BAB IV
IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Implementasi sistem merupakan suatu tahap penerapan perancangan berdasarkan hasil analisis ke dalam bentuk kode-kode pemograman tertentu untuk menghasilkan perangkat lunak yang dibangun pada lingkungan yang sesungguhnya. Adapun pembahasan implementasi sistem terdiri dari spesifikasi perangkat keras, spesifikasi perangkat lunak, implementasi basis data, implementasi class, dan implementasi antarmuka.

4.1.1 Perangkat Keras Pembangun

Dalam penelitian ini, dibutuhkan perangkat keras untuk menerapkan suatu sistem yang dapat mengimplementasikan QR *code* sebagai tiket masuk. Adapun spesifikasi perangkat keras yang digunakan adalah seperti pada Tabel 4.1 dan pada Tabel 4.2.

Tabel 4.1 Spesifikasi Perangkat Keras PC

Perangkat Keras	Spesifikasi
Processor	Intel® Core 2 Duo T5700, 2.00 GHz
Display	1366 x 768 *pixels*, *LED-backlit screen*
Memory	2 GB
Graphic Adapters	Intel® GMA X4500M
Storage	160 GB
Network Adapters	Intel® WiFi Link 5100 AGN

Tabel 4.2 Spesifikasi Perangkat Keras Android

Perangkat Keras	Spesifikasi
Processor	Qualcomm Snapdragon APQ8064, 1.50 GHz
Display	*True* HD IPS *Plus capacitive touchscreen*, 1280 x 768 *pixels*, 4.7 *inches* (320 dpi)
Memory	2 GB
Storage	16 GB
Connectivity	WiFi 802.11 a/b/g/n, DC-HSDPA 42 Mbps

4.1.2 Perangkat Lunak Pembangun

Selain perangkat keras, pada penelitian ini dibutuhkan juga perangkat lunak untuk mendukung dalam pengimplementasian QR *code* sebagai tiket masuk. Adapun spesifikasi perangkat lunak yang digunakan adalah seperti pada Tabel 4.3.

Tabel 4.3 Spesifikasi Perangkat Lunak

Perangkat Lunak	Spesifikasi
Sistem Operasi PC	Microsoft Windows 8.1 Pro x86
Sistem Operasi Android	Android versi 4.4.4 KitKat
DBMS	MySql v5.5.24
Tools Pembangun	*Java Development Kit* Versi 7 ADT (*Android Development Tools*) Notepad++ WampServer v2.4

4.1.3 Implementasi Basis Data

Dalam penelitian ini, implementasi basis data dilakukan dengan menggunakan DBMS MySql. Implementasi basis data dalam bahasa SQL adalah seperti pada Tabel 4.4.

Tabel 4.4 Implementasi Basis Data pada Bahasa SQL

Nama Tabel	Perintah
event	CREATE TABLE IF NOT EXISTS `event` (`noEvent` int(6) NOT NULL AUTO_INCREMENT, `namaEvent` varchar(100) DEFAULT NULL, `evCode` char(3) DEFAULT NULL, `lokasi` varchar(255) DEFAULT NULL, `tglEvent` datetime DEFAULT NULL, `htm` int(7) DEFAULT NULL, `kapasitas` int(6) DEFAULT NULL, PRIMARY KEY (`noEvent`) UNIQUE KEY `evCode` (`evCode`),) ENGINE=InnoDB DEFAULT CHARSET=latin1;
pemesanan	CREATE TABLE IF NOT EXISTS `pesanan` (`idPmsn` int(16) NOT NULL AUTO_INCREMENT, `noEvent` int(6) NOT NULL,

	`noReg` varchar(25) DEFAULT NULL, `nama` varchar(100) DEFAULT NULL, `email` varchar(100) DEFAULT NULL, `telepon` varchar(15) DEFAULT NULL, `jumlahTiket` int(3) DEFAULT '0', `tgl` datetime DEFAULT NULL, `noResi` varchar(25) DEFAULT NULL, `status` enum('Belum Lunas','Lunas') DEFAULT 'Belum Lunas', `admin` int(6) DEFAULT NULL, PRIMARY KEY (`idPmsn`), UNIQUE KEY `noReg` (`noReg`), FOREIGN KEY (`admin`) REFERENCES `user` (`idUser`) ON DELETE NO ACTION ON UPDATE CASCADE, KEY `idPmsn` (`idPmsn`), FOREIGN KEY (`noEvent`) REFERENCES `event` (`noEvent`) ON DELETE CASCADE ON UPDATE CASCADE) ENGINE=InnoDB DEFAULT CHARSET=latin1;
tiket	CREATE TABLE IF NOT EXISTS `tiket` (`idTiket` int(6) NOT NULL AUTO_INCREMENT, `noTiket` varchar(20) DEFAULT NULL, `noEvent` int(6) DEFAULT NULL, `noReg` varchar(25) DEFAULT NULL, `stat` enum('Non-aktif','Aktif','Terverifikasi') DEFAULT 'Non-aktif', `timeCheck` datetime DEFAULT NULL, PRIMARY KEY (`idTiket`), FOREIGN KEY (`noEvent`) REFERENCES `event` (`noEvent`) ON DELETE CASCADE ON UPDATE CASCADE, FOREIGN KEY (`noReg`) REFERENCES `pesanan` (`noReg`) ON DELETE CASCADE ON UPDATE CASCADE) ENGINE=InnoDB DEFAULT CHARSET=latin1;
user	CREATE TABLE IF NOT EXISTS `user` (`idUser` int(6) NOT NULL AUTO_INCREMENT, `nama` varchar(50) DEFAULT NULL, `usrnm` varchar(20) DEFAULT NULL, `pswd` varchar(32) DEFAULT NULL, `lastLogin` datetime DEFAULT NULL, PRIMARY KEY (`idUser`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;

4.1.4 Implementasi Class

Implementasi *class* merupakan implementasi dari perancangan diagram kelas yang terbentuk berdasarkan diagram *use case* kedalam bentuk fisik file dalam pembuatan dan jalannya proses dalam sistem. Dalam penelitian ini digunakan dua jenis bahasa pemograman yaitu php dan java sehingga ekstensi dari file tersebut ada yang berekstensi .php dan .java. Untuk lebih lengkapnya, dapat dilihat pada Tabel 4.5.

Tabel 4.5 Implementasi Class

No.	Nama Kelas	Nama File
1	ControlDecode	ControlDecode.php
2	ControlEncode	ControlEncode.php
3	ControlEvent	ControlEvent.php
4	ControlLogin	ControlLogin.php
5	ControlPemesanan	ControlPemesanan.php
6	ControlUser	ControlUser.php
7	FomLogin	adm-login.php
8	FormAdmin	adm-bdgevent.php
9	FormEvent	adm-event.php
10	FormPesanan	adm-order.php
11	FormPengunjung	index.php
12	FormPemesanan	order.php
13	FormUser	adm-user.php
14	FormDetailPemesanan	order-detail.php
15	FormPembayaran	payment.php
16	FormTiket	ticket.php
17	QRLib	qrlib.php
18	Decoder	MainActivity.java

4.1.5 Implementasi Antarmuka

Dalam penelitian ini, sistem terdiri dari dua subsistem yaitu *Encoder* dan *Decoder*. Seperti sudah dijelaskan sebelumnya, *Encoder* berbasis *website* sedankan *Decoder* berbasis *mobile* (Android) sehingga implementasi pun terbagi menjadi dua jenis.

4.1.5.1 Antarmuka Encoder

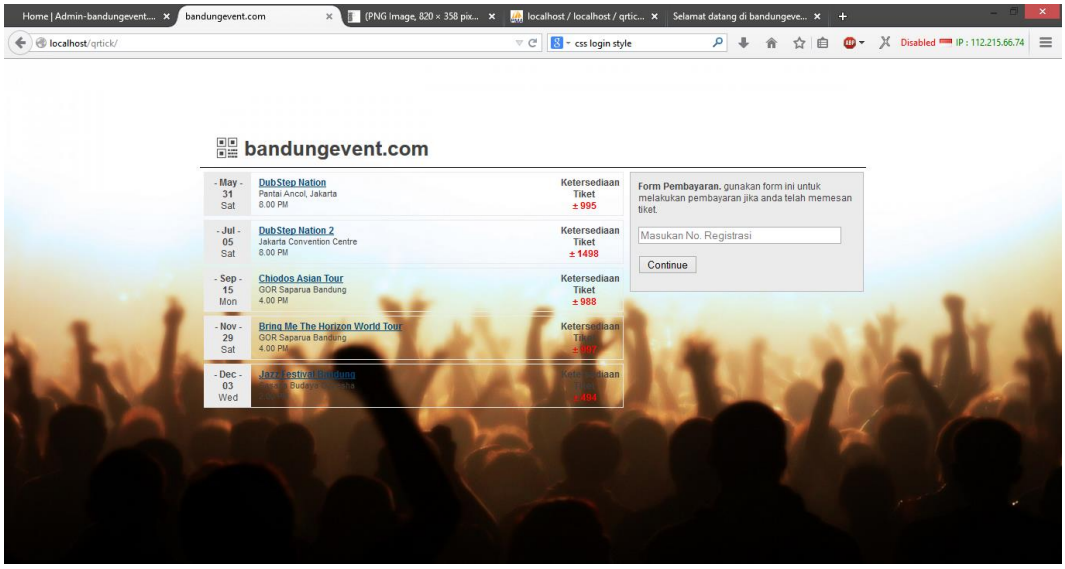
Antarmuka dari *encoder* diimplementasikan dalam bentuk halaman *website*. Setiap halaman *website* ini diimplementasikan pada masing-masing file yang berbeda. Tabel 4.6 berikut merupakan implementasi dari setiap halaman *website* pada subsistem *encoder*.

Tabel 4.6 Implementasi Antarmuka Encoder

No	Nama Halaman	Deskripsi	Nama File
1	Halaman Pengunjung	Halaman pengunjung merupakan halaman utama yang diakses oleh pengunjung. Halaman ini menampilkan daftar acara yang diselenggarakan oleh *Event Organizer*	index.php
2	Halaman Pemesanan	Halaman pemesanan merupakan halaman yang berfungsi untuk melayani pemesanan tiket yang dilakukan oleh pengunjung. Pada halaman ini terdapat fungsional untuk menyimpan data pemesanan pengunjung	order.php
3	Halaman Detail Pemesanan	Halaman detail pemesanan merupakan halaman yang menampilkan detail pemesanan tiket yang dilakukan oleh pengunjung. Pada halaman ini pengunjung dapat melanjutkan ke proses pembayaran atau kembali ke halaman utama	order-detail.php
4	Halaman Pembayaran	Halaman pemesanan merupakan halaman yang berfungsi untuk melayani pembayaran tiket yang dilakukan oleh pengunjung. Pada halaman ini terdapat fungsional untuk menyimpan nomor resi pembayaran yang dikirimkan oleh pengunjung	payment.php

5	Halaman Tiket	Halaman tiket merupakan halaman yang tiket yang dipesan oleh pengunjung. Pengunjung dapat mengakses halaman ini apabila telah melakukan pembayaran	ticket.php
6	Halaman Login Admin	Halaman login admin merupakan halaman yang berfungsi untuk melakukan autentifikasi administrator. Halaman ini hanya dapat diakses oleh administrator	adm-login.php
7	Halaman Admin	Halaman admin adalah halaman utama yang ditampilkan pada administrator. Pada halaman ini ditampilkan daftar *event* baik itu yang sudah lalu dan yang akan berlangsung	adm-bdgevent.php
8	Halaman Kelola Event	Halaman kelola event adalah halaman yang menampilkan detail event dan daftar pesanan. Pada halaman ini administrator dapat melakukan update data *event* dan menghapus *event*	adm-event.php
9	Halaman Lihat Data Pesanan	Halaman lihat data pesanan adalah halaman yang menampilkan detail pesanan. Pada halaman ini administrator dapat melakukan update status pemesanan dan menampilkan tiket yang dibeli pengunjung	adm-order.php
10	Halaman Kelola Pengguna	Halaman kelola pengguna adalah halaman yang menampilkan daftar pengguna. Pada halaman ini administrator dapat melakukan update data pengguna dan menghapus data pengguna.	adm-user.php

Implementasi dari *website* ini dapat diakses secara *online*. Dengan mengakses url http://www.bandungevent.com melalui browser, maka pengunjung akan langsung terhubung ke halaman utama pengunjung. Contoh antarmuka halaman utama pengunjung adalah seperti Gambar 4.1. Untuk implementasi antarmuka lebih lengkapnya dapat dilihat pada lampiran.



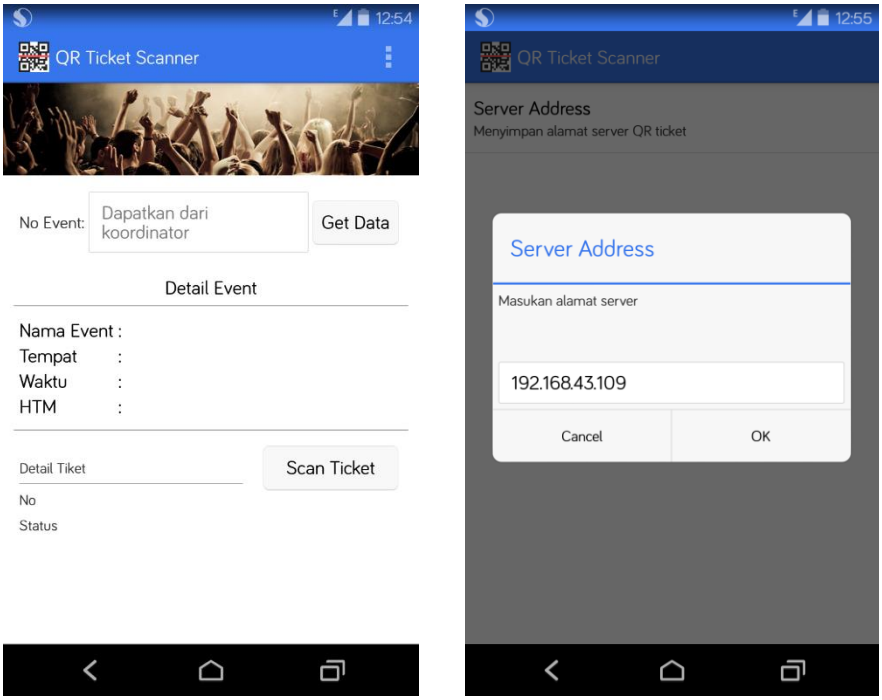
Gambar 4.1 Antarmuka Halaman Utama Pengunjung

4.1.5.2 Antarmuka Decoder

Antarmuka *decoder* merupakan antarmuka dari aplikasi *decoder* yang dibuat pada platform Android. Antarmuka yang diimplementasikan pun berbeda dengan antarmuka *encoder*. Pada *decoder* terdapat dua bagian, pertama adalah halaman utama dan yang kedua adalah halaman scanner.

1) Halaman Utama

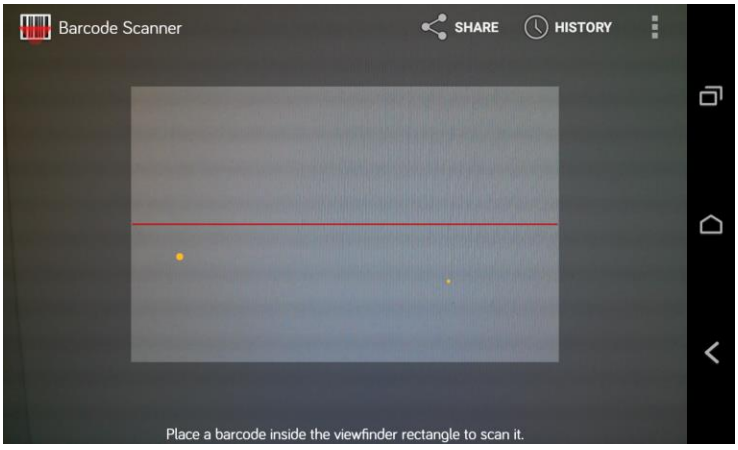
Halaman utama merupakan halaman yang menampilkan data *event* dan data tiket. Pada halaman utama terdapat menu *setting* yang berfungsi untuk pengaturan alamat server.



Gambar 4.2 Antarmuka Halaman Utama Decoder

2) Halaman Scanner

Halaman *scanner* merupakan halaman dari aplikasi *Barcode Scanner* dari ZXing *library*. Pada penelitian ini, *decoder* QR *code* memanfaatkan *library* dari ZXing sehingga pada proses *scanning* akan terhungung dengan aplikasi *Barcode Scanner*. Halaman utama terhubung *scanner* melalui tombol "*Scan Ticket*".



Gambar 4.3 Antarmuka Halaman Scanner

4.2 Pengujian Sistem

Pengujian sistem merupakan hal yang perlu dilakukan dalam penelitian ini dengan tujuan untuk menemukan kesalahan atau kekurangan yang ada pada sistem yang diuji. Pengujian sistem dimaksudkan untuk mengetahui apakah sistem yang dibuat sudah memenuhi kriteria, sesuai dengan tujuan akan dicapai. Pengujian dilakukan dengan dua tahapan. Pertama yaitu tahap pengujian *alpha* dan kedua yaitu tahap pengujian kemampuan baca dan verifikasi tiket.

4.2.1 Pengujian Alpha

Pengujian *alpha* merupakan pengujian terhadap fungsional pada sistem. Pengujian *alpha* ini dilakukan untuk menguji suatu sistem baru. Pengujian *alpha* sendiri berfokus pada persyaratan fungsional dari sistem.

4.2.1.1 Rencana Pengujian Alpha

Pengujian dilakukan dengan menguji setiap fungsional sistem berdasarkan *use case* diagram dan juga kemungkinan kesalahan yang terjadi untuk setiap fungsional. Pengujian ini dilakukan dengan metode pengujian *Black Box* dimana pengujian memperhatikan masukan ke sistem dan keluaran dari sistem. Rincian dari rencana pengujian yang akan dilakukan pada penelitian ini dapat dilihat pada Tabel 4.7.

Tabel 4.7 Rencana Pengujian Sistem

No	Fungsional yang Diuji	Detail Pengujian	Jenis Uji
1	Pemesanan tiket	Menampilkan daftar event	*Black Box*
		Menampilkan halaman pemesanan tiket sesuai dengan tiket yang dipilih	
		Menyimpan data pemesanan kedalam *database* meliputi nama, email, nomor telepon dan jumlah tiket yang dibeli pengunjung	
		Menampilkan halaman detail pemesanan dengan data yang sesuai pada saat pemesanan	

2	*Encoding* QR *code*	Menampilkan halaman pembayaran sesuai nomor registrasi pengunjung	*Black Box*
		Melakukan update nomor resi berdasarkan masukan pengunjung	
		Menyimpan data tiket yang dipesan kedalam *database*	*Black Box*
		Melakukan *encode* QR *code* berdasarkan nomor tiket	
		Menampilkan tiket dalam bentuk QR *code*	
3	Login	Menampilkan halaman login	*Black Box*
		Autentifikasi administrator, kemudian menampilkan halaman admin dan menjalankan session	
		Menghapus session login	
4	Kelola data *event*	Menampilkan data *event* dari *database*	*Black Box*
		Menyimpan data *event* baru kedalam *database*	
		Melakukan update terhadap data *event* kedalam *database*	
		Menghapus data *event* dari dalam *database*	
5	Kelola data pengguna	Menampilkan data pengguna dari *database*	*Black Box*
		Menyimpan data pengguna baru kedalam *database*	
		Melakukan update terhadap data pengguna kedalam *database*	
		Menghapus data pengguna dari dalam *database*	
6	Lihat data pesanan	Menampilkan halaman lihat data pesanan beserta data detail pesanan	*Black Box*
		Melakukan update status pembayaran kedalam *database*	

7	Update status tiket	Menggati status tiket sesuai dengan status pembayaran	*Black Box*
		Menggati status tiket sesuai dengan status verifikasi tiket	
8	*Decoding* QR *code*	Menampilkan *scanner*	*Black Box*
		Melakukan *decode* QR *code*	
		Melakukan verifikasi dari hasil d*ecoding* QR *code* dengan data tiket yang sesuai pada *database*	

4.2.1.2 Kasus dan Hasil Pengujian Alpha

Berdasarkan rencana pengujian, maka dapat dilakukan pengujian *alpha* terhadap sistem ini. Berikut adalah rincian dari kasus dan hasil pengujian *alpha*.

1) Pemesanan Tiket

Pemesanan tiket merupakan fungsional dari sistem yang melayani pemesanan tiket yang dilakukan oleh pengunjung. Tabel 4.8 berikut merupakan hasil pengujian fungsional pemesanan tiket.

Tabel 4.8 Hasil Pengujian Pemesanan Tiket

Kasus dan Hasil Uji (Data Benar)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Masukan *web address* untuk halaman pengunjung	Menampilkan daftar event yang tersedia	Sistem menampilkan halaman pengunjung beserta daftar event yang tersedia	[√] Diterima [ ] Ditolak
Menekan *link* nama *event* yang akan dipilih	Menampilkan halaman pemesanan tiket sesuai dengan *event* yang telah dipilih	Sistem menampilkan halaman pemesanan tiket sesuai dengan *event* yang telah dipilih	[√] Diterima [ ] Ditolak

Masukan data pemesanan, contoh masukan : Nama : "Devy" Email : "devy@yahoo.com" Telepon : "081123456789" Jumlah Tiket : "2"	Menyimpan data pemesanan kedalam *database* meliputi nama, email, nomor telepon dan jumlah tiket yang dibeli pengunjung	Form pemesanan terisi dengan data	[√] Diterima [ ] Ditolak
Menekan tombol "*Submit Order"* dengan data masukan terisi	Menampilkan halaman detail pemesanan dengan data yang sesuai pada saat pemesanan	Sistem menampilkan halaman pemesanan tiket dengan data pesanan sesuai dengan yang disubmit	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Memasukan data tidak valid	Tidak menyimpan data pemesanan kedalam *database*	Sistem menampilkan pesan kesalahan "*Data tidak valid*"	[√] Diterima [ ] Ditolak
Menekan tombol "*Submit Order"* tanpa data masukan	Tidak menyimpan data pemesanan kedalam *database*	Sistem menampilkan pesan kesalahan "*Data belum lengkap*"	[√] Diterima [ ] Ditolak

2) *Encoding* QR *Code*

Encoding QR *code* merupakan fungsional dari sistem untuk melakukan *encode* terhadap nilai nomor tiket menjadi QR *code*. Untuk mendapatkan tiket, pengunjung perlu menginputkan nomor resi pembayaran terlebih dahulu. Tabel 4.9 berikut merupakan hasil pengujian fungsional *encoding* QR *code*.

Tabel 4.9 Hasil Pengujian *Encoding* QR *Code*

Kasus dan Hasil Uji (Data Benar)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Menekan tombol *"Pembayaran"* Menginputkan nomor registrasi yang valid pada halaman pengunjung	Menampilkan halaman pembayaran sesuai dengan nomor registrasi	Sistem menampilkan halaman pembayaran sesuai dengan nomor registrasi pengunjung	[√] Diterima [ ] Ditolak
Menekan tombol bayar dengan masukan nomor resi yang valid	Melakukan update nomor resi berdasarkan masukan pengunjung	Sistem melakukan update terhadap data nomor resi	[√] Diterima [ ] Ditolak
Menekan tombol "*Submit Order*" dengan data masukan : Nama : "Devy" Email : "devy@yahoo.com" Telepon : "081123456789" Jumlah Tiket : "2"	Menyimpan data tiket yang dipesan kedalam *database*	Sistem menyimpan data tiket kedalam *database* sesuai dengan jumlah dan *event* yang dipilih pengunjung	[√] Diterima [ ] Ditolak
Menekan tombol "*View Ticket"*	Melakukan *encode* QR *code* berdasarkan nomor tiket	Sistem melakukan *encode* QR *code*	[√] Diterima [ ] Ditolak
Menekan tombol "*View Ticket"*	Menampilkan tiket dalam bentuk QR *code*	Sistem menampilkan QR *code* hasil *encoding* nomor tiket	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Memasukan nomor registrasi yang salah	Menampilkan pesan kesalahan	Sistem menampilkan pesan	[√] Diterima [ ] Ditolak

		kesalahan "*Nomor registrasi tidak valid*"	
Menekan tombol bayar tanpa ada masukan nomor resi	Menampilkan pesan kesalahan	Sistem menampilkan pesan kesalahan "*Nomor resi kosong*"	[√] Diterima [ ] Ditolak
Menekan tombol "*Submit Order"* tanpa data masukan	Tidak menyimpan data tiket yang dipesan kedalam *database*	Sistem menampilkan pesan kesalahan "*Data belum lengkap*"	[√] Diterima [ ] Ditolak

3) Login

Login merupakan fungsional dari sistem untuk melakukan autentifikasi terhadap administrator. Tabel 4.10 berikut merupakan hasil pengujian fungsional login.

Tabel 4.10 Hasil Pengujian Login

Kasus dan Hasil Uji (Data Benar)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Mengakses halaman login admin	Menampilkan halaman login	Sistem menampilkan halaman login admin	[√] Diterima [ ] Ditolak
Menekan tombol login dengan data masukan: *username*: "admin1", *password*: "123adm"	Melakukan autentifikasi administrator, kemudian menampilkan halaman admin dan menjalankan session	Data masukan diterima dan sistem pun menampilkan halaman admin. Session juga dapat dijalankan.	[√] Diterima [ ] Ditolak
Menekan tombol logout	Menghapus session login	Menampilkan kembali halaman login dan session pun terhapus	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji (Data Salah)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Menekan tombol login tanpa data masukan	Tidak dapat login	Sistem tidak login dan menampilkan pesan *"Username atau password belum diisi"*	[√] Diterima [ ] Ditolak
Menekan tombol login dengan data masukan: *username*: "dsfhsyha", *password*: "yt43erg"	Tidak dapat login	Sistem tidak login dan menampilkan pesan *"Login Gagal"*	[√] Diterima [ ] Ditolak

4) Kelola Data *Event*

Kelola data *event* merupakan fungsional dari sistem untuk melakukan pengelolaan data *event*. Tabel 4.11 Berikut merupakan hasil pengujian fungsional kelola data *event*.

Tabel 4.11 Hasil Pengujian Kelola Data Event

Kasus dan Hasil Uji (Data Benar)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Mengakses halaman admin	Menampilkan data *event* dari *database*	Menampilkan data daftar *event* dari *database*	[√] Diterima [ ] Ditolak
Menekan tombol "*Create*" dengan data masukan : Nama *Event* : "Jazz Festival Bandung", Tempat : "Sasana Budaya	Menyimpan data *event* baru kedalam *database*	Data tersimpan kedalam *database* dan muncul pesan konfirmasi "*Data berhasil disimpan*"	[√] Diterima [ ] Ditolak

Ganesha", Kode *Event* : "JFB", Kapasitas : "500", Tanggal : "3 Desember 2014"			
Menekan tombol "*Update*" dengan mengubah salah satu data *event*	Melakukan update terhadap data *event* kedalam *database*	Data pada *database* berhasil diperbaharui dan muncul pesan konfirmasi "*Update data berhasil*"	[√] Diterima [ ] Ditolak
Menekan tombol "Hapus"	Menghapus data *event* dari dalam *database*	*Event* terhapus dari *database* dan muncul pesan konfirmasi "*Data berhasil dihapus*"	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Menekan tombol "*Create*" tanpa data masukan	Muncul pesan kesalahan "*Mohon maaf, Semua Input Form Harus Diisi !*"	Sistem menampilkan pesan kesalahan "*Mohon maaf, Semua Input Form Harus Diisi !*"	[√] Diterima [ ] Ditolak
Menekan tombol "*Update*" dengan mengosongkan salah ssatu *field* masukan	Muncul pesan kesalahan "*Mohon maaf, Semua Input Form Harus Diisi !*"	Sistem menampilkan pesan kesalahan "*Mohon maaf, Semua Input Form Harus Diisi !*"	[√] Diterima [ ] Ditolak

5) Kelola Data Pengguna

Kelola data pengguna merupakan fungsional dari sistem untuk melakukan pengelolaan data pengguna. Tabel 4.12 Berikut merupakan hasil pengujian fungsional kelola data pengguna.

Tabel 4.12 Hasil Pengujian Kelola Data Pengguna

Kasus dan Hasil Uji (Data Benar)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Mengakses halaman admin	Menampilkan data pengguna dari *database*	Menampilkan data daftar pengguna dari *database*	[√] Diterima [ ] Ditolak
Menekan tombol "*Create*" dengan data masukan : Nama: Edwin Fajar", Username : "edw8", Password: "*******"	Menyimpan data pengguna baru kedalam *database*	Data tersimpan kedalam *database* dan muncul pesan konfirmasi "*Data berhasil disimpan*"	[√] Diterima [ ] Ditolak
Menekan tombol "*Update*" dengan mengubah salah satu data pengguna	Melakukan update terhadap data pengguna kedalam *database*	Data pada *database* berhasil diperbaharui dan muncul pesan konfirmasi "*Update data berhasil*"	[√] Diterima [ ] Ditolak
Menekan tombol "Hapus"	Menghapus data pengguna dari dalam *database*	*Event* terhapus dari *database* dan muncul pesan konfirmasi "*Data berhasil dihapus*"	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji (Data Salah)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Menekan tombol "*Create*" tanpa data masukan	Muncul pesan kesalahan "*Mohon maaf, Semua Input Form Harus Diisi !*"	Sistem menampilkan pesan kesalahan "*Mohon maaf, Semua Input Form Harus Diisi !*"	[√] Diterima [ ] Ditolak
Menekan tombol "*Update*" dengan mengosongkan salah ssatu *field* masukan	Muncul pesan kesalahan "*Mohon maaf, Semua Input Form Harus Diisi !*"	Sistem menampilkan pesan kesalahan "*Mohon maaf, Semua Input Form Harus Diisi !*"	[√] Diterima [ ] Ditolak

6) Lihat Data Pesanan

Lihat data pesanan merupakan fungsional dari sistem untuk melakukan melihat data pesanan. Tabel 4.13 Berikut merupakan hasil pengujian fungsional lihat data pesanan.

Tabel 4.13 Hasil Pengujian Lihat Data Pesanan

Kasus dan Hasil Uji (Data Benar)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Menekan *link* (gambar *check* tiket) pada pesanan yang akan dilihat detailnya	Menampilkan halaman lihat data pesanan beserta data detail pesanan	Sistem menampilkan halaman lihat data pesanan beserta data detail pesanan sesuai dengan yang dipilih	[√] Diterima [ ] Ditolak
Menekan *link "Update* Status Pembayaran"	Melakukan update terhadap status pembayaran pada *database*	Sistem melakukan update status pembayaran pada *database*	[√] Diterima [ ] Ditolak

7) Update Status Tiket

Update status tiket merupakan fungsional dari sistem untuk melakukan update status tiket. Tabel 4.14 Berikut merupakan hasil pengujian fungsional update status tiket.

Tabel 4.14 Hasil Pengujian Update Status Tiket

Kasus dan Hasil Uji (Data Benar)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Menekan *link "Update* Status Pembayaran"	Menggati status tiket sesuai dengan status pembayaran	Sistem menggati status tiket sesuai dengan status pembayaran	[√] Diterima [ ] Ditolak
Menekan tombol "Proses" pada saat tiket terverifikasi	Menggati status tiket sesuai dengan status verifikasi tiket	Sistem menggati status tiket sesuai dengan status verifikasi tiket	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Menekan tombol "Tidak" pada saat tiket terverifikasi	Status tiket tidak berubah	Sistem tidak mengganti status tiket	[√] Diterima [ ] Ditolak

8) Decoding QR code

Decoding QR *code* merupakan fungsional dari sistem untuk melakukan *decoding* QR *code*. Tabel 4.15 Berikut merupakan hasil pengujian fungsional *decoding* QR *code*.

Tabel 4.15 Hasil Pengujian *Decoding* QR *Code*

Kasus dan Hasil Uji (Data Benar)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Menekan tombol "*Scan Ticket*"	Menampilkan QR *code scanner*	Sistem menampilkan QR *code scanner*	[√] Diterima [ ] Ditolak
Scanning QR *code*	Melakukan *decode* QR *code*	Sistem melakukan *decode* QR *code*	[√] Diterima [ ] Ditolak
Scanning tiket *event* yang berbentuk QR *code* dengan status "Aktif"	Melakukan verifikasi dari hasil d*ecoding* QR *code* dengan data tiket yang sesuai pada *database*	Sistem melakukan *decode* QR *code* dan menampilkan nomor dan status tiket	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Aksi / Data Masukan	**Yang Diharapkan**	**Hasil Pengamatan**	**Kesimpulan**
Scanning tiket *event* yang berbentuk QR *code* dengan status "Non-Aktif" dan "Terverifikasi"	Menampilkan pesan "*Tiket tidak valid*"	Sistem menampilkan pesan "*Tiket tidak valid*"	[√] Diterima [ ] Ditolak
Scanning tiket *event* yang berbentuk QR *code* dari *event* yang berbeda	Menampilkan pesan "*Tiket tidak valid*"	Sistem menampilkan pesan "*Tiket tidak valid*"	[√] Diterima [ ] Ditolak

4.2.1.3 Kesimpulan Pengujian Alpha

Berdasarkan hasil pengujian *Alpha* yang telah dilakukan, maka dapat disimpulkan bahwa secara fungsional sistem yang memanfaatkan QR *code* sebagai tiket masuk suatu *event* dalam penelitian ini telah berjalan sesuai dengan yang diharapkan. Sistem dapat melakukan fungsional dari dari mulai proses pemesanan, *encoding* QR *code* hingga melakukan verifikasi tiket.

4.2.2 Pengujian Kemampuan Pembacaan QR Code

Pengujian kemampuan pembacaan QR *code* merupakan pengujian yang perlu dilakukan dalam penelitian ini. Pengujian ini dilakukan untuk mengetahui bagaimana tingkat koreksi kesalahan mempengaruhi kemampuan pembacaan *scanner* terhadap QR *code* dalam berbagai kondisi.

4.2.2.1 Rencana Pengujian Kemampuan Pembacaan QR Code

Pengujian kemampuan pembacaan QR *code* dilakukan pada QR *code* yang memiliki tingkat koreksi kesalahan berbeda, diantaranya *Low*, *Medium*, *Quartile* dan *High*. Dari keempat tingkat koreksi kesalahan tersebut, masing-masing memiliki kemampuan untuk mengembalikan *codewords* yang rusak seperti pada Tabel 4.16 berikut.

Tabel 4.16 Tingkat Kemampuan Koreksi Kesalahan

Tingkat EC	**Kemampuan Koreksi Kesalahan**
Level L (*Low*)	7% dari codewords dapat dikembalikan
Level M (*Medium*)	15% dari codewords dapat dikembalikan
Level Q (*Quartile*)	25% dari codewords dapat dikembalikan
Level H (*High*)	30% dari codewords dapat dikembalikan

QR *code* tersebut kemudian dipindai dari dua media berbeda, yaitu pada media digital (ditampilkan pada layar ponsel) dan media cetak (dicetak pada kertas). Kemudian pengujian dilakukan pada kondisi QR *code* yang utuh dan tidak utuh (sengaja dirobek pada media cetak) dengan tingkat kerusakan 5% dari luas QR *code* dan 10% dari luas QR *code*. Jenis kerusakannya yaitu disobek pada bagian pojok kanan bawah seperti pada Gambar 4.4. Pojok lainnya tidak dirusak karena pada pojok lainnya terdapat *finder pattern* yang bilamana hilang atau tidak dapat terbaca, maka QR *code* tidak dapat dipindai.

QR *code* kemudian diuji pada kondisi pengujian yang kurang cahaya yaitu didalam ruangan pada sore hari dan cukup cahaya yaitu diluar ruangan pada siang dan sore hari. Rincian dari rencana pengujian kemampuan pembacaan QR *code* dapat dilihat pada Tabel 4.17.



Gambar 4.4 Contoh QR *Code* dengan kerusakan 10%

Tabel 4.17 Rencana Pengujian Kemampuan Pembacaan QR Code

Kondisi QR Code	Media QR Code	Kondisi Pengujian	Tingkat Koreksi Kesalahan	Parameter Uji
Utuh	Layar Ponsel	Cukup Cahaya	*Low*	Jumlah dan persentase QR *code* yang terbaca
			Medium	
			Quartile	
			High	
		Cukup Cahaya	*Low*	
			Medium	
			Quartile	
			High	
	Kertas	Kurang Cahaya	*Low*	
			Medium	
			Quartile	
			High	
		Kurang Cahaya	*Low*	
			Medium	
			Quartile	
			High	
Tidak Utuh	Kertas	Cukup Cahaya	*Low*	
			Medium	
			Quartile	
			High	
		Kurang Cahaya	*Low*	
			Medium	
			Quartile	
			High	

4.2.2.2 Hasil Pengujian Kemampuan Pembacaan QR Code

Hasil dari pengujian kemampuan pembacaan QR *code* dibagi menjadi dua bagian berdasarkan kondisi QR *code*. Dari hasil pengamatan sesuai dengan rencana pengujian kemampuan pembacaan QR *code*, maka didapatkanlah hasil pengujian. Adapun hasil pengujian kemampuan pembacaan QR *code* adalah sebagai berikut.

1) Hasil Pengujian pada QR *Code* Utuh

Pengujian ini dilakukan dengan memindai QR *code* yang ada pada dua media berbeda, yaitu yang ditampilkan pada layar ponsel dan yang dicetak pada kertas. Kemudian pengujian dilakukan pada keadaan cukup cahaya dengan lokasi diluar ruangan pada pukul 13.00 – 16.00 dan pada keadaan kurang cahaya dengan mengambil sampel acak dari pengujian diluar ruangan pada pukul 17.00 – 19.00 dan pengujian didalam ruangan. Hasil dari pengujian dapat dilihat pada Tabel 4.18.

Tabel 4.18 Hasil Pengujian Pada Kondisi QR Code Utuh

QR Code Ditampilkan pada Layar Ponsel			
Kondisi Pengujian	**Tingkat Koreksi Kesalahan**	**Jumlah QR Code Terbaca/Jumlah Sampel**	**Persentase QR Code Terbaca**
Cukup Cahaya	*Low*	20/20	100%
	Medium	20/20	100%
	Quartile	20/20	100%
	High	20/20	100%
Kurang Cahaya	*Low*	20/20	100%
	Medium	20/20	100%
	Quartile	20/20	100%
	High	20/20	100%
Rata-Rata		160/160	**100%**
QR Code Dicetak pada Kertas			
Kondisi Pengujian	**Tingkat Koreksi Kesalahan**	**Jumlah QR Code Terbaca/Jumlah Sampel**	**Persentase QR Code Terbaca**
Cukup Cahaya	*Low*	20/20	100%
	Medium	20/20	100%

	Quartile	20/20	100%
	High	20/20	100%
Kurang Cahaya	*Low*	20/20	100%
	Medium	20/20	100%
	Quartile	20/20	100%
	High	20/20	100%
Rata-Rata		160/160	**100%**

2) Hasil Pengujian pada QR *Code* Tidak Utuh

Pengujian ini dilakukan dengan memindai QR *code* yang dicetak pada kertas dengan keadaan QR *code* tidak utuh. Kemudian pengujian dilakukan pada keadaan cukup cahaya dengan lokasi diluar ruangan pada pukul 13.00 – 16.00 dan pada keadaan kurang cahaya dengan mengambil sampel acak dari pengujian diluar ruangan pada pukul 17.00 – 19.00 dan pengujian didalam ruangan. Hasil dari pengujian dapat dilihat pada Tabel 4.19.

Tabel 4.19 Hasil Pengujian Pada Kondisi QR Code Tidak Utuh

QR Code Dicetak pada Kertas				
Kondisi Pengujian	**Tingkat Koreksi Kesalahan**	**Tingkat Kerusakan QR Code**	**Jumlah QR Code Terbaca/Jumlah Sampel**	**Persentase QR Code Terbaca**
Cukup Cahaya	*Low*	5%	5/5	100%
		10%	0/5	0%
	Medium	5%	5/5	100%
		10%	1/5	20%
	Quartile	5%	5/5	100%
		10%	2/5	40%
	High	5%	5/5	100%
		10%	5/5	100%
Kurang Cahaya	*Low*	5%	5/5	100%
		10%	0/5	0%
	Medium	5%	5/5	100%
		10%	1/5	20%
	Quartile	5%	5/5	100%
		10%	2/5	40%

	High	5%	5/5	100%
		10%	5/5	100%
Rata-Rata (Total QR code terbaca / Jumlah sampel)			**56/80**	**70%**

4.2.2.3 Kesimpulan Pengujian Kemampuan Pembacaan QR Code

Berdasarkan hasil pengujian kemampuan pembacaan QR *code* yang telah dilakukan, maka dapat disimpulkan bahwa pada QR *code* dapat terbaca dengan baik pada kondisi yang utuh. Dalam berbagai kondisi, baik itu kondisi pengujian yang cukup dan jurang cahaya, seluruh QR *code* dapat terbaca dengan baik oleh *scanner*. Akan tetapi, pada QR *code* dengan kondisi tidak utuh, QR *code* dengan tingkat koreksi kesalahan selain *level High* tidak dapat terbaca dengan baik oleh *scanner* apabila tingkat kerusakan mencapai ±10% . Hal ini membuktikan bahwa tingkat koreksi kesalahan dapat mempengaruhi kemampuan baca QR *code* apabila QR *code* dalam keadaan tidak utuh.