26         while ($data = mysqli_fetch_array($res_upd)) {
27             $id_baru[] = $data['id_pengiriman'];
28             $kode_upd[] = md5($data['kode_pengiriman']);
29             $count = $count+1;
30         }
31
32         if (mysqli_num_rows($res_upd)>0) {
33             $output = array('status' => true, 'dataBaru' =>
34             $kode_upd, 'id' => $id_baru, 'info' => $info, 'data'
35             => $data);
36             echo json_encode($output);
37         } else {
38             $output = array('status' => false);

```

```

39         echo json_encode($output);
40     }
41     } else{
42         $output = array('status' => 'data empty');
44         echo json_encode($output);
45     }
46     ?>

```

Kode 5.3 update .php

Pada kode 5.3 menjelaskan tentang kode PHP untuk proses *update* data hasil konfirmasi pengiriman barang. Baris 14 sampai 19 adalah kode untuk proses *update* status pengiriman menjadi terkirim untuk setiap ID pengiriman yang sama dengan yang diterima dari aplikasi klien. Baris 21 dan 30 adalah untuk mencari data pengiriman yang belum terkirim dan memiliki ID kurir yang sama dengan pengirim data hasil konfirmasi pengiriman dan mengenkripsi kode pengiriman dengan metode MD5. Baris 32 sampai 44 adalah untuk proses mengirim data pada aplikasi klien

5.5 Implementasi Aplikasi Klien

Implementasi aplikasi klien menjelaskan tentang penerapan fungsi-fungsi yang ada berdasarkan kebutuhan dan perancangan yang telah dibuat pada pembahasan sebelumnya. Pada aplikasi klien terdapat beberapa fungsi, diantara fungsi *login*, fungsi konfirmasi kode pengiriman, dan fungsi *update* data pengiriman barang pada *server*. Fungsi-fungsi ini akan dibuat dalam bahasa pemrograman *Web* dengan kata lain terbuat dari HTML, CSS, dan Javascript.

Terdapat beberapa *script* kode dalam implementasi aplikasi klien dari sistem konfirmasi pengiriman barang menggunakan *QR Code* pada perangkat bergerak pada penelitian ini, antara lain *index.html*, *index.js*, *index.css*, *main.js*, *barcodescanner.js*, *konfirmasi.js*, *sinkronDB.js*, *database.js*.

index.html, *index.css* dan *index.js* adalah *script* untuk tampilan utama dari aplikasi klien seperti pada lampiran 3, lampiran 4, dan lampiran 5. Pada *index.html* juga berisikan *script* Javascript untuk proses yang lebih sederhana. Aplikasi klien menggunakan *framework JQuery Mobile* sebagai tampilan utama, selain *framework JQuery Mobile* untuk tampilan terdapat beberapa *dependency* lain dari *framework Cordova* yang berfungsi sebagai media penghubung pada sistem *Android*. Untuk sistem konfirmasi pengiriman barang menggunakan *QR Code* pada perangkat bergerak sendiri terdiri dari beberapa fungsi yang dibuat pada beberapa *file javascript*. Pertama adalah *script main.js* berisikan kode-kode javascript yang berfungsi untuk *login* dan *logout* seperti dijelaskan pada kode 5.4.

```

1     $(document).on('pageinit', '#login', function() {
2         $(document).on('click', '#submit', function() {
3
4             if($('#username').val().length>0&&$('#password').val
5                 ().length>0) {
6                 var un = $('#username').val();

```

```
7 var pw = $('#password').val();
8 $.ajax({
9
10 url:'http://qrkonfirmasi.16mb.com/delivery/login.php
11 ',
12 data:{ username : un,
13 password : pw
14 },
15 type:'post',
16 async:'false',
17 dataType: 'json',
18 beforeSend:function(){
19 $.mobile.loading ('show', {theme:"a",
20 text:"Login...", textonly:true, textVisible:true});
21 },
22 complete:function(){
23 $.mobile.loading('hide');
24 },
25 success:function(result){
26 console.log('ajax result: '+result);
27 if(result.status==true){
28 user.name=result.message;
29 if (result.kode.length > 0) {
30 console.log('Kode Length:
31 '+result.kode.length);
32 kodePengiriman = [];
33 idPengiriman = [];
34 for (var i =0; i < result.kode.length; i++) {
35 kodePengiriman[i]= result.kode[i];
36 idPengiriman[i] = result.id[i];
37 };
38 queryDB();
39 } else {
40 alert('Tidak ada pengiriman barang');
41 }
42 $.mobile.changePage("#konfirmasi");
44 window.localStorage.setItem('uname', un);
45 window.localStorage.setItem('passwd', pw);
46 console.log('Login berhasil');
47 } else{
48 alert('Login gagal. Username atau password tidak
49 sesuai');
50 }
51 },
52 error:function(request,error){
53 alert('Koneksi error. Silahkan coba beberapa saat
54 lagi!');
55 }
56 });
57 }else{
58 alert('Masukkan username dan password!');
59 }
60 return false;
61 });
62 });
63
```

```

64 $(document).on('pagebeforeshow', '#konfirmasi', function ()
65 {
66     $.mobile.activePage.find('.welcome').html('<h3>Selamat
67     Datang '+user.name+'</h3>' );
68 });
69
70 function logOutUser() {
71     sinkronMyDB();
72     window.localStorage.clear();
73     dropDB();
74     $.mobile.changePage("#home");
75 }
76
77 function exitFromApp() {
78     navigator.app.exitApp();
79 }
80
81 var user={name>window.localStorage['uname']};

```

Kode 5.4 main.js

Pada kode 5.4 baris 4 sampai 62 dijelaskan fungsi untuk *login* ketika kotak teks *username* dan *password* tidak kosong, pada baris 70 sampai 75 adalah fungsi untuk *log out* dan pada baris 77 sampai 79 adalah fungsi untuk keluar dari aplikasi. Pada fungsi *login* ketika aplikasi menerima hasil status “true” berarti berhasil *login* dan aplikasi akan memasukkan data konfirmasi pengiriman yang diterima dari *server* ke dalam *database* aplikasi klien dan mengarahkan ke halaman konfirmasi seperti dijelaskan pada baris 21 sampai 46. Ketika menerima hasil status “false” maka berarti *login* gagal dan aplikasi akan menampilkan pesan “Login gagal. Username atau password tidak sesuai”. Pada fungsi *log out* saat melakukan *log out* aplikasi akan melakukan mengirim hasil konfirmasi pengiriman setelah selesai akan menghancurkan *session username* yang disimpan dan men-drop tabel data pengiriman yang disimpan pada *database* klien, setelah semua proses selesai akan mengarah ke halaman awal aplikasi.

Selanjutnya *script* `barcodescanner.js`, *script* ini berfungsi menghubungkan `index.html` dengan *plugin Barcode Scanner* dari Cordova untuk melakukan *scan QR Code*. Untuk *script* ini ditunjukkan pada kode 5.5.

```

1 document.addEventListener("deviceready", init, false);
2     function init() {
3     }
4
5     function scan() {
6         log("scanning");
7
8         cordova.plugins.barcodeScanner.scan(scanSuccessCallb
9         ack, scanErrorCallback);
10    }
11
12    function scanSuccessCallback(result) {
13        log(JSON.stringify(result));
14        console.log('Scanning success');
15        var kode = result.text;

```

```

16     inputKode = CryptoJS.MD5(kode);
17     // $.getScript('database.js', function(){
18         confirmCode();
19     // });
20 }
21
22 function scanErrorCallback(error) {
23     alert("Scanning failed: " +
24     JSON.stringify(error));
25 }
26
27 function encode() {
28     log("encoding");
29     if (device.platform == "Android") { //Not
30     supported on iOS
31         var stringToEncode = "http://www.sap.com";
32
33     cordova.plugins.barcodeScanner.encode(cordova.plugin
34     s.barcodeScanner.Encode.TEXT_TYPE, stringToEncode,
35     encodeSuccessCallback, encodeErrorCallback);
36     }
37     else {
38         log("Encoding is not supported on iOS. See
39     https://github.com/wildabeast/BarcodeScanner/issues/
40     106");
41     }
42 }
43
44
45 function encodeSuccessCallback(result) {
46     log(JSON.stringify(result));
47 }
48
49 function encodeErrorCallback(error) {
50     alert("Encoding failed: " +
51     JSON.stringify(error));
52 }
53
54 function log(line) {
55     var results =
56     document.getElementById("scan_results");
57     //results.innerHTML+= "<br>" + line;
58 }

```

Kode 5.5 barcodescanner.js

Pada kode 5.5 terdapat beberapa fungsi untuk melakukan *scan QR Code*. Contohnya pada baris 12 sampai 20 adalah fungsi ketika berhasil melakukan *scan QR Code* dan mendapat kode konfirmasi yang tersimpan didalam *QR Code* kode tersebut akan di *hash* menggunakan metode MD5, kemudian kode *hash*-nya di cek dengan data kode pengiriman barang yang terdapat pada *database* dalam aplikasi.

Script selanjutnya adalah *konfirmasi.js script konfirmasi.js* digunakan untuk melakukan proses konfirmasi kode pengiriman secara manual. *konfirmasi.js* ditunjukkan pada kode 5.6.

```

1 $(document).on('pageinit','#manual',function(){
2     $(document).on('click','#submit',function(){
3         if($('#kode').val().length>0){
4             inputKode = CryptoJS.MD5($('#kode').val());
5             confirmCode();
6         }else{
7             alert('Masukkan kode transaksi!');
8         }
9     });
10 });

```

Kode 5.6 konfirmasi.js

Dari kode 5.6 dapat dilihat hanya terdapat sedikit fungsi. Pada konfirmasi.js hanya terdapat fungsi membuat kode *hash* dari kode pengiriman barang yang dimasukkan oleh pengguna pada aplikasi, selanjutnya memanggil fungsi untuk melakukan konfirmasi kode *hash* dengan *database* dalam aplikasi sama seperti fungsi konfirmasi yang dipanggil pada `barcodescanner.js`

Kemudian *script* `sinkronDB.js` berisikan kode javascript yang berfungsi untuk melakukan *update* data pengiriman barang. Ditunjukkan pada kode 5.7.

```

1 function sinkronMyDB() {
2     sinkronDB(function (dataSync) {
3         var u_name = window.localStorage.getItem('uname');
4         if (dataSync.length !== 0) {
5             dataSinkronItem = JSON.stringify(dataSync);
6             console.log('datasync length = '+dataSync.length);
7             console.log('dataSinkronItem length =
8 '+dataSinkronItem.length);
9             $.ajax({
10
11             url:'http://qrkonfirmasi.16mb.com/delivery/update.ph
12 p',
13             data: {data : dataSinkronItem, username:
14 u_name},
15             type:'post',
16             async:'false',
17             dataType: 'json',
18             beforeSend:function(){
19                 $.mobile.loading ('show', {theme:"a",
20 text:"Update...", textonly:true, textVisible:true});
21             },
22             complete:function(){
23                 $.mobile.loading('hide');
24             },
25             success:function(result){
26                 if (result.status===true) {
27                     var len = result.dataBaru.length;
28                     console.log('databaru length =' +len);
29                     dataBaru = [];
30                     idBaru = [];
31                     for (i=0; i<len; i++){
32                         dataBaru[i] = result.dataBaru[i];
33                         idBaru[i] = result.id[i];
34                     }

```

```

35     sinkronUpd();
36     delConfirmed();
37     alert('Database update success.');
```

38	} else{
39	alert('Database update success. Tidak
40	ada pengiriman baru.');
41	}
42	},
44	error:function(request,error) {
45	alert('Koneksi error. Silahkan coba beberapa
46	saat lagi!');
47	alert(error);
48	}
49	});
50	}else alert('Belum ada barang yang terkirim');
51	});
52	}

Kode 5.7 sinkronDB.js

Pada kode 5.7 baris 9 sampai 25 merupakan format JSON yang memanfaatkan ajax dalam mengirim data dari *javascript* ke *server* dengan PHP. Pada baris 26 sampai 37 adalah proses disaat mendapat balasan dari *server* dengan status *"true"* berarti terdapat data pengiriman barang yang baru jika demikian data pengiriman baru yang diterima akan dimasukkan ke dalam *database* aplikasi klien, dan jika berhasil akan menampilkan pesan *update* data pengiriman berhasil. Apabila status *"false"* maka berarti tidak ada data pengiriman baru untuk id kurir tersebut sehingga aplikasi menampilkan pesan *"Database update success. Tidak ada pengiriman baru"*. Untuk baris 44 sampai 48 merupakan *error handling* ketika tidak dapat mengirim data hasil pengiriman ke *server*.

Terakhir adalah *database.js*. *script* ini berisi fungsi-fungsi untuk melakukan transaksi *database* pada SQLite. Semua proses *query database* terdapat pada *script* ini, mulai dari membuat *database* sampai mencari data pada *database* juga terdapat disini. *Script database.js* ditunjukkan pada kode 5.8.

```

1 document.addEventListener("deviceready", connectDB,
2     false);
3
4 //create or open Database
5 function connectDB(){
6     db = window.sqlitePlugin.openDatabase("konfirmasi",
7         "1.0", "Data Konfirmasi Pengiriman", "1000");
8     db.transaction(populateDB,errorCB,successCB);
9 }
10
11 //create table and insert some record
12 function populateDB(tx) {
13     tx.executeSql("CREATE TABLE IF NOT EXISTS
14         data_pengiriman (id_pengiriman int PRIMARY KEY,
15         kode_pengiriman text UNIQUE, status text)");
16 }
17
18 //function will be called when an error occurred
```

```
19 function errorCallback(err) {
20     console.log("Error processing SQL: "+err.message);
21 }
22
23 //function will be called when process succeed
24 function successCB() {
25     console.log("Connected to database.");
26 }
27
28 function queryDB(){
29     db.transaction(insertDB,errorCB,sucessExec);
30 }
31
32 function insertDB(tx){
33     var len = kodePengiriman.length;
34     console.log('Data length: '+len);
35     tx.executeSql("CREATE TABLE IF NOT EXISTS
36         data_pengiriman (id_pengiriman int PRIMARY KEY,
37         kode_pengiriman text UNIQUE, status text)");
38     for (var i = 0; i < len; i++) {
39         tx.executeSql("INSERT OR IGNORE INTO data_pengiriman
40             VALUES
41             ('"+idPengiriman[i]+"','"+kodePengiriman[i]+"',
42             'Belum Terkirim')");
43         console.log('Inserted kode for id
44             '+idPengiriman[i]+' = '+kodePengiriman[i]);
45     }
46 }
47
48
49 function successExec(){
50     console.log('Execution successful. ');
51 }
52
53 function dropDB(){
54     db.transaction(dropQuery,errorDrop,sucessDrop);
55 }
56
57 function dropQuery(tx){
58     tx.executeSql("DROP TABLE data_pengiriman");
59 }
60
61 function successDrop(){
62     console.log('Drop table successful');
63 }
64
65 function errorDrop(err){
66     console.log('Drop table unsuccessful, Error code:
67         '+err.code);
68 }
69
70 function selectData(err){
71     db.transaction(selectQuery,errorCB);
72 }
73
74 function confirmCode(){
75     db.transaction(confirmQuery,errorCB);
```

```
76 }
77
78 function confirmQuery(tx){
79     tx.executeSql("SELECT * FROM data_pengiriman WHERE
80         kode_pengiriman = '"+inputKode+"' AND status =
81         'Belum Terkirim'", [], successQuery);
82 }
83
84 function successQuery(tx, results) {
85     var len = results.rows.length;
86     if (len !== 0){
87         idPengirimanUpd =
88         results.rows.item(0).id_pengiriman;
89         db.transaction(updateDB,errorCB);
90         console.log('Kode =
91         '+results.rows.item(0).kode_pengiriman);
92         alert('[Error!] Kode pengiriman telah dikonfirmasi.
93         Id pengiriman:
94         '+results.rows.item(0).id_pengiriman);
95     } else {
96         tx.executeSql("SELECT * FROM data_pengiriman WHERE
97         kode_pengiriman = '"+inputKode+"' AND status =
98         'Terkirim'", [], errConfirm);
99     }
100 }
101
102 function errConfirm(tx, results){
103     var len = results.rows.length;
104     if (len !== 0) {
105         alert('[Error!] Kode pengiriman telah dikonfirmasi.
106         Id pengiriman:
107         '+results.rows.item(0).id_pengiriman);
108     } else {
109         alert('[Error!] Kode pengiriman tidak sesuai.');
```

```
110 }
111 }
112
113 function updateDB(tx){
114     tx.executeSql("UPDATE data_pengiriman SET status =
115         'Terkirim' WHERE id_pengiriman =
116         '"+idPengirimanUpd+"'");
117 }
118
```

```
119 function sinkronDB(callback){
120     db.transaction(function (tx) {
121         sinkronQuery(tx, callback);
122     }, errorCB);
123 }
124
```

```
125 function sinkronQuery(tx, callback){
126     tx.executeSql("SELECT * FROM data_pengiriman WHERE
127         Status = 'Terkirim'", [], function (tx, result) {
128         successSelect(tx, result, callback);
129     });
130 }
131
```

```

132 function successSelect(tx,result,callback){
133     var len = result.rows.length;
134     dataSync = [];
135     for (var i=0; i<len; i++){
136         dataSync[i] = result.rows.item(i).id_pengiriman;
137         console.log('dataSync['+i+' ] = '+dataSync[i]);
138     }
139     console.log(dataSync);
140     callback(dataSync);
141 }
142
143 function delConfirmed(){
144     db.transaction(delQuery,errorCB);
145 }
146
147 function delQuery(tx){
148     tx.executeSql("DELETE FROM data_pengiriman WHERE
149         status = 'Terkirim'");
150 }
151
152 function sinkronUpd(){
153     db.transaction(updQuery,errorCB);
154 }
155
156 function updQuery(tx){
157     var len = dataBaru.length;
158     console.log('len = '+len);
159     for (var i = 0; i < len; i++) {
160         tx.executeSql("INSERT OR IGNORE INTO data_pengiriman
161             VALUES ('"+idBaru[i]+"', '"+dataBaru[i]+"', 'Belum
162             Terkirim')");
163         console.log('Inserted new kode '+dataBaru[i]+ ' id:
164             ' +idBaru[i]+'..');
165     }
166 }

```

Kode 5.8 database.js

Pada kode 5.8 terdapat fungsi-fungsi yang berisikan *query-query database* SQLite. Fungsi-fungsi pada `database.js` ini yang disediakan untuk setiap proses yang berhubungan dengan *database* seperti memasukkan data pengiriman yang diterima dari *server*, konfirmasi kode pengiriman, dan mengambil id pengiriman yang telah dikonfirmasi untuk dikirim pada *server*. Contoh fungsi `confirmaCode` pada baris 74 sampai 76 merupakan fungsi untuk melakukan konfirmasi kode pengiriman yang berisikan API untuk melakukan transaksi *query* SQLite dengan fungsi `confirmQuery` yang terdapat pada baris 78 sampai 82, berisikan *query* untuk mencari data kode pengiriman yang sama dengan kode pengiriman yang dimasukkan, dan `errorCB` adalah fungsi untuk *error handling* ketika gagal mengeksekusi *query*. Fungsi `confirmaCode` ini dipanggil pada *script* `barcodescanner.js` dan `konfirmasi.js` untuk melakukan proses konfirmasi kode pengiriman dengan *database*.

Selain *script* yang telah disebut ada satu *script* lagi yang digunakan untuk proses mengubah kode pengiriman yang akan dikonfirmasi menjadi kode *hash* dengan metode MD5, yaitu *script* `md5.js`. *Script* ini adalah *script library* yang dibuat secara *open source* dengan nama *project* CryptoJS. *Script* ini terdapat pada lampiran 6.



BAB VI PENGUJIAN

Bab ini berisikan tentang penjelasan pengujian-pengujian yang dilakukan pada sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak. Setelah semua proses implementasi selesai, ditandai dengan selesainya pembuatan sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak akan dilakukan pengujian-pengujian berdasarkan kebutuhan fungsional dan non fungsional yang ada pada penjelasan dalam perancangan.

6.1. Pengujian

Pengujian dilakukan untuk mengetahui apakah sistem telah berjalan dengan baik dan sesuai dengan perancangan yang telah dibuat sebelumnya. Dengan berjalannya sistem sesuai dengan perancangan yang dibuat menandakan sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak telah memenuhi kebutuhan baik fungsional maupun non fungsional. Pengujian dari sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak pada penelitian ini adalah pengujian validasi

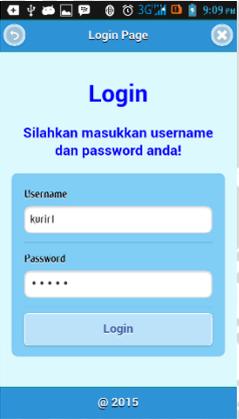
Pengujian validasi adalah pengujian yang dilakukan untuk mengetahui apakah sistem yang telah dibuat apakah bisa berjalan dengan benar dan telah memenuhi kebutuhan fungsional yang telah dijelaskan pada *use case*. Pengujian ini menggunakan metode pengujian *black-box testing*, karena tidak diperlukan konsentrasi terhadap alur jalannya algoritma program dan lebih ditekankan untuk menemukan kecocokan antara kinerja sistem dengan daftar kebutuhan (*use case*). Pengujian validasi dilakukan pada aplikasi klien. Pengujian validasi yang dilakukan pada sistem dari penelitian ini antara lain Pengujian *user login*, Pengujian konfirmasi kode pengiriman, dan Pengujian *update* data pengiriman barang.

6.1.1. Pengujian *user login*

Pengujian *user login* adalah pengujian yang dilakukan untuk mengetahui fungsi *login* untuk memenuhi kebutuhan fungsional yang telah disebutkan pada *use case* melakukan *login*. Pengujian ini dilakukan berdasarkan dengan skenario *use case* melakukan *login*. Untuk hasil penjelasan lengkap pengujian *user login* ditunjukkan pada tabel 6.1.

Tabel 6.1 Hasil pengujian *user login*

Nama Kasus Uji	Kasus melakukan <i>login</i>
Tujuan Pengujian	Pengujian dilakukan untuk memastikan aplikasi klien bisa <i>login</i> menggunakan <i>username</i> dan <i>password</i> milik kurir.
Data Masukan	<ol style="list-style-type: none"> 1. <i>Username</i>: kurir1 2. <i>Password</i>: abcde
Prosedur Uji	<ol style="list-style-type: none"> 1. Membuka aplikasi QRDelivery 2. Menuju halaman <i>login</i> dengan menekan tombol <i>Login</i> pada halaman awal 3. Memasukkan <i>username</i> dan <i>password</i> pada <i>form login</i> dan tekan tombol <i>Login</i>. 4. Aplikasi menampilkan hasil autentikasi login
Hasil Yang Diharapkan	Kurir berhasil melakukan <i>login</i> pada aplikasi klien
Hasil	<ol style="list-style-type: none"> 1. Membuka aplikasi QRDelivery  <ol style="list-style-type: none"> 2. Tekan tombol <i>login</i> pada halaman awal dari aplikasi

	 <p>Selamat datang di QR Delivery!</p> <p>Untuk melakukan konfirmasi pengiriman barang harap login terlebih dahulu menggunakan username dan password anda!</p> <p>Login</p> <p>@ 2015</p> <p>3. Memasukkan <i>username</i> dan <i>password</i> pada form login dan tekan tombol Login.</p>  <p>Login Page</p> <p>Login</p> <p>Silahkan masukkan username dan password anda!</p> <p>Username kurir1</p> <p>Password *****</p> <p>Login</p> <p>@ 2015</p> <p>4. Hasil autentikasi berhasil membuka halaman konfirmasi</p>  <p>QR Delivery - U...</p> <p>Selamat Datang kurir1</p> <p>Scan QR Code</p> <p>Masukkan Kode</p> <p>Update Data Pengiriman</p> <p>@ 2015</p>
<p>Status Validasi</p>	<p>Valid</p>
<p>Alur alternatif: Jika tidak dapat terhubung dengan server</p>	
<p>Tujuan Pengujian</p>	<p>Pengujian dilakukan untuk memastikan bahwa aplikasi memberi peringatan jika tidak dapat terhubung dengan server</p>



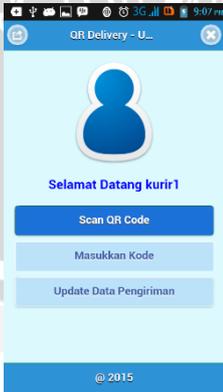
Data Masukan	<ol style="list-style-type: none"> 1. <i>Username</i>: kurir1 2. <i>Password</i>: abcde
Prosedur Uji	<ol style="list-style-type: none"> 1. Membuka aplikasi QRDelivery 2. Menuju halaman <i>login</i> dengan menekan tombol <i>Login</i> pada halaman awal 3. Memasukkan <i>username</i> dan <i>password</i> pada <i>form login</i> dan tekan tombol <i>Login</i>. 4. Aplikasi menampilkan hasil autentikasi login
Hasil Yang Diharapkan	<p>Apikasi memberi peringatan koneksi <i>error</i> karena tidak dapat terhubung dengan <i>server</i></p>
Hasil	<ol style="list-style-type: none"> 1. Memasukkan <i>username</i> dan <i>password</i> pada <i>form login</i> dan tekan tombol <i>Login</i>  <ol style="list-style-type: none"> 2. Menampilkan pesan kesalahan 
Status Validasi	Valid
Alur alternatif: Jika tidak <i>username</i> atau <i>password</i> salah	

Tujuan Pengujian	Pengujian dilakukan untuk memastikan bahwa aplikasi memberi peringatan jika <i>username</i> atau <i>password</i> salah
Data Masukan	<ol style="list-style-type: none"> 1. <i>Username</i>: kurir1 2. <i>Password</i>: 12345
Prosedur Uji	<ol style="list-style-type: none"> 1. Membuka aplikasi QRDelivery 2. Menuju halaman <i>login</i> dengan menekan tombol <i>Login</i> pada halaman awal 3. Memasukkan <i>username</i> dan <i>password</i> pada <i>form login</i> dan tekan tombol <i>Login</i>. 4. Aplikasi menampilkan hasil autentikasi login
Hasil Yang Diharapkan	Apikasi memberi peringatan <i>login</i> gagal karena <i>username</i> atau <i>password</i> tidak sesuai.
Hasil	<ol style="list-style-type: none"> 1. Memasukkan <i>username</i> dan <i>password</i> pada <i>form login</i> dan tekan tombol <i>Login</i> <div data-bbox="705 1003 944 1424" data-label="Image"> </div> 2. Menampilkan pesan kesalahan <div data-bbox="705 1491 944 1912" data-label="Image"> </div>
Status Validasi	Valid

6.1.2. Pengujian konfirmasi kode pengiriman barang

Pengujian konfirmasi kode pengiriman barang adalah pengujian sistem yang dilakukan untuk mengetahui hasil dari implementasi sistem dalam memenuhi kebutuhan konfirmasi pengiriman barang yang disebutkan pada *use case* mengkonfirmasi pengiriman barang. Pengujian dilakukan berdasarkan skenario *use case* mengkonfirmasi pengiriman barang. Hasil pengujian konfirmasi kode pengiriman barang ditunjukkan pada tabel 6.2.

Tabel 6.2 Hasil pengujian konfirmasi kode pengiriman barang

Nama Kasus Uji	Kasus melakukan konfirmasi kode pengiriman barang menggunakan QR Code
Tujuan Pengujian	Pengujian dilakukan untuk memastikan aplikasi klien bisa mengkonfirmasi kode pengiriman milik penerima barang dengan data pengiriman pada <i>database</i> dengan menggunakan QR Code.
Data Masukan	QR Code
Prosedur Uji	<ol style="list-style-type: none"> 1. Kurir membuka kamera untuk memindai QR Code dengan menekan tombol "Scan QR Code" pada halaman konfirmasi 2. Pindai QR Code dengan kamera yang telah dibuka sampai aplikasi menangkap isi dari QR Code. 3. Aplikasi klien menampilkan hasil konfirmasi berhasil
Hasil Yang Diharapkan	Kurir berhasil mengkonfirmasi kode pengiriman milik penerima barang menggunakan cara <i>Scan QR Code</i> .
Hasil	<ol style="list-style-type: none"> 1. Membuka kamera untuk memindai QR Code dengan menekan tombol "Scan QR Code" pada halaman konfirmasi  <ol style="list-style-type: none"> 2. Memindai QR Code dengan kamera sampai mendapatkan isi dari QR Code

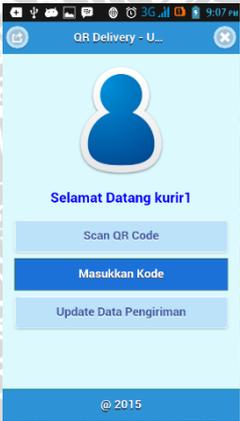


3. Menampilkan hasil konfirmasi berhasil



Status Validasi	Valid
Alur alternatif: Jika QR Code tidak dapat dibaca oleh kamera dan menggunakan cara manual	
Tujuan Pengujian	Pengujian dilakukan untuk mengetahui ketika aplikasi tidak dapat membaca QR Code saat proses pindai.
Data Masukan	<ol style="list-style-type: none"> 1. QR Code (Menggunakan cara <i>Scan QR Code</i>) 2. Kode pengiriman (Menggunakan cara manual)
Prosedur Uji	<ol style="list-style-type: none"> 1. Kurir memilih menu <i>Scan QR Code</i>. 2. Kamera memindai QR Code dan tidak memberi respon dalam waktu lama karena tidak dapat membaca QR Code. 3. Kembali ke halaman Konfirmasi dan memilih menu Masukkan kode untuk memasukkan kode pengiriman secara langsung. 4. Memasukkan kode pengiriman dari penerima barang dan tekan tombol Konfirmasi. 5. Aplikasi menampilkan hasil konfirmasi berhasil.



<p>Hasil Yang Diharapkan</p>	<p>Apikasi memberikan peringatan konfirmasi berhasil dengan cara memasukkan kode pengiriman secara manual.</p>
<p>Hasil</p>	<p>1. Memindai QR Code dan tidak mendapat respon setelah beberapa saat.</p>  <p>2. Kembali ke halaman Konfirmasi dan memilih menu Masukkan kode</p> 
<p>Status Validasi</p>	<p>Valid</p>
<p>Alur alternatif: Jika kode pengiriman tidak sesuai atau data kurir tidak sesuai</p>	
<p>Tujuan Pengujian</p>	<p>Pengujian dilakukan untuk memastikan bahwa aplikasi memberi peringatan jika kode pengiriman tidak sesuai atau data kurir tidak sesuai.</p>
<p>Data Masukan</p>	<p>QR Code</p>
<p>Prosedur Uji</p>	<p>1. Kurir memilih menu <i>Scan QR Code</i>. 2. Aplikasi memindai QR Code.</p>



	<p>3. Aplikasi menampilkan hasil konfirmasi berupa peringatan <i>error</i> kode pengiriman tidak sesuai.</p>
Hasil Yang Diharapkan	<p>Apikasi memberi peringatan <i>error</i> kode pengiriman tidak sesuai.</p>
Hasil	<p>1. Memilih menu <i>Scan QR Code</i></p>  <p>2. Aplikasi memindai QR Code</p>  <p>3. Aplikasi menampilkan hasil <i>error</i> kode pengiriman tidak sesuai.</p> 
Status Validasi	Valid

6.1.3. Pengujian *update* data pengiriman barang

Pengujian *update* data pengiriman barang dilakukan untuk mengetahui fungsi *update database* pengiriman barang yang sesuai kebutuhan fungsional mengirim hasil konfirmasi pengiriman barang yang ditunjukkan dalam *use case* diagram. Hasil dari pengujian *update* data pengiriman barang ditunjukkan pada tabel 6.3.

Tabel 6.3 Hasil pengujian *update* data pengiriman barang

Nama Kasus Uji	Kasus memperbarui data pengiriman pada <i>server</i> dan aplikasi
Tujuan Pengujian	Pengujian dilakukan untuk memastikan aplikasi bisa mengirim data hasil pengiriman ke <i>server</i> dan juga aplikasi klien memperbarui data pengiriman sesuai dengan data yang diterima dari <i>server</i> .
Data Masukan	Tidak ada
Prosedur Uji	<ol style="list-style-type: none"> 1. Melihat data pengiriman pada <i>server</i> 2. Kurir memilih menu <i>Update Data Pengiriman</i> 3. Aplikasi klien menampilkan hasil <i>update</i> data pengiriman berhasil. 4. Melihat hasil <i>update</i> pada <i>server</i>
Hasil Yang Diharapkan	Aplikasi berhasil meng- <i>update database</i> pengiriman pada <i>server</i> dan pada aplikasi klien.
Hasil	<ol style="list-style-type: none"> 1. Melihat data pengiriman pada <i>server</i> dengan membuka alamat qrkonfirmasi.16mb.com/pengiriman-list.php

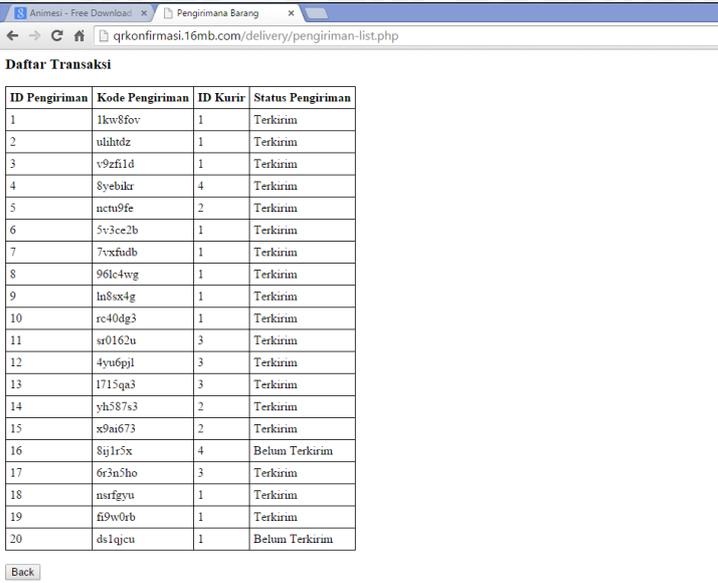
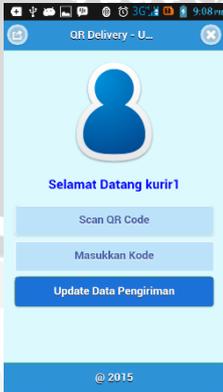
ID Pengiriman	Kode Pengiriman	ID Kurir	Status Pengiriman
1	1kw8fov	1	Terkirim
2	uilihdz	1	Terkirim
3	v9zfi1d	1	Terkirim
4	8yebikr	4	Terkirim
5	nctu9fe	2	Terkirim
6	5v3ce2b	1	Terkirim
7	7xxtfdb	1	Terkirim
8	96lc4wg	1	Terkirim
9	ln8sx4g	1	Terkirim
10	rc40dg3	1	Terkirim
11	sr0162u	3	Terkirim
12	4yu6pjl	3	Terkirim
13	l715qa3	3	Terkirim
14	yh587s3	2	Terkirim
15	x9ai673	2	Terkirim
16	8ij1r5x	4	Belum Terkirim
17	6r3n3ho	3	Belum Terkirim
18	nsrfgyu	1	Belum Terkirim
19	fi9w0rb	1	Belum Terkirim

2. Kurir memilih menu *Update* data pengiriman

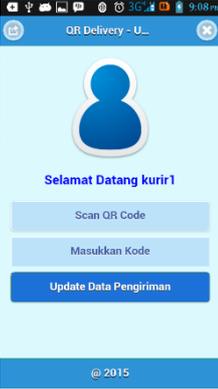
3. Aplikasi menampilkan peringatan *update* data pengiriman berhasil.

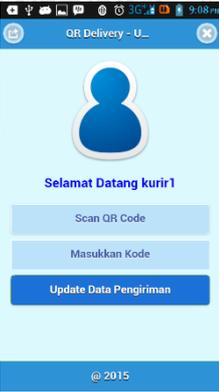
4. Melihat hasil *update* pada server



	 <table border="1"> <thead> <tr> <th>ID Pengiriman</th> <th>Kode Pengiriman</th> <th>ID Kurir</th> <th>Status Pengiriman</th> </tr> </thead> <tbody> <tr><td>1</td><td>1kw8fov</td><td>1</td><td>Terkirim</td></tr> <tr><td>2</td><td>u1h1dz</td><td>1</td><td>Terkirim</td></tr> <tr><td>3</td><td>v9zf1d</td><td>1</td><td>Terkirim</td></tr> <tr><td>4</td><td>8yebikr</td><td>4</td><td>Terkirim</td></tr> <tr><td>5</td><td>nctu9fe</td><td>2</td><td>Terkirim</td></tr> <tr><td>6</td><td>5v3ce2b</td><td>1</td><td>Terkirim</td></tr> <tr><td>7</td><td>7xvfudb</td><td>1</td><td>Terkirim</td></tr> <tr><td>8</td><td>96lc4wg</td><td>1</td><td>Terkirim</td></tr> <tr><td>9</td><td>ln8sx4g</td><td>1</td><td>Terkirim</td></tr> <tr><td>10</td><td>rc40dg3</td><td>1</td><td>Terkirim</td></tr> <tr><td>11</td><td>sr0162u</td><td>3</td><td>Terkirim</td></tr> <tr><td>12</td><td>4yu6p1</td><td>3</td><td>Terkirim</td></tr> <tr><td>13</td><td>1715qa3</td><td>3</td><td>Terkirim</td></tr> <tr><td>14</td><td>yh587s3</td><td>2</td><td>Terkirim</td></tr> <tr><td>15</td><td>x9ai673</td><td>2</td><td>Terkirim</td></tr> <tr><td>16</td><td>8ij1r7x</td><td>4</td><td>Belum Terkirim</td></tr> <tr><td>17</td><td>6r3n5ho</td><td>3</td><td>Terkirim</td></tr> <tr><td>18</td><td>nsrfgyu</td><td>1</td><td>Terkirim</td></tr> <tr><td>19</td><td>fi9w0rb</td><td>1</td><td>Terkirim</td></tr> <tr><td>20</td><td>ds1qjeu</td><td>1</td><td>Belum Terkirim</td></tr> </tbody> </table>	ID Pengiriman	Kode Pengiriman	ID Kurir	Status Pengiriman	1	1kw8fov	1	Terkirim	2	u1h1dz	1	Terkirim	3	v9zf1d	1	Terkirim	4	8yebikr	4	Terkirim	5	nctu9fe	2	Terkirim	6	5v3ce2b	1	Terkirim	7	7xvfudb	1	Terkirim	8	96lc4wg	1	Terkirim	9	ln8sx4g	1	Terkirim	10	rc40dg3	1	Terkirim	11	sr0162u	3	Terkirim	12	4yu6p1	3	Terkirim	13	1715qa3	3	Terkirim	14	yh587s3	2	Terkirim	15	x9ai673	2	Terkirim	16	8ij1r7x	4	Belum Terkirim	17	6r3n5ho	3	Terkirim	18	nsrfgyu	1	Terkirim	19	fi9w0rb	1	Terkirim	20	ds1qjeu	1	Belum Terkirim
ID Pengiriman	Kode Pengiriman	ID Kurir	Status Pengiriman																																																																																		
1	1kw8fov	1	Terkirim																																																																																		
2	u1h1dz	1	Terkirim																																																																																		
3	v9zf1d	1	Terkirim																																																																																		
4	8yebikr	4	Terkirim																																																																																		
5	nctu9fe	2	Terkirim																																																																																		
6	5v3ce2b	1	Terkirim																																																																																		
7	7xvfudb	1	Terkirim																																																																																		
8	96lc4wg	1	Terkirim																																																																																		
9	ln8sx4g	1	Terkirim																																																																																		
10	rc40dg3	1	Terkirim																																																																																		
11	sr0162u	3	Terkirim																																																																																		
12	4yu6p1	3	Terkirim																																																																																		
13	1715qa3	3	Terkirim																																																																																		
14	yh587s3	2	Terkirim																																																																																		
15	x9ai673	2	Terkirim																																																																																		
16	8ij1r7x	4	Belum Terkirim																																																																																		
17	6r3n5ho	3	Terkirim																																																																																		
18	nsrfgyu	1	Terkirim																																																																																		
19	fi9w0rb	1	Terkirim																																																																																		
20	ds1qjeu	1	Belum Terkirim																																																																																		
<p>Status Validasi</p>	<p>Valid</p>																																																																																				
<p align="center">Alur alternatif: Jika belum ada pengiriman yang terkirim</p>																																																																																					
<p>Tujuan Pengujian</p>	<p>Pengujian dilakukan untuk memastikan aplikasi bisa menampilkan peringatan jika belum ada <i>item</i> yang terkirim ketika meng-<i>update</i> data pengiriman.</p>																																																																																				
<p>Data Masukan</p>	<p>Tidak ada</p>																																																																																				
<p>Prosedur Uji</p>	<ol style="list-style-type: none"> 1. Kurir memilih menu <i>Update Data Pengiriman</i> 2. Aplikasi menampilkan pesan hasil <i>update</i> berupa belum ada barang yang terkirim. 																																																																																				
<p>Hasil Yang Diharapkan</p>	<p>Aplikasi berhasil menampilkan pesan bahwa belum ada barang yang terkirim.</p>																																																																																				
<p>Hasil</p>	<ol style="list-style-type: none"> 1. Kurir memilih menu <i>Update data pengiriman</i>  <ol style="list-style-type: none"> 2. Aplikasi menampilkan peringatan belum ada barang yang terkirim 																																																																																				



	
<p>Hasil validasi</p>	<p>Valid</p>
<p align="center">Alur alternatif: Jika tidak ada data pengiriman barang yang baru dari server</p>	
<p>Tujuan Pengujian</p>	<p>Pengujian dilakukan untuk memastikan aplikasi bisa menampilkan peringatan jika tidak ada data pengiriman yang baru dari server.</p>
<p>Data Masukan</p>	<p>Tidak ada</p>
<p>Prosedur Uji</p>	<ol style="list-style-type: none"> 1. Kurir memilih menu <i>Update Data Pengiriman</i> 2. Aplikasi menampilkan pesan tidak ada data pengiriman yang baru dari server
<p>Hasil Yang Diharapkan</p>	<p>Aplikasi berhasil menampilkan pesan bahwa tidak ada data pengiriman yang baru dari server</p>
<p>Hasil</p>	<ol style="list-style-type: none"> 1. Kurir memilih menu <i>Update data pengiriman</i>  <ol style="list-style-type: none"> 2. Aplikasi menampilkan pesan tidak ada data pengiriman baru

	
<p>Hasil validasi</p>	<p>Valid</p>
<p>Alur alternatif: Jika tidak dapat terhubung dengan server</p>	
<p>Tujuan Pengujian</p>	<p>Pengujian dilakukan untuk memastikan aplikasi bisa menampilkan peringatan jika aplikasi tidak dapat terhubung dengan server ketika melakukan update data pengiriman</p>
<p>Data Masukan</p>	<p>Tidak ada</p>
<p>Prosedur Uji</p>	<ol style="list-style-type: none"> 1. Kurir memilih menu <i>Update Data Pengiriman</i> 2. Aplikasi menampilkan peringatan tidak dapat mengirim data pengiriman karena tidak terhubung dengan server
<p>Hasil Yang Diharapkan</p>	<p>Aplikasi berhasil menampilkan peringatan bahwa tidak dapat terhubung dengan server.</p>
<p>Hasil</p>	<ol style="list-style-type: none"> 1. Kurir memilih menu <i>Update data pengiriman</i>  <ol style="list-style-type: none"> 2. Aplikasi menampilkan peringatan bahwa tidak dapat mengirim data karena tidak dapat terhubung dengan server.

	
<p>Hasil validasi</p>	<p>Valid</p>

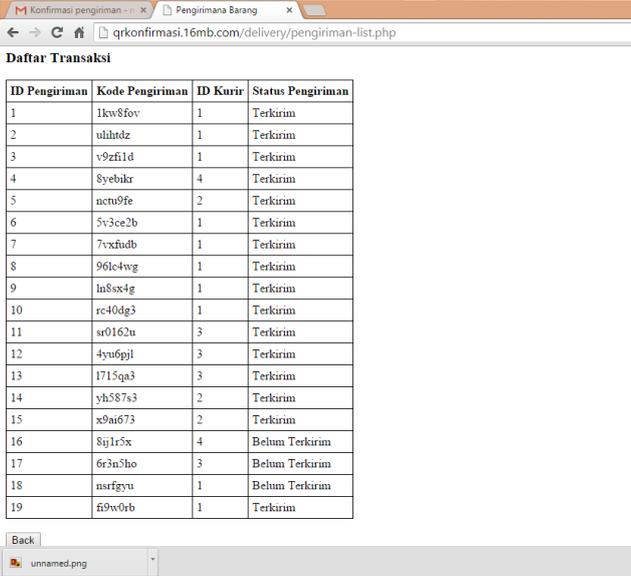
Selain pengujian-pengujian fungsionalitas dari sistem yang berdasarkan kebutuhan fungsional dilakukan juga pengujian validasi dari kebutuhan non fungsional, yaitu keamanan kode pengiriman barang. Pengujian ini dilakukan untuk memastikan sistem dilengkapi dengan metode keamanan berupa pengenkripsian kode pengiriman barang yang ada pada sisi *web API* dan juga aplikasi klien. Metode enkripsi pada sistem ini adalah menggunakan Hash dengan metode MD5. Metode pengujian yang digunakan adalah melihat apakah kode pengiriman berhasil di-*hash* atau tidak dengan cara memonitor jalan aplikasi menggunakan “adb logcat” pada *Console / Command Prompt*. Sebelumnya aplikasi telah dibuat agar bisa melakukan log untuk setiap data yang akan dimasukkan pada *database* dalam aplikasi klien disaat proses berhasil melakukan *login* dan *update* data pengiriman. Kemudian untuk melihat data mentah atau data sebelum dikirim ke *database* aplikasi dengan cara melihat isi dari *database server*.

6.1.4. Pengujian enkripsi data ketika berhasil login

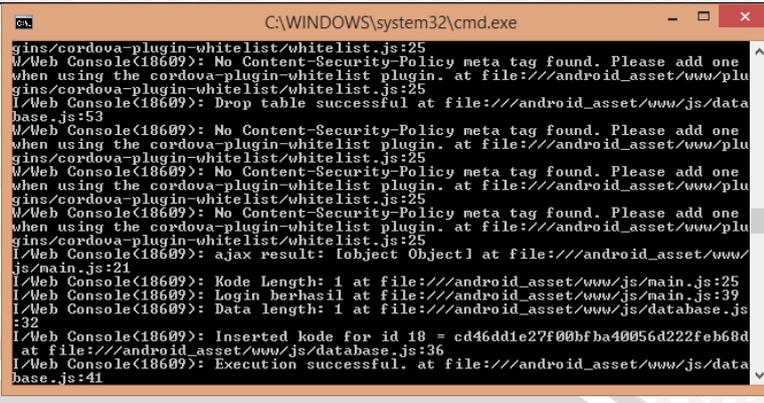
Pada saat aplikasi berhasil *login* menggunakan *username* dan *password* yang dimasukkan kurir, aplikasi klien akan menerima daftar kode pengiriman barang milik kurir yang bertanggung jawab. Pada saat itulah akan dipastikan apakah kode pengiriman barang telah berhasil dienkripsi atau tidak. Hasil dari pengujian enkripsi pada ketika berhasil *login* ditunjukkan pada tabel 6.4.

Tabel 6. 4 Hasil pengujian enkripsi ketika berhasil *login*

<p>Nama Kasus Uji</p>	<p>Kasus melakukan login</p>
<p>Tujuan Pengujian</p>	<p>Pengujian dilakukan untuk memastikan aplikasi klien menerima kode pengiriman barang yang telah terenkripsi dari <i>server</i> ketika berhasil <i>login</i>.</p>
<p>Data Masukan</p>	<p>Tidak ada</p>
<p>Prosedur Uji</p>	<p>1. Melihat data pengiriman pada <i>server</i></p>

	<p>2. Melihat <i>console log</i> pada <i>Command Prompt</i> dengan mengetikkan perintah <code>adb logcat findstr :/C"Web Console"</code></p> <p>3. Melakukan login pada aplikasi klien.</p>																																																																																
<p>Hasil Yang Diharapkan</p>	<p>Aplikasi klien menerima kode pengiriman barang yang telah terenkripsi dari <i>server</i> dan memasukkannya ke dalam <i>database</i> aplikasi klien ketika berhasil <i>login</i>.</p>																																																																																
<p>Hasil</p>	<p>1. Melihat data pengiriman pada <i>server</i></p>  <table border="1" data-bbox="730 707 1082 1149"> <thead> <tr> <th>ID Pengiriman</th> <th>Kode Pengiriman</th> <th>ID Kurir</th> <th>Status Pengiriman</th> </tr> </thead> <tbody> <tr><td>1</td><td>1kw8fov</td><td>1</td><td>Terkirim</td></tr> <tr><td>2</td><td>ulihtdz</td><td>1</td><td>Terkirim</td></tr> <tr><td>3</td><td>v9zfi1d</td><td>1</td><td>Terkirim</td></tr> <tr><td>4</td><td>8yebikr</td><td>4</td><td>Terkirim</td></tr> <tr><td>5</td><td>actn9fe</td><td>2</td><td>Terkirim</td></tr> <tr><td>6</td><td>5v3ce2b</td><td>1</td><td>Terkirim</td></tr> <tr><td>7</td><td>7vxfudb</td><td>1</td><td>Terkirim</td></tr> <tr><td>8</td><td>96ic4wg</td><td>1</td><td>Terkirim</td></tr> <tr><td>9</td><td>ln8sx4g</td><td>1</td><td>Terkirim</td></tr> <tr><td>10</td><td>rc40dg3</td><td>1</td><td>Terkirim</td></tr> <tr><td>11</td><td>sr0162u</td><td>3</td><td>Terkirim</td></tr> <tr><td>12</td><td>4yu6ppl</td><td>3</td><td>Terkirim</td></tr> <tr><td>13</td><td>l715qn3</td><td>3</td><td>Terkirim</td></tr> <tr><td>14</td><td>yh587s3</td><td>2</td><td>Terkirim</td></tr> <tr><td>15</td><td>x9ai673</td><td>2</td><td>Terkirim</td></tr> <tr><td>16</td><td>8j1r5x</td><td>4</td><td>Belum Terkirim</td></tr> <tr><td>17</td><td>6r3n3ho</td><td>3</td><td>Belum Terkirim</td></tr> <tr><td>18</td><td>nsrfgyu</td><td>1</td><td>Belum Terkirim</td></tr> <tr><td>19</td><td>fi9w0rb</td><td>1</td><td>Terkirim</td></tr> </tbody> </table> <p>2. Melakukan <i>login</i> menggunakan ID kurir 1.</p>  <p>3. Melihat hasil monitor dari <i>command prompt</i> ditunjukkan oleh log "Inserted kode for id 18 ..."</p>	ID Pengiriman	Kode Pengiriman	ID Kurir	Status Pengiriman	1	1kw8fov	1	Terkirim	2	ulihtdz	1	Terkirim	3	v9zfi1d	1	Terkirim	4	8yebikr	4	Terkirim	5	actn9fe	2	Terkirim	6	5v3ce2b	1	Terkirim	7	7vxfudb	1	Terkirim	8	96ic4wg	1	Terkirim	9	ln8sx4g	1	Terkirim	10	rc40dg3	1	Terkirim	11	sr0162u	3	Terkirim	12	4yu6ppl	3	Terkirim	13	l715qn3	3	Terkirim	14	yh587s3	2	Terkirim	15	x9ai673	2	Terkirim	16	8j1r5x	4	Belum Terkirim	17	6r3n3ho	3	Belum Terkirim	18	nsrfgyu	1	Belum Terkirim	19	fi9w0rb	1	Terkirim
ID Pengiriman	Kode Pengiriman	ID Kurir	Status Pengiriman																																																																														
1	1kw8fov	1	Terkirim																																																																														
2	ulihtdz	1	Terkirim																																																																														
3	v9zfi1d	1	Terkirim																																																																														
4	8yebikr	4	Terkirim																																																																														
5	actn9fe	2	Terkirim																																																																														
6	5v3ce2b	1	Terkirim																																																																														
7	7vxfudb	1	Terkirim																																																																														
8	96ic4wg	1	Terkirim																																																																														
9	ln8sx4g	1	Terkirim																																																																														
10	rc40dg3	1	Terkirim																																																																														
11	sr0162u	3	Terkirim																																																																														
12	4yu6ppl	3	Terkirim																																																																														
13	l715qn3	3	Terkirim																																																																														
14	yh587s3	2	Terkirim																																																																														
15	x9ai673	2	Terkirim																																																																														
16	8j1r5x	4	Belum Terkirim																																																																														
17	6r3n3ho	3	Belum Terkirim																																																																														
18	nsrfgyu	1	Belum Terkirim																																																																														
19	fi9w0rb	1	Terkirim																																																																														



	 <pre> C:\WINDOWS\system32\cmd.exe jins/cordova-plugin-whitelist/whitelist.js:25 I/Web Console(18609): No Content-Security-Policy meta tag found. Please add one when using the cordova-plugin-whitelist plugin. at file:///android_asset/www/plu jins/cordova-plugin-whitelist/whitelist.js:25 I/Web Console(18609): Drop table successful at file:///android_asset/www/js/data base.js:53 I/Web Console(18609): No Content-Security-Policy meta tag found. Please add one when using the cordova-plugin-whitelist plugin. at file:///android_asset/www/plu jins/cordova-plugin-whitelist/whitelist.js:25 I/Web Console(18609): No Content-Security-Policy meta tag found. Please add one when using the cordova-plugin-whitelist plugin. at file:///android_asset/www/plu jins/cordova-plugin-whitelist/whitelist.js:25 I/Web Console(18609): No Content-Security-Policy meta tag found. Please add one when using the cordova-plugin-whitelist plugin. at file:///android_asset/www/plu jins/cordova-plugin-whitelist/whitelist.js:25 I/Web Console(18609): ajax result: {object Object} at file:///android_asset/www/ js/main.js:21 I/Web Console(18609): Kode Length: 1 at file:///android_asset/www/js/main.js:25 I/Web Console(18609): Login berhasil at file:///android_asset/www/js/main.js:39 I/Web Console(18609): Data length: 1 at file:///android_asset/www/js/database.js :32 I/Web Console(18609): Inserted kode for id 18 = cd46dd1e27f00bfa40056d222feb60d at file:///android_asset/www/js/database.js:36 I/Web Console(18609): Execution successful. at file:///android_asset/www/js/data base.js:41 </pre>
<p>Status Validasi</p>	<p>Valid</p>

6.1.5. Pengujian enkripsi data ketika *update* data pengiriman barang

Pengujian enkripsi data kode pengiriman yang kedua adalah ketika melakukan *update* data pengiriman barang. Pada saat ini aplikasi pengiriman ID pengiriman yang telah berhasil dikonfirmasi dan menerima data pengiriman barang yang baru dari *server*. Sama dengan ketika berhasil *login* data kode pengiriman barang yang diterima oleh aplikasi dalam kondisi terenkripsi. Adapun hasil dari pengujian enkripsi data ketika *update* data pengiriman barang ditunjukkan pada tabel 6.5.

Tabel 6. 5 Hasil pengujian enkripsi data ketika *update* data pengiriman barang.

<p>Nama Kasus Uji</p>	<p>Kasus melakukan login</p>
<p>Tujuan Pengujian</p>	<p>Pengujian dilakukan untuk memastikan aplikasi klien menerima kode pengiriman barang yang telah terenkripsi dari <i>server</i> ketika melakukan <i>update</i> data pengiriman.</p>
<p>Data Masukan</p>	<p>Tidak ada</p>
<p>Prosedur Uji</p>	<ol style="list-style-type: none"> 1. Melihat data pengiriman pada <i>server</i> 2. Melihat <i>console log</i> pada <i>Command Prompt</i> dengan mengetikkan perintah <code>adb logcat findstr :/C"Web Console"</code> 3. Melakukan <i>update</i> data pengiriman dalam aplikasi klien.
<p>Hasil Yang Diharapkan</p>	<p>Aplikasi klien menerima kode pengiriman barang yang telah terenkripsi dari <i>server</i> dan memasukkannya ke dalam <i>database</i> aplikasi klien ketika berhasil melakukan <i>update</i> data pengiriman.</p>
<p>Hasil</p>	<ol style="list-style-type: none"> 1. Melihat data pengiriman pada <i>server</i>



ID Pengiriman	Kode Pengiriman	ID Kurir	Status Pengiriman
1	1kw8fov	1	Terkirim
2	ulhtdz	1	Terkirim
3	v9zfi1d	1	Terkirim
4	8yebukr	4	Terkirim
5	nctu9fe	2	Terkirim
6	5v3ce2b	1	Terkirim
7	7vxftdb	1	Terkirim
8	96lc4wg	1	Terkirim
9	lu8sx4g	1	Terkirim
10	rc40dg3	1	Terkirim
11	sr0162u	3	Terkirim
12	4yu6ppi	3	Terkirim
13	l715qa3	3	Terkirim
14	yh587s3	2	Terkirim
15	x9n673	2	Terkirim
16	8y1r5x	4	Belum Terkirim
17	6r3n3ho	3	Terkirim
18	nsrfgyw	1	Belum Terkirim
19	fi9w0rb	1	Terkirim
20	ds1qjeu	1	Belum Terkirim

2. Melakukan *update* data pengiriman barang setelah melakukan konfirmasi pengiriman barang.



3. Melihat hasil monitor dari *command prompt* ditunjukkan oleh log "Inserted new kode ... id: 20"

```

C:\WINDOWS\system32\cmd.exe
I/Web Console(21789): Inserted kode for id 20 = 82599911b279de49cb7bc11210126f79
at file:///android_asset/www/js/database.js:36
I/Web Console(21789): Execution successful. at file:///android_asset/www/js/data
base.js:41
W/Web Console(21789): No Content-Security-Policy meta tag found. Please add one
when using the cordova-plugin-whitelist plugin. at file:///android_asset/www/plu
gins/cordova-plugin-whitelist/whitelist.js:25
W/Web Console(21789): No Content-Security-Policy meta tag found. Please add one
when using the cordova-plugin-whitelist plugin. at file:///android_asset/www/plu
gins/cordova-plugin-whitelist/whitelist.js:25
W/Web Console(21789): No Content-Security-Policy meta tag found. Please add one
when using the cordova-plugin-whitelist plugin. at file:///android_asset/www/plu
gins/cordova-plugin-whitelist/whitelist.js:25
I/Web Console(21789): dataSync[0] = 18 at file:///android_asset/www/js/database.
js:114
I/Web Console(21789): 18 at file:///android_asset/www/js/database.js:116
I/Web Console(21789): datasync length = 1 at file:///android_asset/www/js/sinkro
nDB.js:6
I/Web Console(21789): dataSinkronItem length = 4 at file:///android_asset/www/js
/sinkronDB.js:7
I/Web Console(21789): databaru length=1 at file:///android_asset/www/js/sinkron
DB.js:23
I/Web Console(21789): len = 1 at file:///android_asset/www/js/database.js:134
I/Web Console(21789): Inserted new kode 82599911b279de49cb7bc11210126f79 id: 20.
at file:///android_asset/www/js/database.js:137
    
```

Status Validasi

Valid

6.2. Analisis Pengujian

Analisis pengujian merupakan proses untuk mendapat kesimpulan dari hasil pengujian sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak yang telah dilakukan. Analisis pengujian dilakukan untuk mengetahui hasil pengujian terhadap permasalahan yang telah disebutkan pada rumusan masalah dalam pendahuluan.

Berdasarkan hasil pengujian validasi yang telah dilakukan diketahui semua fungsi dalam sistem telah berjalan dengan benar dengan hasil validitas valid. Kemudian dari hasil pengujian enkripsi kode pengiriman pada sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak untuk menjamin barang sampai pada penerima sebenarnya diperlukan enkripsi pada kode pengiriman barang yang dikirim oleh *server* pada aplikasi untuk memastikan kode pengiriman tidak dapat dibaca oleh orang lain diketahui bahwa kode pengiriman telah berhasil dienkrpsi sehingga kode pengiriman aman dari pembacaan penyaringan data transfer *internet*.



BAB VII

KESIMPULAN DAN SARAN

7.1. Kesimpulan

Berdasarkan hasil analisis kebutuhan, perancangan, implementasi, dan pengujian yang dilakukan didapatkan kesimpulan sebagai berikut:

1. Berdasarkan hasil analisis kebutuhan dan perancangan untuk masalah yang disebutkan pada rumusan masalah, yaitu bagaimana membangun sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak yang dapat menjamin barang sampai pada penerima yang tepat didapatkan beberapa solusi dalam membangun sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak, yaitu:
 - a. QR Code yang berisikan kode pengiriman barang pembelian dijadikan penyimpanan kode pengiriman barang dan identitas yang menunjukkan penerima barang yang sebenarnya pada kurir.
 - b. Kode konfirmasi yang didapatkan dari hasil *scan* QR Code akan menunjukkan ID barang yang harus diserahkan pada penerima barang ketika QR Code berhasil di-*scan*. Dengan demikian barang tidak akan tertukar dengan barang kiriman yang lain.
 - c. Data kode-kode pengiriman yang ditransfer dari *server* menuju aplikasi klien melalui internet telah dienkripsi, sehingga kode pengiriman aman karena tidak dapat diketahui dengan menggunakan media penyaringan data.
2. Berdasarkan pengujian yang dilakukan fungsi dari sistem berjalan dengan baik dan sesuai dengan analisis kebutuhan fungsional sistem. Selain berdasarkan kebutuhan fungsional terdapat beberapa tambahan pada fungsi sistem seperti pada saat melakukan update yang hanya bisa dilakukan jika terdapat data pengiriman yang telah dikonfirmasi. Kemudian dari kebutuhan pengujian non fungsional, yaitu enkripsi data kode pengiriman juga dapat berjalan dengan benar. Setiap kode pengiriman yang dikirim dari *server* dan diterima oleh aplikasi klien berhasil terenkripsi dengan benar.

7.2. Saran

Saran yang dapat diberikan pada pengembangan sistem konfirmasi pengiriman barang menggunakan QR Code pada perangkat bergerak selanjutnya adalah:

1. Karena sistem ini adalah prototipe yang memiliki fungsi-fungsi dasar, untuk penelitian lebih lanjut pengembang dari sistem ini bisa menerapkan sistem pada sebuah implementasi lapangan secara nyata untuk mengetahui sistem yang dapat berfungsi secara akurat.

2. Untuk penelitian selanjutnya, pengembang bisa membuat *server* sesuai dengan sistem ini agar dapat berfungsi lebih baik.
3. Untuk penelitian selanjutnya, pengembang dari sistem ini bisa meneliti bagaimana sistem ini berjalan pada sistem pengiriman umum yang menangani pengiriman dari semua pihak tidak hanya dari jual beli online.



DAFTAR PUSTAKA

- [AAS-13] Ardy, Rhyca Putri, Atista, Riske, Sudrajat, Antonius Wahyu. 2013. Pemanfaatan Sms *Gateway* Dalam Pelayanan Informasi Aktifitas Siswa Pada TK Xaverius 5 Palembang.
- [AHW-12] Putra, Achmad Gazza, Ciptaningtyas, Henning Titi, Wibisono, Waskitho. 2012. Rancang Bangun Aplikasi *Android Virtual Shopping* Berbasis *QR Code* dan *Global Positioning System* untuk User Bergerak.
- [AKS-13] Arvi, Alifvia, Kridalukmana, Rinta, Satoto, Kodrat Iman. 2013. Perancangan Aplikasi Toko Online "Xo-Licious" Berbasis *Mobile Web* Pada Sistem Operasi *Android*.
- [AMS-06] Aminah, Siti. 2006. Sistem Pengiriman Barang Pada PT. Laju Merapi Dengan *Microsoft Visual Basic 6.0*.
- [AND-15] *Android Developer*. URL: <http://developer.android.com/index.html>. Diakses tanggal 25 Januari 2015.
- [CMW-14] Cahyanti, Apriliya Fitri, Murti, Antonius Bima, Wicaksono, Viko Basmalah. 2014. Analisis Perancangan Model Bisnis *Mobile E-Commerce*: Studi Kasus *Noraolshop.com*.
- [DAI-12] Daigneau, Robert. 2012. "*Service Design Patterns: Fundamental Design Solutions for SOAP/WSDL and RESTful Web Services*". Boston: *Pearson Education, Inc.*
- [GGH-11] Gardner, Lyza Danger, Grigsby, Jason. 2011. *Head First Mobile Web*. O'Reilly.
- [HNS-13] Handojo, Andreas, Noertjahyana, Agustinus, Sampurna, Osbert Tjitro. 2013. Pembuatan Aplikasi *Sales* berbasis *Mobile Device* menggunakan *Phonegap*.
- [HPS-12] Hardo, Putut Dewanto, Pulungan, Reza, Suprpto. 2012. Perancangan Sistem Pengawasan Pengiriman Barang Menggunakan *GPRS, GPS, Google Maps, Android, Dan RFID* Pada *Intelligent Warehouse Management System*.
- [JNE-11] P, Dian Kusuma A, Deviza, Linda, W, Imas Maulin A. 2011 *Marketing Plan for Service JNE*.
- [JQM-15] *JQuery Mobile*. URL: <http://jquerymobile.com> . Diakses tanggal 24 Juni 2015.
- [JWA-11] Jackson, Wallace. 2011. *Android Apps for Absolute Beginners*. *Springer Science+Business Media, LLC*.

- [KPS-14] Kompas. "Ponsel Pintar Berkuasa, Samsung dan *Android* Memimpin". URL: <http://teknokompas.com/read/2013/08/15/1532592/ponsel.pintar.berkuasa.samsung.dan.android.memimpin>. Diakses tanggal 21 September 2014.
- [NPR-11] Nugraha, M Pasca, Munir, Rinaldi. 2011. Pengembangan Aplikasi *QR Code Generator* dan *QR Code Reader* dari Data Berbentuk *Image*.
- [PAS-10] Pramono, Asep. 2010. Analisis Dan Penerapan *Sms Gateway* Pada "Media Info Bandara Jogja" Di Bandar Udara Internasional Adisucipto Yogyakarta.
- [PRA-12] Prayogo, Renusa Andra. 2012. Penggunaan Enkripsi pada *QR Code*
- [PRS-02] Pressman, Roger S. 2002. *Rekayasa Perangkat Lunak: Pendekatan Praktisi*. CN Harnaningrum. Yogyakarta.
- [QRC-15] *QR Code.com*. "Information Capacity and Versions of the QR Code" URL: <http://www.qrcode.com/en/about/version.html>. Diakses pada 8 Januari 2015.
- [RMR-14] Ramadhan, Mohammad Rizki. 2014. Implementasi Teknik *Web Scraping* Untuk Pembuatan *Web Service* Harga Barang Situs *E-Commerce* Dengan Perangkat Bergerak Sebagai Klien.
- [RYD-06] Rahayu, Yeni Dwi, dkk. 2006. Pembuatan Aplikasi Pembacaan *Quick Response Code* Menggunakan Perangkat *Mobile* Berbasis *J2ME* Untuk Identifikasi Suatu Barang. Surabaya: Politeknik Elektronika Negeri Surabaya Institut Teknologi Sepuluh Nopember.
- [SPM-08] Proboyekti, Umi. 2008. *Software Process Model I*.
- [SUT-12] Sutanta, Edhy et.al. 2012. "Kebutuhan *Web Service* untuk Sinkronisasi Data Antar Sistem Informasi dalam E-Gov di Pemkab Bantul Yogyakarta". *JURTIK - STMIK BANDUNG* (edisi Mei 2012).
- [TKP-14] Kompas. "Prospek Belanja "Online"". URL: <http://teknokompas.com/read/2012/10/05/02371027/prospek.belanja.quotonlinequot>. Diakses pada 24 Desember 2014.
- [TLG-15] *The Telegraph*. "Orange San Diego Intel Android mobile phone review". URL: <http://www.telegraph.co.uk/technology/mobile-phone-reviews/9314362/Orange-San-Diego-Intel-Android-mobile-phone-review.html>. Diakses tanggal 25 Januari 2015
- [WAX-14] Watung, Ivan Arifard, Sinsuw, Alicia A E, Paturusi, Sary D E, Najooan, Xaverius B N. 2014. Perancangan Sistem Informasi Data Alumni Fakultas Teknik Unsrat Berbasis *Web*.

LAMPIRAN

Lampiran 1. Kode DDL Database Server

```
CREATE DATABASE u736861288_deliv

CREATE TABLE IF NOT EXISTS `pengiriman` (
  `id_pengiriman` int(11) NOT NULL,
  `kode_pengiriman` varchar(11) NOT NULL,
  `id_kurir` int(5) NOT NULL,
  `status` varchar(20) NOT NULL DEFAULT 'Belum Terkirim',
  PRIMARY KEY (`id_pengiriman`),
  UNIQUE KEY `kode_transaksi` (`kode_pengiriman`),
  KEY `id_kurir` (`id_kurir`),
  KEY `id_kurir_2` (`id_kurir`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

CREATE TABLE IF NOT EXISTS `user` (
  `id_kurir` int(11) NOT NULL,
  `username` varchar(20) NOT NULL,
  `password` varchar(16) NOT NULL,
  PRIMARY KEY (`id_kurir`),
  UNIQUE KEY `username` (`username`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Lampiran 2. connect.php

```
1 <?php
2 $server = 'mysql.idhostinger.com';
3 $user = 'u736861288_foody';
4 $passwd = 'rampage';
5 $db = 'u736861288_deliv';
6
7 $dbc = mysqli_connect($server, $user, $passwd, $db) or
8 die('Gagal terhubung dengan server');
9 ?>
```

Lampiran 3. index.html

```
1 <html>
2 <head>
3 <title>QR Delivery</title>
4 <meta charset="utf-8" />
5 <meta name="viewport" content="width=device-width,
6 height=device-height, initial-scale=1.0" />
7 <script src="cordova.js"></script>
8 <script src="js/barcodescanner.js"></script>
9 <script src="libs/jquery.js"></script>
10 <script src="libs/jquery.mobile.js"></script>
11 <link rel="stylesheet" type="text/css"
12 href="libs/jquery.mobile.css" />
13 <link rel="stylesheet" type="text/css"
14 href="css/JQTheme.css" />
15 <script type="text/javascript">
```

```
16     var u_name=window.localStorage.getItem("uname");
17     var u_pwd=window.localStorage.getItem("passwd");
18
19     $(document).on('pageinit', '#home', function(){
20         if(u_name=="||u_name==null||u_name=="null"
21         ||u_name=="undefined" ||
22         u_pwd=="||u_pwd==null ||u_pwd=="null" ||
23         u_pwd=="undefined"
24         ){
25             $.mobile.changePage("#home");
26             console.log('Please Login!');
27         } else{
28             $.mobile.changePage("#konfirmasi");
29             console.log('You can confirm a code');
30         }
31
32     });
33     console.log(u_name);
34 </script>
35 <script src="js/crypto-js/rollups/md5.js"></script>
36 <script src="js/database.js"></script>
37 <script src="js/main.js"></script>
38 <script src="js/konfirmasi.js"></script>
39 <script src="js/sinkronDB.js"></script>
40 </head>
41 <body>
42     <div data-role="page" id="home">
43         <div data-role="header" data-position="fixed">
44             <h1>QR Delivery - Home</h1>
45             <button class="ui-btn-right" id="exitApp" data-
46             icon="delete" data-iconpos="notext"
47             onclick="exitFromApp();">Exit</button>
48         </div> <!-- /header -->
49         <div data-role="content" style="text-align:
50         center">
51             
53             <h3 style="color: blue">Selamat datang di QR
54             Delivery!</h3>
55             <p style="background-color: red; color: orange;
56             margin: 10px 30px 10px 30px; padding: 5px">Untuk
57             melakukan konfirmasi pengiriman barang harap login
58             terlebih menggunakan username dan password
59             anda!</p><br><br>
60             <a href="#login" data-role="button">Login</a>
61         </div> <!-- /contain -->
62         <div data-role="footer" data-position="fixed">
63             <h1>© 2015</h1>
64         </div> <!-- /footer -->
65     </div> <!-- /home page -->
66     <div data-role="page" id="login">
67         <div data-role="header">
68             <a href="#home" data-icon="back" data-
69             iconpos="notext">Home</a>
70
71
72
```

```

73     <button id="exitApp" data-icon="delete" data-
74     iconpos="notext"
75     onclick="exitFromApp();">Exit</button>
76     <h3>Login Page</h3>
77     </div>
78     <div data-role="content">
79         <center>
80             <h1 style="color: blue">Login</h1>
81             <h3 style="color: blue">Silahkan masukkan
82             username dan password anda!</h3>
83         </center>
84         <form id="loginForm" class="ui-body ui-body-a
85         ui-corner-all" data-ajax="false">
86             <fieldset>
87                 <div data-role="fieldcontain">
88                     <label for="username" >Username</label>
89                     <input type="text" value=""
90                     name="username" id="username" data-theme="b"/>
91                 </div>
92                 <div data-role="fieldcontain">
93                     <label for="password">Password</label>
94                     <input type="password" value=""
95                     name="password" id="password" data-theme="b" />
96                 </div>
97                 <input type="button" name="submit"
98                 id="submit" value="Login">
99             </fieldset>
100         </form>
101     </div>
102     <div data-role="footer" data-position="fixed">
103         <h1>@ 2015</h1>
104     </div> <!-- /footer -->
105 </div>
106 <div data-role="page" id="konfirmasi">
107     <div data-role="header" data-position="fixed">
108         <button id="logout" name="logout" data-
109         icon="action" data-iconpos="notext"
110         onclick="logOutUser();">Logout</button>
111         <h1>QR Delivery - User</h1>
112         <button id="exitApp" data-icon="delete" data-
113         iconpos="notext"
114         onclick="exitFromApp();">Exit</button>
115     </div> <!-- /header -->
116     <div data-role="content" style="text-align:
117     center">
118         
120         <div class="welcome" style="color: blue"></div>
121         <button name="scanCode" id="scanCode"
122         onclick="scan()">Scan QR Code</button>
123         <a href="#manual" class="ui-btn">Masukkan
124         Kode</a>
125         <button name="sinkronDB" id="sinkronDB"
126         onclick="sinkronMyDB();">Update Database</button>
127         <div id="scan_results"></div>
128     </div> <!-- /contain -->

```

```

129     <div data-role="footer" data-position="fixed">
130         <h1>@ 2015</h1>
131     </div> <!-- /footer -->
132 </div> <!-- /konfirmasi page -->
133 <div data-role="page" id="manual">
134     <div data-role="header" data-position="fixed">
135         <button id="logout" name="logout" data-
136             icon="action" data-iconpos="notext"
137             onclick="logOutUser();">Logout</button>
138         <h1>QR Delivery - Konfirmasi Kode</h1>
139         <button id="exitApp" data-icon="delete" data-
140             iconpos="notext"
141             onclick="exitFromApp();">Exit</button>
142     </div> <!-- /header -->
143     <div data-role="content" style="text-align:
144         center">
145         <h3 style="color: blue">Masukkan kode
146             konfirmasi.</h3>
147         <form id="konfirmasi" class="ui-body ui-body-a
148             ui-corner-all" data-ajax="false">
149             <fieldset>
150                 <div data-role="fieldcontain">
151                     <input type="text" value="" name="kode"
152                         id="kode" placeholder="Kode Konfirmasi" data-
153                         theme="b" />
154                 </div>
155                 <input type="button" name="submit"
156                     id="submit" value="Konfirmasi" />
157             </fieldset>
158         </form>
159     </div> <!-- /contain -->
160     <div data-role="footer" data-position="fixed">
161         <h1>@ 2015</h1>
162     </div> <!-- /footer -->
163 </div> <!-- /manual page -->
164 </body>
</html>

```

Lampiran 4. index.css

```

1  * {
2  -webkit-tap-highlight-color: rgba(0,0,0,0); /* make
3      transparent link selection, adjust last value
4      opacity 0 to 1.0 */
5  }
6  body {
7      -webkit-touch-callout: none; /* prevent
8          callout to copy image, etc when tap to hold */
9      -webkit-text-size-adjust: none; /* prevent
10         webkit from resizing text to fit */
11         -webkit-user-select: none; /* prevent
12             copy paste, to allow, change 'none' to 'text' */
13         background-color:#E4E4E4;
14         background-image:linear-gradient(top, #A7A7A7 0%,
15             #E4E4E4 51%);

```

```

16 background-image:-webkit-linear-gradient(top, #A7A7A7
17     0%, #E4E4E4 51%);
18 background-image:-ms-linear-gradient(top, #A7A7A7 0%,
19     #E4E4E4 51%);
20 background-image:-webkit-gradient(
21     linear,
22     left top,
23     left bottom,
24     color-stop(0, #A7A7A7),
25     color-stop(0.51, #E4E4E4)
26 );
27 background-attachment:fixed;
28 font-family:'HelveticaNeue-Light', 'HelveticaNeue',
29     Helvetica, Arial, sans-serif;
30 font-size:12px;
31 height:100%;
32 margin:0px;
33 padding:0px;
34 text-transform:uppercase;
35 width:100%;
36 }
37 /* Portrait layout (default) */
38 .app {
39     background:url(..img/logo.png) no-repeat center top;
40     /* 170px x 200px */
41     position:absolute; /* position in the
42         center of the screen */
43     left:50%;
44     top:50%;
45     height:50px; /* text area height */
46     width:225px; /* text area width */
47     text-align:center;
48     padding:180px 0px 0px 0px; /* image height is
49         200px (bottom 20px are overlapped with text) */
50     margin:-115px 0px 0px -112px; /* offset vertical:
51         half of image height and text area height */
52 }
53
54
55 /* Landscape layout (with min-width) */
56 @media screen and (min-aspect-ratio: 1/1) and (min-
57     width:400px) {
58     .app {
59         background-position:left center;
60         padding:75px 0px 75px 170px; /* padding-top +
61             padding-bottom + text area = image height */
62         margin:-90px 0px 0px -198px; /* offset vertical:
63             half of image height */
64     }
65 }
66 h1 {
67     font-size:24px;
68     font-weight:normal;
69     margin:0px;
70     overflow:visible;
71     padding:0px;
72     text-align:center;

```

```

73 }
74 .event {
75   border-radius:4px;
76   -webkit-border-radius:4px;
77   color:#FFFFFFF;
78   font-size:12px;
79   margin:0px 30px;
80   padding:2px 0px;
81 }
82 .event.listening {
83   background-color:#333333;
84   display:block;
85 }
86 .event.received {
87   background-color:#4B946A;
88   display:none;
89 }
90 @keyframes fade {
91   from { opacity: 1.0; }
92   50% { opacity: 0.4; }
93   to { opacity: 1.0; }
94 }
95 @-webkit-keyframes fade {
96   from { opacity: 1.0; }
97   50% { opacity: 0.4; }
98   to { opacity: 1.0; }
99 }
100 .blink {
101   animation:fade 3000ms infinite;
102   -webkit-animation:fade 3000ms infinite;
103 }

```

Lampiran 5. index.js

```

1  var app = {
2    // Application Constructor
3    initialize: function() {
4      this.bindEvents();
5    },
6    // Bind Event Listeners
7    //
8    // Bind any events that are required on startup.
9    Common events are:
10   // 'load', 'deviceready', 'offline', and 'online'.
11   bindEvents: function() {
12     document.addEventListener('deviceready',
13     this.onDeviceReady, false);
14   },
15   // deviceready Event Handler
16   //
17   // The scope of 'this' is the event. In order to call
18   the 'receivedEvent'
19   // function, we must explicitly call
20   'app.receivedEvent(...);'
21   onDeviceReady: function() {

```

```

22     app.receiveEvent('deviceready');
23     },
24     // Update DOM on a Received Event
25     receivedEvent: function(id) {
26         var parentElement = document.getElementById(id);
27         var listeningElement =
28     parentElement.querySelector('.listening');
29         var receivedElement =
30     parentElement.querySelector('.received');
31         listeningElement.setAttribute('style',
32     'display:none;');
33         receivedElement.setAttribute('style',
34     'display:block;');
35
36         console.log('Received Event: ' + id);
37     }
38 };
39
40 app.initialize();

```

Lampiran 6. md5.js

```

1  var CryptoJS=CryptoJS||function(s,p){var
2  m={},l=m.lib={},n=function(){},r=l.Base={extend:func
3  tion(b){n.prototype=this;var h=new
4  n;b&&h.mixin(b);h.hasOwnProperty("init")||(h.init=fu
5  nction(){h.$super.init.apply(this,arguments)});h.ini
6  t.prototype=h;h.$super=this;return
7  h},create:function(){var
8  b=this.extend();b.init.apply(b,arguments);return
9  b},init:function(){},mixin:function(b){for(var h in
10 b)b.hasOwnProperty(h)&&(this[h]=b[h]);b.hasOwnProper
11 ty("toString")&&(this.toString=b.toString)},clone:fu
12 nction(){return this.init.prototype.extend(this)}},
13 q=l.WordArray=r.extend({init:function(b,h){b=this.words=
14 b||[];this.sigBytes=h!=p?h:4*b.length},toString:func
15 tion(b){return(b||t).stringify(this)},concat:functio
16 n(b){var
17 h=this.words,a=b.words,j=this.sigBytes;b=b.sigBytes;
18 this.clamp();if(j%4)for(var
19 g=0;g<b;g++)h[j+g>>2]|=(a[g>>2]>>>24-
20 8*(g%4)&255)<<<24-8*((j+g)%4);else
21 if(65535<a.length)for(g=0;g<b;g+=4)h[j+g>>2]=a[g>>>
22 2];else h.push.apply(h,a);this.sigBytes+=b;return
23 this},clamp:function(){var
24 b=this.words,h=this.sigBytes;b[h>>>2]&=4294967295<<
25 32-8*(h%4);b.length=s.ceil(h/4)},clone:function(){var
26 b=r.clone.call(this);b.words=this.words.slice(0);ret
27 urn b},random:function(b){for(var
28 h=[],a=0;a<b;a+=4)h.push(4294967296*s.random()|0);re
29 turn new
30 q.init(h,b)}},v=m.enc={},t=v.Hex={stringify:functio
31 n(b){var a=b.words;b=b.sigBytes;for(var
32 g=[],j=0;j<b;j++){var k=a[j>>>2]>>>24-
33 8*(j%4)&255;g.push((k>>>4).toString(16));g.push((k&1
34 5).toString(16))}return

```

```

35     g.join(""), parse: function(b) {for (var
36     a=b.length, g=[], j=0; j<a; j+=2) g[j]>>>3] |= parseInt(b.substr(j,
37     2), 16) << 24 - 4 * (j%8); return new
38     q.init(g, a/2) }}, a=v.Latin1={stringify: function(b) {var
39     r a=b.words; b=b.sigBytes; for (var
40     g=[], j=0; j<b; j++) g.push(String.fromCharCode(a[j]>>>2
41     >>>24 - 8 * (j%4) & 255)); return
42     g.join(""), parse: function(b) {for (var
43     a=b.length, g=[], j=0; j<a; j++) g[j]>>>2] |= (b.charCodeAt(
44     j) & 255) << 24 - 8 * (j%4); return new
45     q.init(g, a) }}, u=v.Utf8={stringify: function(b) {try {re
46     turn
47     decodeURIComponent(escape(a.stringify(b))) } catch (g) {
48     throw Error("Malformed UTF-8
49     data"); }}, parse: function(b) {return
50     a.parse(unescape(encodeURIComponent(b))) }},
51     g=l.BufferedBlockAlgorithm=r.extend({reset: function() {th
52     is._data=new
53     q.init; this._nDataBytes=0}, _append: function(b) {"stri
54     ng"==typeof
55     b && (b=u.parse(b)); this._data.concat(b); this._nDataBy
56     tes+=b.sigBytes}, _process: function(b) {var
57     a=this._data, g=a.words, j=a.sigBytes, k=this.blockSize
58     , m=j / (4*k), m=b?s.ceil(m):s.max((m|0) -
59     this._minBufferSize, 0); b=m*k; j=s.min(4*b, j); if (b) {fo
60     r (var
61     l=0; l<b; l+=k) this._doProcessBlock(g, l); l=g.splice(0,
62     b); a.sigBytes-=j} return new
63     q.init(l, j)}, clone: function() {var
64     b=r.clone.call(this);
65     b._data=this._data.clone(); return
66     b}, _minBufferSize: 0}); l.Hasher=g.extend({cfg:r.exten
67     d(), init: function(b) {this.cfg=this.cfg.extend(b); thi
68     s.reset()}, reset: function() {g.reset.call(this); this.
69     _doReset()}, update: function(b) {this._append(b); this.
70     _process(); return
71     this}, finalize: function(b) {b && this._append(b); return
72     this._doFinalize()}, blockSize: 16, _createHelper: funct
73     ion(b) {return function(a, g) {return (new
74     b.init(g)).finalize(a)}}, _createHmacHelper: function(
75     b) {return function(a, g) {return (new k.HMAC.init(b,
76     g)).finalize(a)}}}); var k=m.algo={}; return m}(Math);
77     (function(s) {function
78     p(a, k, b, h, l, j, m) {a=a+(k&b|~k&h)+l+m; return (a<<j|a>>>
79     32-j)+k} function
80     m(a, k, b, h, l, j, m) {a=a+(k&h|b&~h)+l+m; return (a<<j|a>>>
81     32-j)+k} function
82     l(a, k, b, h, l, j, m) {a=a+(k^b^h)+l+m; return (a<<j|a>>>32-
83     j)+k} function
84     n(a, k, b, h, l, j, m) {a=a+(b^(k|~h))+l+m; return (a<<j|a>>>
85     32-j)+k} for (var
86     r=CryptoJS, q=r.lib, v=q.WordArray, t=q.Hasher, q=r.algo
87     , a=[], u=0; 64>u; u++) a[u]=4294967296*s.abs(s.sin(u+1))
88     |0; q=q.MD5=t.extend({_doReset: function() {this._hash=
89     new
90     q.init(a, u)}});
91     }

```

```

92     v.init([1732584193,4023233417,2562383102,271733878])
93     },
94     _doProcessBlock:function(g,k){for(var b=0;16>b;b++){var
95     h=k+b,w=g[h];g[h]=(w<<8|w>>>24)&16711935|(w<<24|w>>>
96     8)&4278255360}var
97     b=this._hash.words,h=g[k+0],w=g[k+1],j=g[k+2],q=g[k+
98     3],r=g[k+4],s=g[k+5],t=g[k+6],u=g[k+7],v=g[k+8],x=g[
99     k+9],y=g[k+10],z=g[k+11],A=g[k+12],B=g[k+13],C=g[k+1
100    4],D=g[k+15],c=b[0],d=b[1],e=b[2],f=b[3],c=p(c,d,e,f
101    ,h,7,a[0]),f=p(f,c,d,e,w,12,a[1]),e=p(e,f,c,d,j,17,a
102    [2]),d=p(d,e,f,c,q,22,a[3]),c=p(c,d,e,f,r,7,a[4]),f=p
103    (f,c,d,e,s,12,a[5]),e=p(e,f,c,d,t,17,a[6]),d=p(d,e,
104    f,c,u,22,a[7]),
105    c=p(c,d,e,f,v,7,a[8]),f=p(f,c,d,e,x,12,a[9]),e=p(e,f,c,d
106    ,y,17,a[10]),d=p(d,e,f,c,z,22,a[11]),c=p(c,d,e,f,A,7
107    ,a[12]),f=p(f,c,d,e,B,12,a[13]),e=p(e,f,c,d,C,17,a[1
108    4]),d=p(d,e,f,c,D,22,a[15]),c=m(c,d,e,f,w,5,a[16]),f
109    =m(f,c,d,e,t,9,a[17]),e=m(e,f,c,d,z,14,a[18]),d=m(d,
110    e,f,c,h,20,a[19]),c=m(c,d,e,f,s,5,a[20]),f=m(f,c,d,e
111    ,y,9,a[21]),e=m(e,f,c,d,D,14,a[22]),d=m(d,e,f,c,r,20
112    ,a[23]),c=m(c,d,e,f,x,5,a[24]),f=m(f,c,d,e,C,9,a[25]
113    ),e=m(e,f,c,d,q,14,a[26]),d=m(d,e,f,c,v,20,a[27]),c=
114    m(c,d,e,f,B,5,a[28]),f=m(f,c,
115    d,e,j,9,a[29]),e=m(e,f,c,d,u,14,a[30]),d=m(d,e,f,c,A,20,
116    a[31]),c=l(c,d,e,f,s,4,a[32]),f=l(f,c,d,e,v,11,a[33]
117    ),e=l(e,f,c,d,z,16,a[34]),d=l(d,e,f,c,C,23,a[35]),c=
118    l(c,d,e,f,w,4,a[36]),f=l(f,c,d,e,r,11,a[37]),e=l(e,f
119    ,c,d,u,16,a[38]),d=l(d,e,f,c,y,23,a[39]),c=l(c,d,e,f
120    ,B,4,a[40]),f=l(f,c,d,e,h,11,a[41]),e=l(e,f,c,d,q,16
121    ,a[42]),d=l(d,e,f,c,t,23,a[43]),c=l(c,d,e,f,x,4,a[44
122    ]),f=l(f,c,d,e,A,11,a[45]),e=l(e,f,c,d,D,16,a[46]),d
123    =l(d,e,f,c,j,23,a[47]),c=n(c,d,e,f,h,6,a[48]),f=n(f,
124    c,d,e,u,10,a[49]),e=n(e,f,c,d,
125    C,15,a[50]),d=n(d,e,f,c,s,21,a[51]),c=n(c,d,e,f,A,6,a[52
126    ]),f=n(f,c,d,e,q,10,a[53]),e=n(e,f,c,d,y,15,a[54]),d
127    =n(d,e,f,c,w,21,a[55]),c=n(c,d,e,f,v,6,a[56]),f=n(f,
128    c,d,e,D,10,a[57]),e=n(e,f,c,d,t,15,a[58]),d=n(d,e,f,
129    c,B,21,a[59]),c=n(c,d,e,f,r,6,a[60]),f=n(f,c,d,e,z,1
130    0,a[61]),e=n(e,f,c,d,j,15,a[62]),d=n(d,e,f,c,x,21,a[
131    63]);b[0]=b[0]+c[0];b[1]=b[1]+d[0];b[2]=b[2]+e[0];b[3]=
132    b[3]+f[0];_doFinalize:function(){var
133    a=this._data,k=a.words,b=8*this._nDataBytes,h=8*a.si
134    gBytes;k[h>>>5]|=128<<24-h%32;var l=s.floor(b/
135    4294967296);k[(h+64>>>9<<4)+15]=(l<<8|l>>>24)&16711935|(
136    l<<24|l>>>8)&4278255360;k[(h+64>>>9<<4)+14]=(b<<8|b>
137    >>24)&16711935|(b<<24|b>>>8)&4278255360;a.sigBytes=4
138    *(k.length+1);this._process();a=this._hash;k=a.words
139    ;for(b=0;4>b;b++)h=k[b],k[b]=(h<<8|h>>>24)&16711935|
140    (h<<24|h>>>8)&4278255360;return
141    a},clone:function(){var
142    a=t.clone.call(this);a._hash=this._hash.clone();retu
143    rn
144    a}});r.MD5=t._createHelper(q);r.HmacMD5=t._createHma
145    cHelper(q)})(Math);

```

