



**RANCANG BANGUN APLIKASI PENENTUAN VENDOR PERNIKAHAN  
UNTUK USAHA *WEDDING ORGANIZER* MENGGUNAKAN METODE  
SMARTER BERBASIS *WEB***



**TUGAS AKHIR**

**Program Studi**

**S1 Sistem Informasi**

**Oleh:**

**FACHREZA ILHAM DWI CAHYO**

**14410100020**

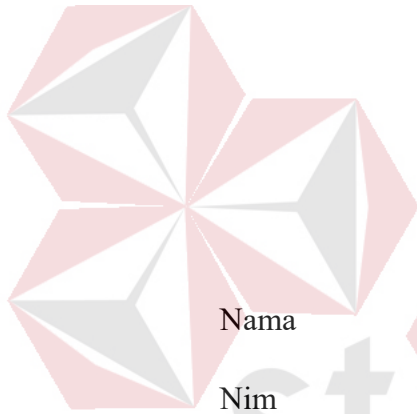
---

**FAKULTAS TEKNOLOGI DAN INFORMATIKA  
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA  
2018**

**RANCANG BANGUN APLIKASI PENENTUAN VENDOR PERNIKAHAN  
UNTUK USAHA *WEDDING ORGANIZER* MENGGUNAKAN METODE  
SMARTER BERBASIS *WEB***

**TUGAS AKHIR**

Diajukan sebagai syarat untuk mengerjakan Tugas Akhir



Oleh: **INSTITUT BISNIS  
DAN INFORMATIKA**

Nama : Fachreza Ilham Dwi Cahyo

Nim : 14410100020

Program Studi : S1 (Strata Satu)

Jurusan : Sistem Informasi **B A Y A**

**FAKULTAS TEKNOLOGI DAN INFORMATIKA  
INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA**

**2018**





***"All our dreams can come true if we have the courage to pursue them."***

***- Walt Disney***

*Semua impian kita bisa terwujud jika kita memiliki keberanian untuk mengejar  
mereka*

INSTITUT BISNIS  
DAN INFORMATIKA

stikom  
SURABAYA



*Saya persembahkan Skripsi ini kepada*

*Mama, papa tersayang, adik dan kakak, orang yang selalu membantu (Selly Marselia), teman-teman dan sahabat yang selalu memberi semangat dan motivasi*

INSTITUT BISNIS  
DAN INFORMATIKA

stikom

SURABAYA

**TUGAS AKHIR**  
**RANCANG BANGUN APLIKASI PENENTUAN VENDOR PERNIKAHAN**  
**UNTUK USAHA WEDDING ORGANIZER MENGGUNAKAN METODE**  
**SMARTER BERBASIS WEB**

Dipersiapkan dan disusun oleh

**Fachreza Ilham Dwi Cahyo**

**NIM : 14.41010.0020**

Telah diperiksa, diuji dan disetujui oleh Dewan Penguji

Pada : Maret 2018

**Susunan Dewan Penguji**

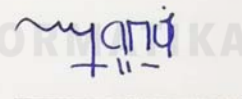
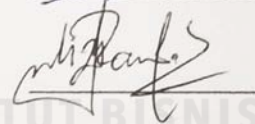
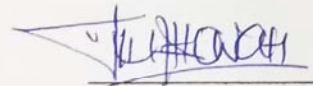
Pembimbing

I. Sulistiowati, S.Si., M.M.  
NIDN. 0719016801

II. Julianto Lemantara, S.Kom., M.Eng.  
NIDN. 0722108601

Pembahas

I. Sri Hariani Eko Wulandari, S.Kom., M.MT.  
NIDN. 0726017801



Tugas Akhir ini diterima sebagai salah satu persyaratan  
untuk memperoleh gelar Sarjana



Dr. Jusak

NIDN. 0708017101

**Dekan Fakultas Teknologi dan Informatika**

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**  
**INSTITUT BISNIS DAN INFORMATIKA STIKOM SURABAYA**

16/18  
/3

## SURAT PERNYATAAN

### PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa Institut Bisnis dan Informatika Stikom Surabaya, saya:

Nama : Fachreza Ilham Dwi Cahyo  
NIM : 14.41010.0020  
Program Studi : S1 Sistem Informasi  
Fakultas : Fakultas Teknologi dan Informatika  
Jenis Karya : Laporan Tugas Akhir  
Judul Karya : **RANCANG BANGUN APLIKASI PENENTUAN  
VENDOR PERNIKAHAN UNTUK USAHA *WEDDING*  
ORGANIZER MENGGUNAKAN METODE SMARTER  
BERBASIS *WEB***

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Institut Bisnis dan Informatika Stikom Surabaya Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalti Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar keserjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, Maret 2018

Yang menyatakan



Fachreza Ilham Dwi Cahyo

NIM: 14410100020

## ABSTRAK

Intimate Production (IP) adalah salah satu startup di bidang *wedding Organizer online* yang ada di Surabaya. Saat ini IP bergerak di bidang jasa untuk membantu calon pengantin dan keluarganya mempersiapkan pesta pernikahan sesuai dengan yang diimpikan. *Customer* mengalami kesulitan ketika ingin menentukan *vendor* apa yang ingin digunakan karena semakin banyak pilihan yang beragam dari harga dan kualitas yang juga beragam. IP dapat menjembatani pelaku usaha (*vendor*) dengan *Customer*, dengan bergabung bersama IP, *vendor* akan mendapat manfaat peluang tambahan *order* dari *customer* IP.

Solusi yang akan dibuat adalah, sebuah aplikasi berbasis *web* yang dapat menjembatani *vendor* untuk memberikan penawaran dan peluang untuk dipilih *customer*. seperti: 1) Memberikan sebuah kemudahan kepada *customer* untuk memilih paket pernikahan dari hasil menerapkan metode SMARTER, 2) Memberikan sebuah kemudahan kepada *customer* untuk perhitungan biaya acara, 3) Menjadi sebuah wadah untuk mengembangkan potensi *vendor*.

Hasil uji coba menunjukkan bahwa aplikasi ini dapat melakukan pembuatan rekomendasi paket berdasarkan kriteria pilihan *customer* dengan metode SMARTER. Aplikasi dapat membantu *customer* untuk membuat paket pernikahan secara kustomisasi dan dapat menghitung biaya paket secara otomatis. Aplikasi dapat mengembangkan potensi *vendor* dengan cara menawarkan paket yang dimiliki *vendor* kepada *customer* dengan melakukan *generate* paket sesuai kriteria *customer*. Sehingga dapat menjembatani kebutuhan *customer* dan *vendor*.

**Keywords:** *Wedding Organizer, Intimate Production, Vendor*

## KATA PENGANTAR

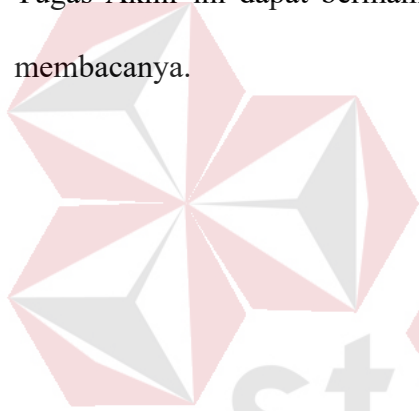
Segala puji dan syukur penulis panjatkan kehadirat Allah SWT, karena berkat rahmat dan hidayah-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul “**Rancang Bangun Aplikasi Penentuan Vendor Pernikahan Untuk Usaha *Wedding Organizer* Menggunakan Metode SMARTER Berbasis Web**”. Tugas Akhir ini merupakan syarat untuk menyelesaikan program studi Strata Satu di Fakultas Teknologi dan Informatika pada Institut Bisnis dan Informatika Stikom Surabaya.

Selama menyelesaikan Tugas Akhir ini, penulis tidak terlepas dari bantuan berbagai pihak yang telah memberikan bantuan, masukan, dan saran. Oleh karena itu, penulis menyampaikan rasa terima kasih kepada:

1. Kedua orang tua penulis yang selalu mendoakan, memberi kasih sayang, serta dorongan moril maupun materil yang tak terhingga.
2. Ibu Sulistiowati, S.Si., M.M. selaku Dosen Pembimbing I yang telah membimbing, memotivasi, dan memberi arahan kepada penulis dalam menyelesaikan Tugas Akhir
3. Bapak Julianto Lemantara, S.Kom., M.Eng. selaku Dosen Pembimbing II yang telah membimbing, memotivasi, dan memberi arahan kepada penulis dalam menyelesaikan Tugas Akhir
4. Ibu Sri Hariani Eko Wulandari, S.Kom., M.MT. selaku Pembahas.
5. Selly Marselia yang telah memberikan dukungan, semangat, dan bantuan lainnya dalam menyelesaikan Tugas Akhir.

6. Tim gopoh (Marsya, Dhya, Muchlis, Iqbal, Dani) yang telah membantu penulis dalam menyelesaikan Tugas Akhir.
7. Teman-teman Semolowaru Squad (Alwi, Dwik, Gaga, Ismail) dan teman-teman Kost Woles Cak Samsul (Evita, Fikri, Andri, Pijar, Ilham, Dimas, Mas Kemal, Mas Vicky) atas segala bantuan dan dukungannya.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan. Oleh karena itu, penulis memohon saran dan kritik dari semua pihak untuk perbaikan di masa mendatang. Semoga semua pemikiran yang tertuang dalam Tugas Akhir ini dapat bermanfaat terutama bagi penulis dan semua pihak yang membacanya.



Surabaya, Maret 2018

INSTITUT BISNIS  
DAN INFORMATIKA

stikom

SURABAYA

Penulis

## DAFTAR ISI

	Halaman
SURAT PERNYATAAN .....	v
ABSTRAK .....	vii
KATA PENGANTAR .....	viii
DAFTAR ISI.....	x
DAFTAR GAMBAR .....	xiii
DAFTAR TABEL.....	xxi
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang Masalah .....	1
1.2 Perumusan Masalah.....	4
1.3 Batasan Masalah.....	4
1.4 Tujuan.....	4
1.5 Manfaat.....	5
1.6 Sistematika Penulisan.....	5
BAB II LANDASAN TEORI .....	7
2.1 Penelitian Terdahulu.....	7
2.2 Konsep Sistem Informasi .....	9
2.3 <i>Simple Multi Attribute Rating Technique Exploiting Ranks</i> (SMARTER).....	9
2.4 <i>System Development Life Cycle (SDLC)</i> .....	11
2.5 <i>Website</i> .....	13
2.6 MySQL.....	15



2.7 PHP ( <i>Hypertext Preprocessor</i> ).....	17
2.8 <i>Framework Codeigniter</i> .....	18
2.9 <i>Crowdsourcing</i> .....	23
2.10 <i>Testing</i> .....	24
2.10.1 <i>Black Box Testing</i> .....	24
<b>BAB III ANALISIS DAN PERANCANGAN SISTEM</b> .....	<b>26</b>
3.1 Analisis Sistem .....	26
3.1.1 Analisis Proses Bisnis.....	28
3.1.2 Analisis kebutuhan pengguna .....	31
3.1.3 Analisis Kebutuhan Fungsional.....	34
3.1.4 Analisis kebutuhan non fungsional.....	66
3.1.5 Analisis kebutuhan sistem .....	66
3.2 Perancangan Sistem.....	68
3.2.1 Blok Diagram.....	69
3.2.2 Proses Modelling .....	72
3.2.3 Data Modelling .....	113
3.2.4 Perancangan Antar Muka ( <i>Interface</i> ) .....	126
3.2.5 Perancangan Uji Coba Sistem Dengan <i>Black Box</i> .....	145
<b>BAB IV IMPLEMENTASI DAN EVALUASI</b> .....	<b>165</b>
4.1 Implementasi Sistem .....	165
4.1.1 Kebutuhan Perangkat Keras.....	165
4.1.2 Kebutuhan Perangkat Lunak.....	166
4.1.3 Implementasi Halaman <i>Login</i> Pengguna .....	166

4.1.4 Implementasi Halaman Pada Admin .....	168
4.1.5 Implementasi Halaman Pada <i>Vendor</i> .....	187
4.1.6 Implementasi Halaman Pada <i>Customer</i> .....	195
4.1.7 Notification .....	205
4.1.8 <i>Forgot Password</i> .....	206
4.1.9 <i>Withdraw</i> .....	208
4.2 Uji Coba dan Evaluasi Sistem .....	209
4.2.1 Uji Coba <i>Form Login</i> .....	209
4.2.2 Pengelolaan Pada <i>Role Admin</i> .....	210
4.2.3 Pengelolaan Pada <i>Role Vendor</i> .....	218
4.2.4 Pengelolaan Pada <i>Role Customer</i> .....	223
4.2.5 Hasil Uji Coba Notifikasi .....	227
4.2.6 Hasil Uji Coba <i>Forgot Password</i> .....	228
4.2.7 Hasil Uji Coba <i>Withdraw</i> .....	228
4.2.8 Evaluasi Sistem.....	229
BAB V PENUTUP.....	231
5.1 Kesimpulan.....	231
5.2 Saran .....	232
DAFTAR PUSTAKA .....	233

## DAFTAR GAMBAR

	Halaman
Gambar 2. 1 SDLC Metode Waterfall .....	12
Gambar 2. 2 Cara Kerja CodeIgniter .....	20
Gambar 2. 3 Cara Kerja MVC .....	21
Gambar 3. 1 Diagram Model Penelitian .....	26
Gambar 3. 2 Blok Diagram Aplikasi Penentuan Vendor Pernikahan 1 .....	70
Gambar 3. 3 Blok Diagram Aplikasi Penentuan Vendor Pernikahan 2 .....	71
Gambar 3. 4 Blok Diagram Aplikasi Penentuan Vendor Pernikahan 3 .....	72
Gambar 3. 5 System Flow Login Pengguna .....	73
Gambar 3. 6 System Flow Logout .....	74
Gambar 3. 7 System Flow Dashboard Pada Admin .....	75
Gambar 3. 8 System Flow Pengelolaan Master Kategori .....	76
Gambar 3. 9 System Flow Master Sub Kriteria .....	77
Gambar 3. 10 System Flow Pengelolaan Master Nilai .....	78
Gambar 3. 11 System Flow Pengelolaan Master Bank .....	79
Gambar 3. 12 System Flow Confirmation Payment .....	80
Gambar 3. 13 System Flow Histori Konfirmasi Pembayaran .....	81
Gambar 3. 14 System Flow Legality Verification .....	82
Gambar 3. 15 System Flow Confirmation Withdraw .....	83
Gambar 3. 16 System Flow Registrasi Vendor .....	84
Gambar 3. 17 System Flow Legality Vendor .....	85
Gambar 3. 18 System Flow Dashboard Pada Vendor .....	86

Gambar 3. 19 System Flow Pengelolaan Mastrer Kategori Vendor .....	87
Gambar 3. 20 System Flow Master Package Vendor .....	88
Gambar 3. 21 System Flow Pengelolaan Profile Vendor .....	89
Gambar 3. 22 System Flow Konfirmasi Order .....	90
Gambar 3. 23 System Flow Jadwal Vendor.....	91
Gambar 3. 24 System Flow Register Customer .....	93
Gambar 3. 25 System Flow Pengelolaan Profile Customer.....	94
Gambar 3. 26 System Flow Pencarian Vendor .....	95
Gambar 3. 27 System Flow Customize Order Vendor .....	96
Gambar 3. 28 System Flow Generate dan Pemesanan Paket.....	98
Gambar 3. 29 System Flow Histori Transaksi .....	99
Gambar 3. 30 System Flow Konfirmasi Pembayaran Customer .....	100
Gambar 3. 31 System Flow Forgot Password Pengguna .....	101
Gambar 3. 32 System Flow Withdraw.....	102
Gambar 3. 33 System Flow Pembuatan Laporan Periodik.....	103
Gambar 3. 34 System Flow Rating & Review.....	104
Gambar 3. 35 Context Diagram Aplikasi Penentuan Vendor.....	105
Gambar 3. 36 Diagram Berjenjang Aplikasi Penentuan Vendor Pernikahan .....	106
Gambar 3. 37 Data Flow Diagram Level 0 .....	107
Gambar 3. 38 DFD Level 1 Input Data Master.....	108
Gambar 3. 39 DFD Level 1 Pendaftaran Vendor Baru.....	109
Gambar 3. 40 DFD Level 1 Pendaftaran Customer Baru .....	109
Gambar 3. 41 DFD Level 1 Transaksi .....	110

Gambar 3. 42 DFD Level 1 Pengelolaan User.....	111
Gambar 3. 43 DFD Level 1 Pembuatan Laporan.....	112
Gambar 3. 44 DFD Level 2 Transaksi .....	113
Gambar 3. 45 CDM Aplikasi Penentuan Vendor Pernikahan.....	115
Gambar 3. 46 PDM Aplikasi Penentuan Vendor Pernikahan .....	117
Gambar 3. 47 Desain Halaman Login Pengguna .....	127
Gambar 3. 48 Desain Halaman Pengelolaan Master Kategori.....	127
Gambar 3. 49 Desain Halaman Pengelolaan Master Nilai.....	128
Gambar 3. 50 Desain Halaman Pengelolaan Master Sub Kriteria .....	129
Gambar 3. 51 Desain Halaman Pengelolaan Master Bank .....	129
Gambar 3. 52 Desain Halaman Konfirmasi Pembayaran Pada Admin .....	130
Gambar 3. 53 Desain Halaman Histori Konfirmasi Pembayaran .....	131
Gambar 3. 54 Desain Halaman Legality Verification.....	131
Gambar 3. 55 Desain Halaman Dashboard Admin.....	132
Gambar 3. 56 Desain Halaman Registrasi Vendor .....	132
Gambar 3. 57 Desain Halaman Legality Vendor .....	133
Gambar 3. 58 Desain Halaman Dashboard Vendor .....	134
Gambar 3. 59 Desain Halaman Pengelolaan Kategori Vendor.....	135
Gambar 3. 60 Desain Halaman Pengelolaan Data Paket Vendor .....	136
Gambar 3. 61 Desain Halaman Pengelolaan Profile Vendor.....	137
Gambar 3. 62 Desain Halaman Konfirmasi Order.....	137
Gambar 3. 63 Desain Halaman Jadwal Vendor .....	138
Gambar 3. 64 Desain Halaman Registrasi Customer.....	138

Gambar 3. 65 Desain Halaman Pengelolaan Profile Customer .....	139
Gambar 3. 66 Desain Halaman Pencarian Vendor.....	140
Gambar 3. 67 Desain Halaman Customize Order.....	141
Gambar 3. 68 Desain Halaman Cart .....	141
Gambar 3. 69 Desain Halaman Generate Paket dan Pemesanan Paket .....	142
Gambar 3. 70 Desain Halaman Histori Transaksi.....	143
Gambar 3. 71 Desain Halaman Konfirmasi Pembayaran Pada Customer.....	143
Gambar 3. 72 Desain Halaman Forgot Password .....	144
Gambar 3. 73 Desain Halaman Withdraw .....	144
Gambar 3. 74 Desain Halaman History Withdraw .....	145
Gambar 4. 1 Halaman Login Pengguna .....	167
Gambar 4. 2 Pesan Login Error.....	167
Gambar 4. 3 Implementasi Halaman Pengelolaan Master Kategori.....	168
Gambar 4. 4 Notifikasi Add New Category Success .....	168
Gambar 4. 5 Notifikasi Update Category Success.....	169
Gambar 4. 6 Konfirmasi Hapus Data Kategori.....	169
Gambar 4. 7 Notifikasi data kategori berhasil dihapus .....	169
Gambar 4. 8 Implementasi Halaman Pengelolaan Master Nilai .....	170
Gambar 4. 9 Notifikasi Add Nilai Success .....	171
Gambar 4. 10 Notifikasi Update Master Nilai Success.....	171
Gambar 4. 11 Konfirmasi hapus data nilai.....	171
Gambar 4. 12 Notifikasi berhasil hapus data nilai .....	171
Gambar 4. 13 Implementasi Halaman Pengelolaan Master Sub Kriteria .....	172

Gambar 4. 14 Notifikasi Add New Sub Kriteria Success .....	173
Gambar 4. 15 Notifikasi Update Sub Kriteria Success .....	173
Gambar 4. 16 Konfirmasi Hapus Data Sub Kriteria .....	173
Gambar 4. 17 Notifikasi hapus data sukses .....	174
Gambar 4. 18 Implementasi Halaman Pengelolaan Master Bank.....	174
Gambar 4. 19 Notifikasi Add New Bank Success .....	175
Gambar 4. 20 Notifikasi Update Master Bank Success .....	175
Gambar 4. 21 Konfirmasi hapus data bank.....	175
Gambar 4. 22 Notifikasi berhasil hapus data bank .....	176
Gambar 4. 23 Implementasi Halaman Konfirmasi Pembayaran Pada Admin....	176
Gambar 4. 24 Implementasi Menampilkan Bukti Transfer .....	177
Gambar 4. 25 Konfirmasi Penerimaan Pembayaran.....	177
Gambar 4. 26 Notifikasi Kofirmasi Penerimaan Pembayaran.....	178
Gambar 4. 27 Konfirmasi Reject Pembayaran.....	178
Gambar 4. 28 Notifikasi Reject Pembayaran.....	178
Gambar 4. 29 Implementasi Halaman Histori Pembayaran Pada Admin.....	179
Gambar 4. 30 Implementasi Halaman Legality Verification.....	179
Gambar 4. 31 Implementasi View NPWP Legality Verification.....	180
Gambar 4. 32 Implementasi View KTP Legality Verification .....	180
Gambar 4. 33 Konfirmasi Legality Verification .....	180
Gambar 4. 34 Notifikasi Konfirmasi Legality Verification.....	181
Gambar 4. 35 Konfirmasi Reject Legality Verification.....	181
Gambar 4. 36 Notifikasi Legality Verification .....	181

Gambar 4. 37 Implementasi Halaman Dashboard Pada Admin .....	182
Gambar 4. 38 Implementasi Halaman Laporan Customer.....	182
Gambar 4. 39 Implementasi Halaman Cetak Laporan Customer .....	183
Gambar 4. 40 Implementasi Halaman Laporan Vendor .....	183
Gambar 4. 41 Implementasi Halaman Cetak Laporan Vendor .....	184
Gambar 4. 42 Implementasi Halaman Laporan Transaksi.....	184
Gambar 4. 43 Implementasi Halaman Cetak Laporan Transaksi .....	185
Gambar 4. 44 Implementasi Konfirmasi Penarikan Pada Admin.....	185
Gambar 4. 45 Konfirmasi Penarikan.....	186
Gambar 4. 46 Notifikasi Penarikan Berhasil Dikonfirmasi .....	186
Gambar 4. 47 Konfirmasi Reject Penarikan .....	186
Gambar 4. 48 Notifikasi Penarikan Berhasil di Reject .....	186
Gambar 4. 49 Implementasi Halaman Registrasi Vendor.....	187
Gambar 4. 50 Implementasi Halaman Legality Vendor .....	188
Gambar 4. 51 Implementasi Halaman Dashboard Pada Vendor 1.....	188
Gambar 4. 52 Implementasi Halaman Dashboard 2 .....	189
Gambar 4. 53 Implementasi Halaman Pengelolaan Kategori oleh Vendor .....	189
Gambar 4. 54 Implementasi Halaman List Kategori Vendor .....	190
Gambar 4. 55 Notifikasi Add New Category Vendor.....	190
Gambar 4. 56 Konfirmasi Hapus Category Vendor.....	190
Gambar 4. 57 Notifikasi Hapus Category Vendor Success .....	190
Gambar 4. 58 Implementasi Halaman Pengelolaan Data Paket Vendor.....	191
Gambar 4. 59 Implementasi Halaman List Paket Vendor.....	191



Gambar 4. 60 Notifikasi Add New Package Success .....	191
Gambar 4. 61 Notifikasi Update Package Success .....	192
Gambar 4. 62 Konfirmasi Hapus Package .....	192
Gambar 4. 63 Notifikasi Package Berhasil Dihapus.....	192
Gambar 4. 64 Implementasi Halaman Pengelolaan Profile Vendor .....	193
Gambar 4. 65 Implementasi Halaman Konfirmasi Order Pada Vendor .....	193
Gambar 4. 66 Konfirmasi Order .....	194
Gambar 4. 67 Notifikasi Konfirmasi Order .....	194
Gambar 4. 68 Konfirmasi Reject Order .....	194
Gambar 4. 69 Notifikasi Order Berhasil di <i>Reject</i> .....	194
Gambar 4. 70 Implementasi Halaman Jadwal Vendor.....	195
Gambar 4. 71 Implementasi Halaman Registrasi Customer .....	196
Gambar 4. 72 Notifikasi Registrasi Sukses.....	196
Gambar 4. 73 Implementasi Halaman Pengelolaan Profile Customer.....	197
Gambar 4. 74 Implementasi Halaman Pencarian Vendor.....	198
Gambar 4. 75 Implementasi Halaman Customize Order .....	199
Gambar 4. 76 Implementasi Halaman Review .....	199
Gambar 4. 77 Implementasi Halaman Pemberian Rating/Ulasan.....	200
Gambar 4. 78 Implementasi Halaman Cart.....	200
Gambar 4. 79 Implementasi Halaman Form Data Order .....	201
Gambar 4. 80 Implementasi Halaman Generate Paket 1 .....	201
Gambar 4. 81 Implementasi Halaman Generate Paket 2 .....	202
Gambar 4. 82 Implementasi Halaman Hasil Generate Paket.....	202

Gambar 4. 83 Implementasi Halaman Detail Hasil <i>Generate</i> Paket.....	203
Gambar 4. 84 Implementasi Halaman Pemesanan Paket Pada Customer .....	203
Gambar 4. 85 Implementasi Halaman Histori Transaksi.....	204
Gambar 4. 86 Implementasi Halaman Detail Pemesanan.....	204
Gambar 4. 87 Implementasi Konfirmasi Pembayaran Pada <i>Customer</i> .....	205
Gambar 4. 88 Notifikasi Order Pada Customer .....	206
Gambar 4. 89 Implementasi Halaman Notifikasi Pada Customer .....	206
Gambar 4. 90 Implementasi Halaman Recover Password.....	207
Gambar 4. 91 Implementasi Halaman <i>Reset Password</i> .....	207
Gambar 4. 92 Implementasi Halaman Histori Withdraw.....	208
Gambar 4. 93 Implementasi Halaman Withdraw.....	208



## DAFTAR TABEL

	Halaman
Tabel 3. 1 Nilai Alternatif.....	31
Tabel 3. 2 Tabel Kebutuhan Pengguna Admin Sistem .....	31
Tabel 3. 3 Tabel Kebutuhan Pengguna Vendor .....	32
Tabel 3. 4 Tabel Kebutuhan Pengguna Customer.....	33
Tabel 3. 5 Kebutuhan Fungsi Dashboard Pada Admin.....	35
Tabel 3. 6 Kebutuhan Fungsi Pengelolaan Data Master Kategori Pada Admin ...	35
Tabel 3. 7 Kebutuhan Fungsi Pengelolaan Data Master Sub Kriteria Pada Admin.....	37
Tabel 3. 8 Kebutuhan Fungsi Pengelolaan Data Master Nilai Pada Admin .....	40
Tabel 3. 9 Kebutuhan Fungsi Pengelolaan Data Master Bank .....	42
Tabel 3. 10 Kebutuhan Fungsi Confirmation Payment Pada Admin.....	44
Tabel 3. 11 Kebutuhan Fungsi Histori Pembayaran .....	45
Tabel 3. 12 Kebutuhan Fungsi Legality Verification.....	45
Tabel 3. 13 Kebutuhan Fungsi Confirmation Withdraw.....	46
Tabel 3. 14 Kebutuhan Fungsi Pembuatan Laporan Tahunan .....	47
Tabel 3. 15 Kebutuhan Fungsi Registrasi Vendor .....	48
Tabel 3. 16 Kebutuhan Fungsi Legality Vendor.....	49
Tabel 3. 17 Kebutuhan Fungsi Dashboard Pada Admin.....	50
Tabel 3. 18 Kebutuhan Fungsi Pengelolaan Data Master Kategori Vendor .....	50
Tabel 3. 19 Kebutuhan Fungsi Pengelolaan Data Master Paket Vendor .....	52
Tabel 3. 20 Kebutuhan Fungsi Pengelolaan Profile Vendor.....	55
Tabel 3. 21 Kebutuhan Fungsi Konfirmasi Order Pada Vendor.....	56
Tabel 3. 22 Kebutuhan Fungsi Withdraw .....	57

Tabel 3. 23 Kebutuhan Fungsi Jadwal Vendor .....	57
Tabel 3. 24 Kebutuhan Fungsi Pembuatan Laporan Tahunan .....	58
Tabel 3. 25 Kebutuhan Fungsi Registrasi Customer.....	58
Tabel 3. 26 Kebutuhan Fungsi Pengelolaan Profile Customer .....	59
Tabel 3. 27 Kebutuhan Fungsi Pencarian Vendor .....	60
Tabel 3. 28 Kebutuhan Fungsi Customize Order Vendor.....	61
Tabel 3. 29 Kebutuhan Fungsi Generate Paket dan Pemesanan Paket Pada Customer .....	62
Tabel 3. 30 Kebutuhan Fungsi Histori Transaksi Pada Customer .....	63
Tabel 3. 31 Kebutuhan Fungsi Konfirmasi Pembayaran .....	64
Tabel 3. 32 Kebutuhan Fungsi Rating dan Review.....	65
Tabel 3. 33 Kebutuhan Fungsi Withdraw .....	65
Tabel 3. 34 Tabel Analisis Kebutuhan Non Fungsional .....	66
Tabel 3. 35 Tabel User .....	118
Tabel 3. 36 Tabel Vendor.....	118
Tabel 3. 37 Tabel Category .....	119
Tabel 3. 38 Tabel Category Vendor.....	120
Tabel 3. 39 Sub Kriteria .....	120
Tabel 3. 40 Tabel Nilai.....	121
Tabel 3. 41 Tabel Paket Vendor.....	121
Tabel 3. 42 Tabel Customer .....	122
Tabel 3. 43 Tabel Transaksi .....	122
Tabel 3. 44 Tabel Detail Transaksi .....	123
Tabel 3. 45 Tabel Rating.....	123

Tabel 3. 46 Tabel Bank .....	124
Tabel 3. 47 Tabel Pembayaran.....	124
Tabel 3. 48 Tabel Withdraw.....	125
Tabel 3. 49 Tabel Buku Besar.....	125
Tabel 3. 50 Tabel Jadwal Vendor .....	126
Tabel 3. 51 Tabel Notifikasi.....	126
Tabel 3. 52 Desain Uji Coba Form Login.....	145
Tabel 3. 53 Desain Uji Coba Form Master Kategori .....	146
Tabel 3. 54 Desain Uji Coba Form Master Nilai .....	147
Tabel 3. 55 Desain Uji Coba Form Sub Kriteria.....	148
Tabel 3. 56 Desain Uji Coba Form Master Bank.....	148
Tabel 3. 57 Desain Uji Coba Form Konfirmasi Pembayaran .....	149
Tabel 3. 58 Desain Uji Coba History Confirmation .....	150
Tabel 3. 59 Desain Uji Coba Legality Verification .....	151
Tabel 3. 60 Desain Uji Coba Dashboard.....	152
Tabel 3. 61 Desain Uji Coba Laporan Customer pada Admin .....	152
Tabel 3. 62 Desain Uji Coba Laporan Vendor pada Admin.....	153
Tabel 3. 63 Desain Uji Coba Laporan Transaksi pada Admin .....	153
Tabel 3. 64 Desain Uji Coba Withdraw Pada Admin .....	154
Tabel 3. 65 Desain Uji Coba Form Registrasi Vendor .....	155
Tabel 3. 66 Desain Uji Coba Form Legality Vendor.....	155
Tabel 3. 67 Desain Uji Coba Dashboard.....	155
Tabel 3. 68 Desain Uji Coba Form Master Kategori Vendor .....	156

Tabel 3. 69 Desain Uji Coba Form Master Paket Vendor .....	157
Tabel 3. 70 Desain Uji Coba Form Profile Vendor .....	157
Tabel 3. 71 Desain Uji Coba Form Konfirmasi Order.....	158
Tabel 3. 72 Desain Uji Coba Jadwal Vendor.....	159
Tabel 3. 73 Desain Uji Coba Form Registrasi Customer.....	159
Tabel 3. 74 Desain Uji Coba Form Profile Customer.....	159
Tabel 3. 75 Desain Uji Coba Pencarian Vendor .....	160
Tabel 3. 76 Desain Uji Coba Customize Order.....	160
Tabel 3. 77 Desain Uji Coba Form Generate Paket dan Pemesanan Paket .....	161
Tabel 3. 78 Desain Uji Coba Form Histori Transaksi.....	162
Tabel 3. 79 Desain Uji Coba Form Konfirmasi Pembayaran Pada Customer ....	162
Tabel 3. 80 Desain Uji Coba Notifikasi .....	162
Tabel 3. 81 Desain Uji Coba Form Forgot Password .....	163
Tabel 3. 82 Desain Uji Coba Withdraw .....	163
Tabel 4. 1 Hasil Uji Coba Form Login .....	209
Tabel 4. 2 Hasil Uji Coba Form Master Kategori.....	210
Tabel 4. 3 Hasil Uji Coba Form Master Nilai.....	211
Tabel 4. 4 Hasil Uji Coba Form Sub Kriteria .....	212
Tabel 4. 5 Hasil Uji Coba Form Master Bank .....	212
Tabel 4. 6 Hasil Uji Coba Form Konfirmasi Pembayaran.....	213
Tabel 4. 7 Hasil Uji Coba History Confirmation.....	214
Tabel 4. 8 Hasil Uji Coba Legality Verification.....	215
Tabel 4. 9 Hasil Uji Coba Dashboard .....	216

Tabel 4. 10 Hasil Uji Coba Laporan Customer pada Admin .....	216
Tabel 4. 11 Hasil Uji Coba Laporan Vendor pada Admin.....	217
Tabel 4. 12 Hasil Uji Coba Laporan Transaksi pada Admin .....	217
Tabel 4.13 Hasil Uji Coba Withdraw Pada Admin.....	218
Tabel 4. 14 Hasil Uji Coba Form Registrasi Vendor .....	219
Tabel 4. 15 Hasil Uji Coba Form Legality Vendor.....	219
Tabel 4. 16 Hasil Uji Coba Dashboard .....	220
Tabel 4. 17 Hasil Uji Coba Form Master Kategori Vendor.....	220
Tabel 4. 18 Hasil Uji Coba Form Master Paket Vendor .....	221
Tabel 4. 19 Hasil Uji Coba Form Profile Vendor .....	222
Tabel 4. 20 Hasil Uji Coba Form Konfirmasi Order .....	222
Tabel 4. 21 Hasil Uji Coba Form Jadwal Vendor.....	223
Tabel 4. 22 Hasil Uji Coba Form Registrasi Customer .....	224
Tabel 4. 23 Hasil Uji Coba Form Profile Customer.....	224
Tabel 4. 24 Hasil Uji Coba Pencarian Vendor.....	225
Tabel 4. 25 Hasil Uji Coba Customize Order .....	225
Tabel 4. 26 Hasil Uji Coba Form Generate Paket dan Pemesanan Paket.....	226
Tabel 4. 27 Hasil Uji Coba Form Histori Transaksi .....	227
Tabel 4. 28 Hasil Uji Coba Form Konfirmasi Pembayaran Pada Customer.....	227
Tabel 4. 29 Hasil Uji Coba Notifikasi.....	228
Tabel 4. 30 Hasil Uji Coba <i>Form Forgot Password</i> .....	228
Tabel 4. 31 Hasil Uji Coba <i>Withdraw</i> .....	229

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Masalah

Intimate Production (IP) adalah salah satu startup di bidang *wedding Organizer* online yang ada di Surabaya. IP didirikan pada 07 Juli 2017 oleh Fachreza Ilham Dwi Cahyo, dengan kantor yang beralamat di Sidoarjo Indah Permai Blok B-1. Saat ini IP bergerak di bidang jasa / layanan yang membantu calon pengantin dan keluarganya merencanakan dan melaksanakan pesta pernikahan sesuai dengan yang diimpikan. IP telah menjalin kerja sama dengan Ninda *Make Up Artist* (MUA) untuk kategori (MUA) dan undangyuk.com untuk kategori invitation. Target *customer* yang ingin dicapai oleh IP yaitu calon pengantin yang berusia 20-50 tahun. Dari target *customer* yang sudah ditentukan, *customer* cukup susah ketika ingin menentukan *vendor* apa yang ingin digunakan karena semakin banyak pilihan yang beragam dari harga dan kualitas yang juga beragam. Untuk mencapai *customer* tersebut saat ini IP menggunakan sarana sosial media Instagram sebagai sarana promosi dan juga menggunakan aplikasi Whatsapp untuk sarana komunikasi secara langsung. Biaya yang dikeluarkan saat ini yaitu ketika *customer* telah setuju untuk menggunakan jasa IP, maka akan keluar biaya untuk persiapan acara seperti pembuatan rundown untuk seluruh panitia, dan lain-lain. Untuk pendapatan IP didapatkan dari persetujuan penggunaan jasa IP oleh *customer* sesuai kesepakatan.

Proses bisnis yang ada saat ini pada IP adalah calon pengantin (*customer*) menghubungi IP untuk berkonsultasi tentang rencana pernikahannya. Setelah setuju



maka *customer* akan melakukan *Down Payment* (DP) sebagai tanda persetujuan menggunakan jasa IP. Setelah itu IP akan membuat rundown awal berdasar hasil diskusi dengan *customer*, IP akan melakukan rapat untuk membahas konsep acara, *rundown* acara, dan *vendor* yang akan terlibat. Setelah semua persiapan tuntas, *customer* dapat melunasi semua pembayaran maksimal h-7 sebelum acara. Ketika acara, IP melakukan *controlling* ke semua *vendor* dan menjalankan susunan acara sesuai skenario yang telah direncanakan. Dan terakhir setelah acara selesai IP meminta *feedback* dari *customer*.

Proses bisnis yang ada saat ini dikembangkan sesuai visi yang dimiliki IP yaitu “Menjadi Perusahaan Yang Mampu Mewujudkan Rencana Acara *Customer* Dan Membantu *Vendor* Terhubung Dengan Calon *Customer*”. Untuk mencapai visi tersebut IP memiliki misi : 1) Memberikan sebuah kemudahan memilih *vendor* oleh *customer*, 2) Memberikan sebuah kemudahan perhitungan biaya acara, 3) Menjadi sebuah wadah untuk mengembangkan potensi *vendor*. IP dapat menjembatani pelaku usaha (*vendor*) dengan *Customer*, dengan bergabung bersama IP, *vendor* akan mendapat keuntungan peluang tambahan *order* dari *customer* yang menggunakan jasa IP. Tidak hanya *vendor*, *customer* juga akan mendapatkan keuntungan. *Customer* akan lebih mudah memilih *vendor* untuk pernikahan *customer* sesuai kriteria yang diinginkan *customer*. Dengan menggunakan sistem ini sehingga *customer* tidak direpotkan untuk memilih ataupun menghitung biaya pada setiap *vendor*.

Dengan visi yang diinginkan IP maka terjadi gap antara kondisi sekarang dan visi yang diharapkan. Hal-hal yang akan di selesaikan untuk mencapai harapan dari IP, yaitu : data *vendor* yang bergabung pada IP sementara hanya dua Ninda

Make Up Artist (MUA) dan Undangyuk.com (*Invitation Online*), sementara data *vendor* yang harus dimiliki ada beberapa kategori : *Venue, Catering, Decoration, Entertainment, MC, Photo and Videographer*. Sedangkan dengan menghubungkan antara *vendor* dan *customer* akan memberikan kemudahan *customer* untuk memilih *vendor* dengan cara menentukan kriteria sehingga *customer* tidak perlu kebingungan lagi untuk memilih *vendor* apa yang akan digunakan dengan pilihan *vendor* terbaik sesuai kriteria pilihan dan budget *customer*. Sistem akan memberikan pilihan paket *vendor* yang dapat digunakan oleh *Customer*. Selain itu *customer* dapat memilih *vendor* dari mana saja dan kapan saja tidak perlu lagi harus bertemu dengan IP untuk berkonsultasi. Sehingga dengan cara tersebut IP dapat memiliki keunggulan bisnis dibanding *Wedding Organizer* lain yang masih menggunakan cara tradisional dan juga IP dapat mengoptimalkan keuntungan yang didapat dari komisi yang telah disepakati dari setiap *vendor* yang tergabung pada IP yang dipilih oleh *customer*. Jika harapan dan kendala tidak diselesaikan maka IP akan kalah bersaing dengan *Wedding Organizer* yang sudah memiliki nama dan jam terbang melebihi IP.

Solusi yang akan dibuat adalah sebuah aplikasi berbasis *web* dengan menggunakan metode *Simple Multi Attribute Rating Technique Exploiting Ranks* (SMARTER) dan konsep *crowdsourcing*. Metode ini digunakan untuk menyusun rekomendasi paket pernikahan sesuai dengan bobot kriteria yang telah ditentukan oleh *customer*. Sedangkan konsep *crowdsourcing* digunakan untuk memudahkan *customer* memilih *vendor* pernikahan dalam satu tempat. Sehingga *customer* tidak perlu lagi kebingungan untuk memilih *vendor* mana yang terbaik untuk pernikahan *customer*. Adapun modul yang dibuat didalam tugas akhir ini yaitu pengelolaan

master, pengelolaan *dashbord*, cetak laporan (member *customer*, vendor, transaksi, dan pendapatan), *withdraw*, pengelolaan notifikasi, pendaftaran akun, pengelolaan *profile*, pengelolaan *password* (*change* dan *forgot password*), transaksi, kustomisasi paket pernikahan, *generate* paket, pembayaran, dan rating.

## 1.2 Perumusan Masalah

Berdasarkan latar belakang permasalahan di atas, maka rumusan masalahnya yaitu bagaimana merancang bangun aplikasi penentuan *vendor* pernikahan untuk usaha *Wedding Organizer* menggunakan metode SMARTER berbasis *web*.

## 1.3 Batasan Masalah

Berdasarkan perumusan masalah di atas, adapun batasan masalah pada penelitian ini adalah sebagai berikut:

1. *Vendor* yang dilibatkan meliputi *Venue, Catering, Decoration, Entertainment, MC, Photo and Videographer, Make up Artist*.
2. Menghasilkan maksimal 3 kombinasi pilihan paket pernikahan yang terbaik.
3. Menangani kriteria & sub kriteria secara dinamis.
4. Metode pembayaran yang digunakan adalah transfer *bank*.

## 1.4 Tujuan

Berdasarkan rumusan masalah di atas, maka tujuan dari penyusunan tugas akhir ini adalah menghasilkan aplikasi penentuan vendor pernikahan berbasis *web* sesuai kebutuhan calon pengantin menggunakan metode SMARTER.

## 1.5 Manfaat

Adapun manfaat dari sistem yang akan dibangun nantinya adalah:

1. Membantu *customer* dalam memilih paket pernikahan sesuai kriteria yang diinginkan.
2. Mengurangi kesalahan dalam perhitungan biaya paket pernikahan
3. Meningkatkan peluang vendor untuk mendapatkan pesanan dari *customer*.

## 1.6 Sistematika Penulisan

Di dalam penyusunan laporan tugas akhir ini secara sistematis diatur dan disusun dalam lima bab, yang masing-masing terdiri dari beberapa sub bab. Adapun urutan dari bab pertama sampai bab terakhir adalah sebagai berikut:

### **BAB I PENDAHULUAN**

Bab ini membahas tentang latar belakang masalah, perumusan masalah, batasan masalah, tujuan pembuatan sistem, manfaat bagi penggunaanya, serta sistematika penulisan laporan.

### **BAB II LANDASAN TEORI**

Bab ini membahas mengenai berbagai macam teori yang mendukung dalam pembuatan rancang bangun Aplikasi Penentuan Vendor Pernikahan Untuk Usaha *Wedding Organizer* Menggunakan Metode SMARTER Berbasis *Web*

### **BAB III ANALISA DAN PERANCANGAN SISTEM**

Bab ini membahas analisa dan perancangan sistem. Analisa berisi penjelasan dari timbulnya masalah beserta penyelesaiannya, sedangkan perancangan sistem berisi Blok Diagram, *System Flowchart*,

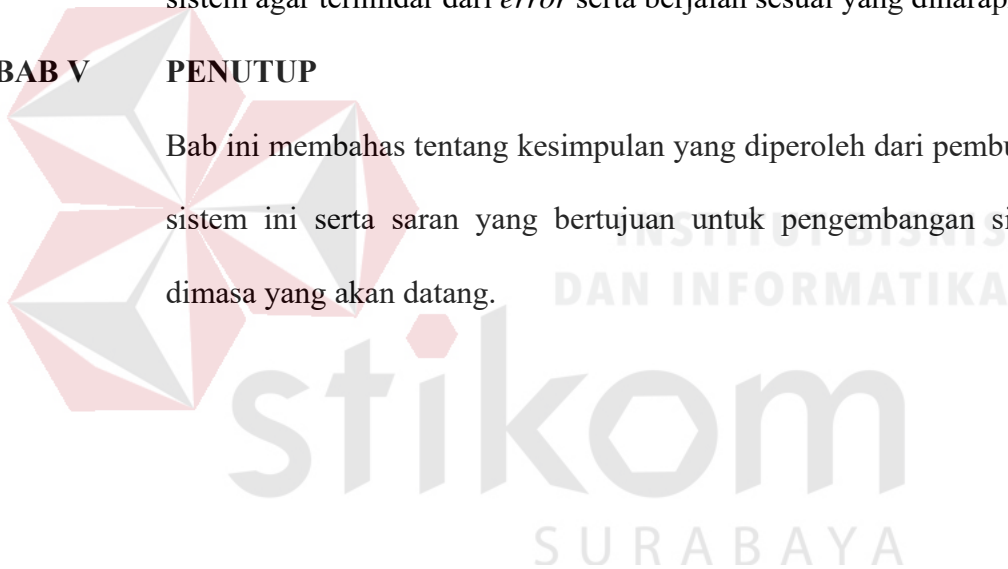
*Conceptual Data Model, Physical Data Model, Desain User Interface*  
dan Rancangan Pengujian dan Evaluasi Sistem

#### **BAB IV IMPLEMENTASI DAN EVALUASI SISTEM**

Bab ini membahas tentang kebutuhan sistem, pembuatan program, langkah-langkah implementasi, penggunaan sistem dalam proses bisnis serta pengujian untuk mengetahui apakah sistem yang dibangun dapat menyelesaikan permasalahan yang dihadapi sesuai dengan yang diharapkan. Sedangkan evaluasi sistem berisi validasi dan uji coba sistem agar terhindar dari *error* serta berjalan sesuai yang diharapkan.

#### **BAB V PENUTUP**

Bab ini membahas tentang kesimpulan yang diperoleh dari pembuatan sistem ini serta saran yang bertujuan untuk pengembangan sistem dimasa yang akan datang.



## BAB II

### LANDASAN TEORI

#### 2.1 Penelitian Terdahulu

Penelitian terdahulu ini menjadi salah satu acuan penulis dalam melakukan penelitian sehingga penulis dapat memperkaya teori yang digunakan dalam mengkaji penelitian yang dilakukan. Dari penelitian terdahulu, penulis tidak menemukan penelitian dengan judul yang sama seperti judul penelitian penulis. Namun penulis mengangkat beberapa penelitian sebagai referensi dalam memperkaya bahan kajian pada penelitian penulis. Berikut merupakan penelitian terdahulu berupa beberapa jurnal terkait dengan penelitian yang dilakukan penulis.

Tabel 2. 1 Penelitian Terdahulu 1

Nama Peneliti	Judul Penelitian	Hasil Penelitian
A. Priyolistiyanto, 2013.	Implementasi Metode <i>Simple Multi Attribute Rating Technique Exploiting Rank</i> (SMARTER) pada Sistem Pendukung Keputusan Sanksi Pelanggaran Tata Tertib Sekolah	Sistem Pendukung Keputusan SPTTS yang telah dibuat dapat digunakan untuk mendukung menentukan pemberian sanksi kepada siswa yang melakukan pelanggaran berdasarkan kriteria pelanggaran dan bobot pelanggaran.
Perbedaan: penelitian yang dilakukan (Priyolistiyanto, 2013). Di Implementasikan pada Sanksi Pelanggaran Tata Tertib Sekolah, sedangkan penulis melakukan Implementasi pada <i>wedding Organizer</i> .		

Tabel 2. 2 Penelitian Terdahulu 2

Nama Peneliti	Judul Penelitian	Hasil Penelitian
Robby Setiadi Guntoro, 2014	Pembuatan Sistem Pendukung Keputusan Untuk Pemilihan <i>Vendor</i> Acara Pernikahan Menggunakan Metode <i>Multi – Attribute Global Inference Of Quality</i> (MAGIQ)	Dengan aplikasi sistem pendukung keputusan ini pengguna bisa mendapatkan informasi mengenai <i>vendor</i> pernikahan dan fasilitasnya yang sesuai dengan keinginan mereka dengan lebih cepat apabila dibandingkan dengan pencarian secara maual dengan mendatangi pameran pernikahan maupun menghubungi kantor <i>vendor</i> yang bersangkutan. Aplikasi juga membantu mempermudah pengguna dalam mendapatkan informasi dan penawaran terbaru dari <i>vendor-vendor</i> pernikahan.
Perbedaan: penelitian yang dilakukan oleh (Guntoro, 2014) menggunakan metode <i>Multi – Attribute Global Inference Of Quality</i> (MAGIQ) dengan pengguna aplikasi adalah <i>customer</i> , sedangkan penulis menggunakan metode <i>Simple Multi Attribute Rating Technique Exploiting Ranks</i> (SMARTER) dengan pengguna adalah WO.		

Tabel 2. 3 Penelitian Terdahulu 3

Nama Peneliti	Judul Penelitian	Hasil Penelitian
Ganang Afif Rijazim, 2016	Pembangunan Modul Penawaran Pada Siapsiapnikah.com Dengan Metode <i>Extreme Programming</i> dan Konsep <i>Crowdsourcing</i>	<i>Website</i> dapat memenuhi kebutuhan penggunanya. Calon pengantin dapat mencari WO dan paket yang sesuai dengan biaya dan kualitas yang diinginkan pada satu <i>website</i> dan WO dapat memasarkan produknya pada satu <i>website</i> di internet. <i>Website</i> siapsiapnikah.com dapat dioperasikan baik pada dua <i>web browser</i> , yaitu

Nama Peneliti	Judul Penelitian	Hasil Penelitian
		<i>Mozilla Firefox</i> dan <i>Google Chrome</i> berdasarkan hasil penelitian ini.
Perbedaan: Pada penelitian (Rijazim, 2016) aplikasi menggunakan modul penawaran untuk paket pernikahan yang diajukan pada WO, dan WO akan membuatkan paket sesuai penawaran <i>customer</i> . Sedangkan pada penelitian ini pembuatan paket pernikahan dibuat oleh sistem yang terdiri dari paket yang dimiliki oleh vendor, paket akan disusun dan dikombinasi dengan paket antar kategori sesuai kriteria yang ditentukan <i>customer</i> .		

## 2.2 Konsep Sistem Informasi

Menurut Leitch dan Davis dalam Jogiyanto (2005) sistem informasi adalah suatu sistem di dalam organisasi yang menggabungkan kebutuhan pengelolaan transaksi harian, untuk mendukung operasi, yang bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan informasi yang dibutuhkan oleh pihak-pihak diluar manajemen. Sehingga dapat diartikan bahwa sistem informasi merupakan sekumpulan kegiatan mulai dari pengelolaan data transaksional hingga menjadi informasi yang berguna bagi manajemen maupun pihak-pihak diluar manajemen.

## 2.3 *Simple Multi Attribute Rating Technique Exploiting Ranks* (SMARTER)

Metode pembobotan *Simple Multi-Attribute Rating Technique Exploiting Ranks* (SMARTER) merupakan modifikasi yang diusulkan oleh Edwards dan Baron untuk metode pembobotan yang ditemukan sebelumnya, yaitu metode SMART (*Simple MultiAttribute Rating Technique*). Kedua metode ini digunakan untuk menentukan bobot dari setiap kriteria (Priyolistiyanto, 2013).

Metode SMARTER, merupakan metode pembobotan yang menggunakan range antara 0 sampai 1, sehingga mempermudah perhitungan dan perbandingan



nilai pada masing-masing alternatif. Bobot dihitung dengan menggunakan rumus *pembobotan Rank Order Centroid* (ROC). ROC ini didasarkan pada tingkat kepentingan atau prioritas dari kriteria.

Pembobotan ROC didapat dengan prosedur matematika sederhana dari prioritas. Ide dasarnya dapat diilustrasikan dengan 2 atribut, A dan B. Jika A ranking pertama, maka bobotnya harus berada diantara 0,5 dan 1 sehingga titik tengah interval 0,75 diambil sebagai bobot perkiraan, yang merupakan dasar dari sebuah prinsip komitmen minimum. Seperti bobot B akan menjadi 0,25 (merupakan titik tengah antara 0 dan 0,5). Prosedur ini dapat dirumuskan sebagai berikut (jika ada k kriteria):

$$W_1 \geq W_2 \geq W_3 \geq \dots \dots \geq W_k$$

$$W_1 = \frac{(1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{k})}{k}$$

$$W_2 = \frac{(0 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{k})}{k}$$

$$W_3 = \frac{(0 + 0 + \frac{1}{3} + \dots + \frac{1}{k})}{k}$$

Secara umum, jika k adalah jumlah kriteria, maka bobot dari kriteria ke k, adalah:

$$W_k = \left(\frac{1}{k}\right) \cdot \sum_{i=k}^k \left(\frac{1}{i}\right)$$

Keterangan ;

$W_k$  = Bobot Kriteria ke k

K = Jumlah Kriteria

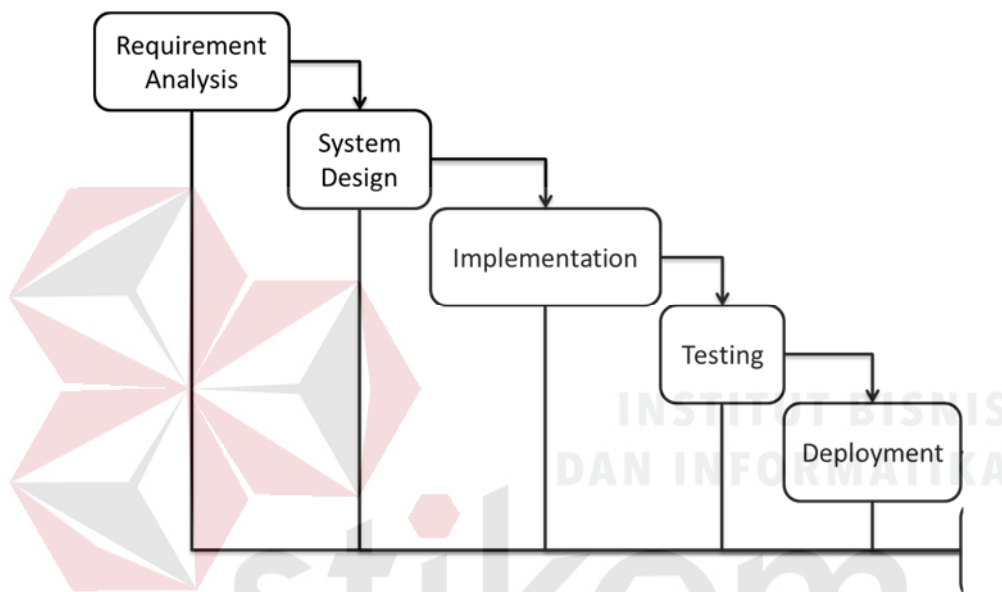
Langkah-langkah metode SMARTER adalah sebagai berikut:

- 1) Identifikasi permasalahan, sehingga dapat merumuskan keputusan yang akan diambil.
- 2) Tentukan alternatif, kriteria dan sub kriteria.
- 3) Memberikan peringkat untuk setiap kriteria dan sub kriteria.
- 4) Menghitung bobot menggunakan pembobotan ROC untuk setiap kriteria, hal ini bergantung pada peringkat yang telah diberikan pada langkah 3.
- 5) Menghitung bobot menggunakan pembobotan ROC untuk setiap sub kriteria, hal ini bergantung pada peringkat yang telah diberikan pada langkah 3.
- 6) Menghitung bobot akhir setiap kriteria, dengan mengalikan hasil langkah ke 4 dan hasil langkah ke 5.
- 7) Memberikan penilaian pada semua kriteria untuk setiap alternatif. Nilai diberikan dalam skala 0-100 dimana 0 sebagai nilai minimum dan 100 sebagai nilai maksimum.
- 8) Menghitung utilitas terhadap setiap alternatif dengan menggunakan persamaan maksimumkan  $\sum_{j=1}^k W_j U_{ij}, \forall i = 1 \text{ sampai } n$
- 9) Memutuskan, jika hanya satu alternatif yang akan dipilih, maka akan dipilih alternatif dengan utilitas paling besar.

#### 2.4 *System Development Life Cycle (SDLC)*

*System Development Life Cycle (SDLC)* merupakan suatu gambaran tahapan utama serta langkah-langkah pengembangan suatu sistem (Jogiyanto, 2005). Saat ini terdapat banyak model SDLC, namun yang akan digunakan dalam penelitian ini akan menggunakan model *waterfall*.

Model *waterfall* diciptakan pertama kali oleh William Riyce pada tahun 1970. Metode ini mendefinisikan bahwa sebuah proses hidup sistem memiliki proses yang linear dan sekuensial. Sehingga prinsip utama dari metode *waterfall* adalah tiap tahapan tidak dapat dilakukan jika tahap sebelumnya belum selesai. Pada model SDLC terdiri dari enam tahapan, namun yang akan digunakan hanya empat tahapan saja.



Gambar 2. 1 SDLC Metode *Waterfall*

SDLC metode *waterfall* memiliki beberapa tahapan yang terdiri dari:

1. *Requirement Analysis*

Pada fase ini semua kebutuhan sistem diidentifikasi. Daftar kebutuhan sistem dapat diperoleh melalui *survei* dan analisis proses bisnis pada organisasi yang diteliti.

## 2. *System Design*

Setelah melakukan identifikasi semua kebutuhan sistem, selanjutnya *developer* membuat desain sistem. Hal ini diperlukan untuk menganalisis kebutuhan *hardware and system requirements*.

## 3. *Implementation*

Pada tahapan ini, *programmer* melakukan *coding* program berdasarkan hasil desain sistem.

## 4. *Testing*

Setelah program selesai dibuat, selanjutnya dilakukan *testing*. Hal ini dilakukan agar dapat mengetahui atau mengidentifikasi *bug* atau *error* pada program.

## 5. *Deployment*

Fase *deployment* merupakan fase akhir dari pengembangan suatu sistem. Seluruh *bug* atau *error* telah di atasi dan program siap dipublikasikan.

### 2.5 *Website*

*Website* merupakan fasilitas internet yang menghubungkan dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada *website* disebut dengan *web page* dan *link* dalam *website* memungkinkan pengguna bisa berpindah dari satu *page* ke *page* lain (*hyper text*), baik diantara *page* yang disimpan dalam *server* yang sama maupun *server* diseluruh dunia. *Pages* diakses dan dibaca melalui *browser* seperti Netscape Navigator, Internet Explorer, Mozilla Firefox, Google Chrome dan aplikasi *browser* lainnya (Hakim & Musalini, 2004).

Jenis-jenis *web* berdasarkan sifat atau *style*-nya:

1. *Website* Dinamis, merupakan sebuah *website* yang menyediakan *content* atau isi yang selalu berubah-ubah setiap saat. Bahasa pemrograman yang digunakan antara lain PHP, ASP, .NET dan memanfaatkan *database* MySQL atau MS SQL. Misalnya *website* [www.artikel-it.com](http://www.artikel-it.com), [www.detik.com](http://www.detik.com), [www.technomobile.co.cc](http://www.technomobile.co.cc), [www.polinpdg.ac.id](http://www.polinpdg.ac.id) dan lain-lain.
2. *Website* Statis, merupakan *website* yang *content*-nya sangat jarang diubah. Bahasa pemrograman yang digunakan adalah HTML dan belum memanfaatkan *database*. Misalnya: *web profile* organisasi, dan lain-lain.

Menurut Nugraha Dian (2015) Kriteria *website* yang baik meliputi :

1. Desain Visual, yaitu tampilan yang menarik sehingga pengguna merasa nyaman ketika menggunakan *website*.
2. *Loading Time*, kecepatan loading pada *website* harus cepat, sehingga pengguna tidak menunggu terlalu lama.
3. *Interactivity*, *website* memiliki kemampuan untuk responsif terhadap pengguna, sehingga pengguna nyaman didalam menggunakan *website*.
4. *Content*, konten yang ada didalam *website* harus teratur dan tertata secara rapih, dari tata tulis hingga penggunaan kata, sehingga pengguna dapat memahami konten yang disajikan.
5. Aksesibilitas dan Kompabilitas, *website* yang telah dibangun harus mudah diakses pengguna, *website* harus mampu menyesuaikan dengan *platform* yang digunakan oleh pengguna seperti (*mobile, web*) dan juga *website* mampu menyesuaikan dengan *browser* pengguna seperti (*chrome, mozilla, edge dll*).

6. *Functionality*, Setiap komponen situs *web* harus bekerja dengan cepat dan benar adalah salah satu faktor penting sebuah situs web. Hindari kesalahan dalam penulisan baik tata bahasa, tanda baca ataupun salah ejaan. Komponen yang rusak akan membuat pengunjung frustrasi dan kecewa dengan perusahaan/instansi. *Functionality* ini akan melibatkan programmer dengan *script-scriptnya* misal PHP, ASP, Java, CGI dsb, untuk menciptakan sebuah *website* yang dinamis, interaktif dan ‘hidup’ yang dapat mengajak pengunjung berkomunikasi secara langsung (Dian, Kriteria Website yang Baik, 2013). Seberapa baik sebuah *website* bekerja dari aspek teknologikalnya.

7. *Usability*, kemampuan *website* agar mudah digunakan oleh pengguna dan sederhana dalam pengoperasiannya.

## 2.6 MySQL

*My Structured Query Language* (MySQL) adalah *software* basis data yang tergolong tipe *database* server dan bersifat *open source* (Kadir, 2009). *Database Server* merupakan jenis basis data yang secara aktif memantau permintaan akses terhadap data. MySQL memiliki beberapa keuntungan yaitu bersifat gratis yang dapat dengan mudah diunduh di internet, selain itu basis data MySQL bersifat *multiplatform* yang dapat diakses berbagai sistem operasi. *Database* MySQL dapat digunakan dalam pembuatan aplikasi berbasis *web* dan *desktop*. Penelitian ini menggunakan MySQL sebagai *database* untuk penyimpanan data.

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan

mudah secara otomatis. Keandalan suatu sistem database (DBMS) dapat diketahui dari cara kerja *optimizer*-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh *user* maupun program-program aplikasinya. Sebagai *database server*, MySQL dapat dikatakan lebih unggul dibandingkan *database server* lainnya dalam *query* data. Hal ini terbukti untuk *query* yang dilakukan oleh *single user*, kecepatan *query* MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan *Interbase* (Ratna, 2014).

MySQL memiliki beberapa keistimewaan menurut Ratna (2014), antara lain:

1. *Portabilitas*. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. *Open Source*. MySQL didistribusikan secara *open source*, dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.
3. “*Multiuser*”. MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. “*Performance tuning*”. MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.
5. *Jenis Kolom*. MySQL memiliki tipe kolom yang sangat kompleks, seperti *signed / unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.
6. *Perintah dan Fungsi*. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *Select* dan *Where* dalam perintah (*query*).

7. Keamanan. MySQL memiliki beberapa lapisan sekuritas seperti level *subnetmask*, nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, *Unix socket* (UNIX), atau *Named Pipes* (NT).
10. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.
11. Antar Muka. MySQL memiliki interface (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan (*tool*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.

## 2.7 PHP (*Hypertext Preprocessor*)

PHP (*Hypertext Preprocessor*) adalah sebuah bahasa *scripting* yang terpasang pada HTML (*Hypertext Markup Language*). PHP menurut Sutarman (2003), merupakan salah satu bahasa *server-side* yang didesain khusus untuk aplikasi *web*. PHP merupakan sebuah *server-side embedded script language*,



dimana perintah-perintah yang diberikan akan sepenuhnya dijalankan oleh *server* tetapi disertakan pada halaman HTML biasa.

PHP memiliki keunggulan dimana bahasa ini mendukung banyak basis data seperti MySQL, MS-SQL, Ms. Access, PostgreSQL, mSQL, dan Oracle.

Kelebihan lain dari PHP menurut Sutarman (2003), yaitu:

- a. PHP mudah dibuat dan kecepatan akses tinggi.
- b. PHP dapat berjalan dalam web server yang berbeda dan dalam sistem operasi yang berbeda.
- c. PHP merupakan bahasa yang *embedded* yang artinya dapat diletakkan dalam *tag* HTML.
- d. PHP dapat digunakan secara gratis.

Pada penelitian ini bahasa pemrograman yang digunakan yaitu PHP sebagai back end untuk pemrosesan data.

## 2.8 *Framework Codeigniter*

*CodeIgniter* adalah sebuah *web framework* yang dikembangkan oleh Rick Ellis dari Ellis Lab. *CodeIgniter* dirancang untuk menjadi sebuah *web framework* yang ringan dan mudah untuk digunakan. Bahkan pengakuan dari Rasmus Lerdorf, pencipta bahasa pemrograman PHP, mengatakan bahwa *CodeIgniter* merupakan *web framework* mudah dan handal (Septian, 2014).

*Codeigniter* menerapkan lingkungan pengembangan dengan metode MVC (*Model-View-Controller*). MVC memisahkan antara logika proses bisnis dengan *template* atau halaman depan. Dengan MVC, kode program menjadi terstruktur, terorganisir dan akan mudah dikembangkan kembali oleh pengembang awalnya.

Fitur-fitur standar yang sudah tersedia di *CodeIgniter* seperti *Paging*, *Validation*, *Routing* dan masih banyak lagi (Julisman, 2014).

*CodeIgniter* mempunyai berbagai keunggulan seperti yang disebutkan di dokumentasinya (Septian, 2014):

1. *Free*, karena berada dibawah lisensi open source mirip Apache/BSD, kita dapat melakukan apapun dengan *CodeIgniter*. Lisensi lengkapnya dapat dilihat di halaman dokumentasi
2. *Light Weight*, sistem inti *CodeIgniter* memerlukan *library* yang sedikit. Berbeda sekali dengan *framework* lainnya yang membutuhkan banyak sumber daya tambahan. *Library* tambahan akan digunakan ketika *request* secara dinamis, membuat sistem yang dibangun menjadi efisien dan cukup cepat.
3. *Fast*, menurut dokumentasi, performa yang dimiliki *CodeIgniter* terbukti cepat setelah dibandingkan dengan *web framework* lainnya
4. Menggunakan kaidah M-V-C, Dengan menggunakan *Model-View-Controller*, kita dapat memisahkan bagian *logic* dan *presentation* dari aplikasi yang kita bangun. Hal ini tentu sangat cocok dan bagus untuk proyek yang memfokuskan desainer fokus pada *template* file dan programmer fokus pada pembangunan *logic* dari aplikasi yang dibangun
5. Menghasilkan URL yang bersih, URL yang dihasilkan oleh *CodeIgniter* bersih dan ramah terhadap mesin pencari. *CodeIgniter* menggunakan pendekatan *segment-based* dibandingkan dengan *query* string yang biasa digunakan oleh programmer yang tidak menggunakan *web framework*.
6. *Packs a Punch*, *CodeIgniter* hadir dengan berbagai *library* yang akan membantu tugas – tugas di pengembangan web yang sudah umum dan sering

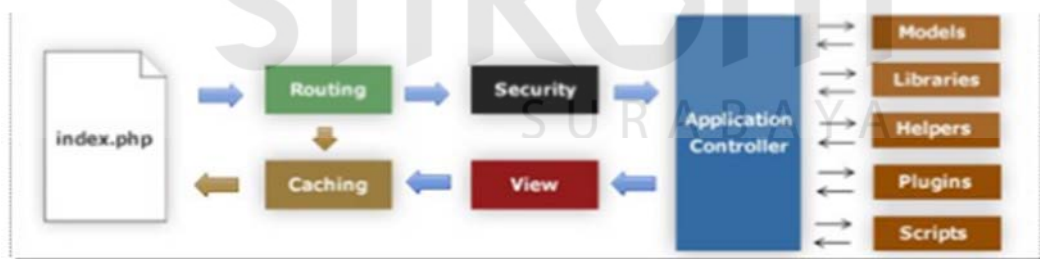
dilakukan seperti mengakses database, mengirim email, validasi data dari form, mengelola *session*, manipulasi gambar, bekerja dengan XML-RPC dan masih banyak lagi

7. *Extensible*, kita dapat menambahkan *library* atau *helper* yang kita ciptakan sendiri ke dalam *CodeIgniter*. Selain itu kita dapat juga menambahkan fitur lewat *class extension* atau *system hooks*.

*Thoroughly Documented*, hampir semua fitur, *library*, dan *helper* yang ada di *CodeIgniter* telah terdokumentasi dengan lengkap dan tersusun dengan baik. Ketika mendapatkan unduhan *CodeIgniter*, dokumentasinya sudah tersedia dan siap digunakan.

### A. Cara Kerja *CodeIgniter*

Untuk melengkapi pemahaman mengenai *CodeIgniter*, berikut terdapat sebuah diagram yang menjelaskan bagaimana *CodeIgniter* bekerja:



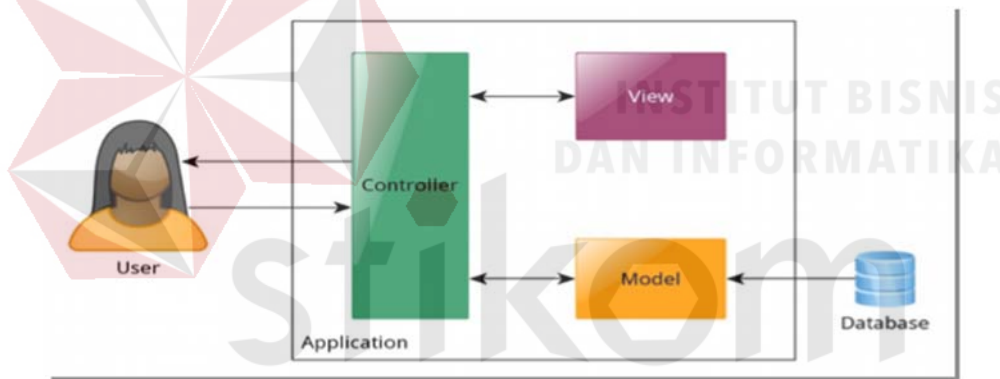
Gambar 2. 2 Cara Kerja *CodeIgniter*

Berikut adalah penjelasan cara kerja *CodeIgniter*:

1. *Index.php* bertindak sebagai *controller* terdepan, dan menginisialisasi resource yang diperlukan untuk menjalankan *CodeIgniter*
2. *Router* memeriksa *HTTP request* untuk menentukan apa yang harus dikerjakan

3. Jika *cache file* ada, maka akan ditampilkan langsung, dengan melewati eksekusi normal sistem
4. Sebelum memuat *controller*, *HTTP request* akan memeriksa apa yang disubmit *user* dan memfilternya untuk keamanan
5. *Controller* memuat model, *core libraries*, *plugin*, *helper*, dan *resource* lainnya untuk memproses permintaan tertentu
6. *View* ditampilkan di *browser* sesuai proses yang dikerjakan *controller*. Jika *caching* dijalankan, *view* akan di-*cache* terlebih dahulu agar dapat ditampilkan di *request* selanjutnya

## B. Cara Kerja MVC



Gambar 2. 3 Cara Kerja MVC

*CodeIgniter* menggunakan pendekatan *Model-View-Controller*, yang bertujuan untuk memisahkan logika dan presentasi. Konsep ini mempunyai keunggulan dimana desainer dapat bekerja pada *template file*, sehingga redundansi kode presentasi dapat diperkecil. Berikut adalah konsep *Model View-Controller* yang diterapkan di *CodeIgniter*:

- 1 Model menggambarkan struktur data. Biasanya kelas model akan berisi fungsi yang digunakan untuk mengambil, menambah, dan memperbaharui informasi yang ada di *database*.
- 2 *View* adalah informasi yang diperlihatkan kepada *user*. *View* adalah halaman web yang terdiri dari HTML, CSS dan Javascript, tapi pada *Code Igniter*, *view* dapat juga sebagai potongan halaman seperti *header* atau *footer*. Bahkan dapat juga halaman RSS atau tipe halaman lainnya.
- 3 *Controller* adalah perantara Model, *View*, dan *resource* lainya yang dibutuhkan untuk menangani HTTP *request* dan menghasilkan halaman *web*. Tapi pada *CodeIgniter*, *developer* juga dapat mengabaikan pemakaian Model dan cukup menggunakan *Controller* dan *View*.

### C. Istilah Pada *CodeIgniter*

Berikut adalah daftar istilah yang akan hadir selama pelatihan:

1. *Model*, *class* PHP yang dirancang untuk bekerja dengan informasi dari *database*
2. *Controller*, inti aplikasi yang menentukan penanganan HTTP *request*
3. *View*, halaman web seperti *header*, *footer*, *sidebar* dan lainnya yang ditanamkan di halaman *web* yang lainnya. *View* tidak pernah dipanggil secara langsung. *View* harus dipanggil dari *controller*
4. *Library*, *class* yang berisi fungsi – fungsi untuk penyelesaian kasus tetentu
5. *Helper*, pembantu tugas untuk kategori tertentu yang terdiri dari kumpulan *function*
6. *Driver*, *library* khusus yang mempunyai *class* induk dan beberapa *class* turunan yang dapat digunakan untuk kasus tertentu

Pada penelitian ini, peneliti menggunakan *framework CodeIgniter* yang digunakan untuk mempercepat proses *development* aplikasi penentuan vendor pernikahan.

## 2.9 *Crowdsourcing*

Menurut Howe (2009), *crowdsourcing* adalah suatu kegiatan atau aktifitas yang dikerjakan oleh perusahaan atau institusi dengan mengambil salah satu fungsi pekerjaan atau tugas yang seharusnya dilakukan oleh karyawannya menjadi disebarluaskan secara terbuka dan bebas untuk orang banyak atau kerumunan yang terkoneksi dengan jaringan komputer, dalam hal ini Internet (Howe, 2009). *Crowdsourcing* merupakan konsep yang melibatkan secara luas dan tidak terbatas dengan latar belakang, warga negara, pendidikan, ataupun pekerjaan.

### **Kelebihan dan Kelemahan *Crowdsourcing*** (Calvin, 2015):

1. Sebagai sebuah konsep, *crowdsourcing* memiliki kelebihan dan juga kekurangan. Kelebihannya diantaranya dapat meningkatkan produktivitas tanpa menambah tenaga kerja. Dengan menggunakan *internet*, masyarakat, dan konsumen, dapat terlibat secara langsung dalam usaha tersebut. Misalnya, memberikan respon positif saat membeli produk atau menggunakan jasa secara *online*. Dengan demikian, konsumen juga membantu memasarkan sebuah produk.
2. Namun *crowdsourcing* juga memiliki kelemahan. Diantaranya adalah dalam sisi keamanan. Besarnya partisipasi masyarakat membuat kontrol terhadap segala informasi menjadi tidak terkontrol. Sehingga semakin terbukanya dan

cepatnya akses menyebabkan penyebaran konten yang tidak bertanggung jawab.

## 2.10 *Testing*

Menurut Romeo (2003), testing adalah proses pemantapan kepercayaan akan kinerja program atau sistem sebagaimana yang diharapkan. *Testing Software* adalah proses pengoperasian *software* dalam suatu kondisi yang dikendalikan untuk verifikasi, mendeteksi *error* dan validasi. Verifikasi adalah pengecekan atau pengetesan entitas-entitas, termasuk *software*, untuk pemenuhan dan konsistensi dengan melakukan evaluasi hasil terhadap kebutuhan yang telah ditetapkan. Validasi adalah melihat kebenaran sistem apakah proses yang telah dituliskan sudah sesuai dengan apa yang dibutuhkan oleh pengguna.

Deteksi *error* adalah testing yang berorientasi untuk membuat kesalahan secara intensif, untuk menentukan apakah suatu hal tersebut tidak terjadi. *Test case* merupakan suatu tes yang dilakukan berdasarkan pada suatu inisialisasi, masukan, kondisi ataupun 52 hasil yang telah ditentukan sebelumnya. Adapun kegunaan dari *test case* ini adalah sebagai berikut:

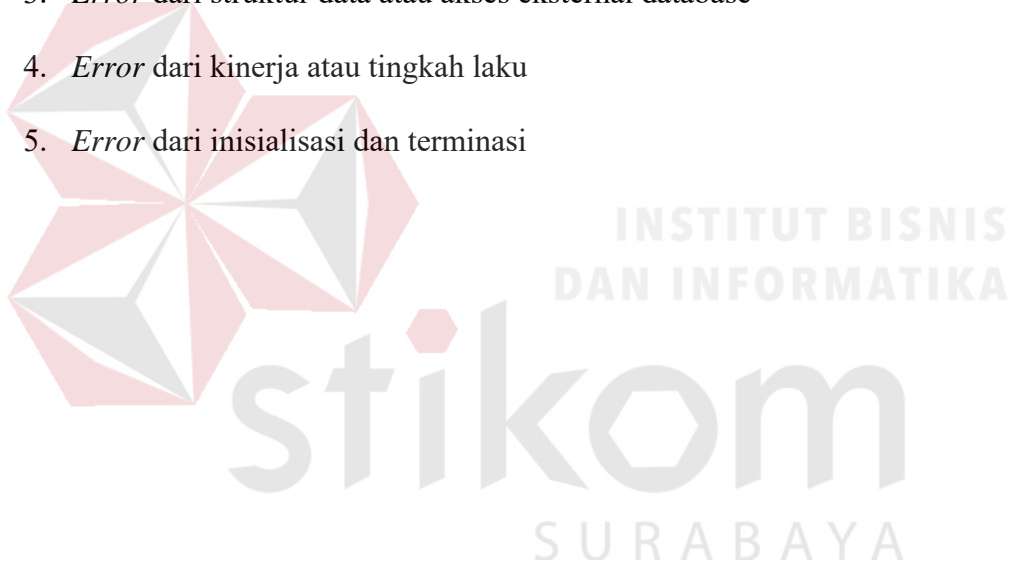
1. Untuk melakukan testing kesesuaian suatu komponen terhadap desain *White box Testing*.
2. Untuk melakukan testing kesesuaian suatu komponen terhadap spesifikasi *Black box Testing*.

### 2.10.1 *Black Box Testing*

*Black Box Testing* biasa disebut sebagai *behavioral testing*, *specification-based testing*, *input/output testing* atau *functional testing* dimana pengujian *black*

*box testing* ini dapat dilakukan tanpa pengetahuan detail struktur internal sistem atau komponen yang dites (Romeo, 2003). Penggunaan *black-box testing* dapat melakukan perancangan perangkat lunak dengan menggunakan sekumpulan kondisi masukan yang dapat secara penuh memeriksa keseluruhan kebutuhan fungsional pada suatu program. Beberapa kategori *error* yang dapat diketahui dengan menggunakan *black-box testing* ialah sebagai berikut:

1. Fungsi yang hilang atau tidak benar
2. *Error* dari antar-muka
3. *Error* dari struktur data atau akses eksternal database
4. *Error* dari kinerja atau tingkah laku
5. *Error* dari inisialisasi dan terminasi





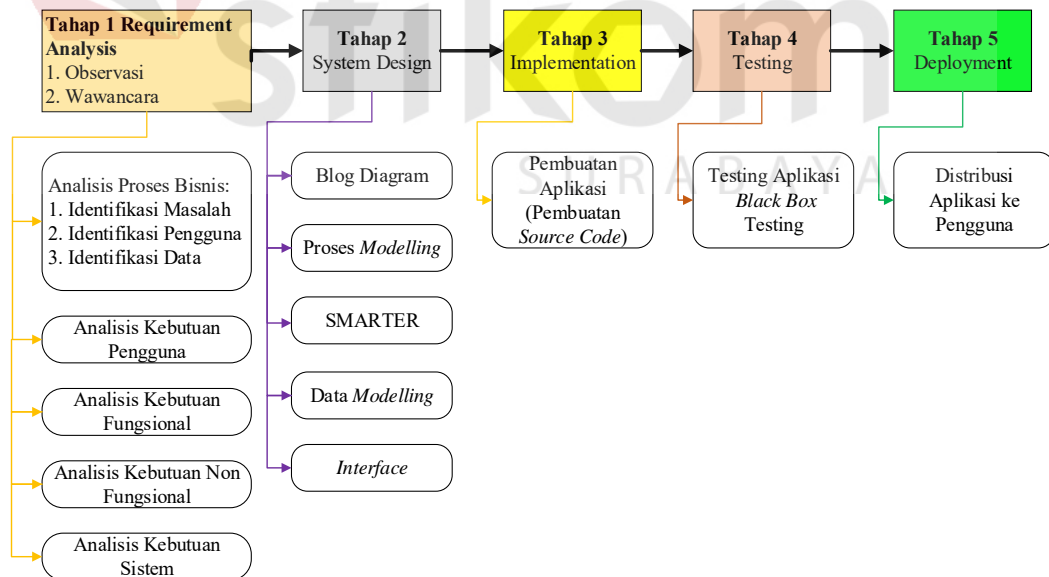
## BAB III

### ANALISIS DAN PERANCANGAN SISTEM

#### 3.1 Analisis Sistem

Penyusunan pada rancangan Aplikasi Penentuan *Vendor* Pernikahan Untuk Usaha *Wedding Organizer* Menggunakan Metode SMARTER Berbasis *Web* ini menggunakan *System Development Life Cycle (SDLC) Waterfall Model* yang digunakan untuk menggambarkan tahapan-tahapan melakukan perancangan sistem serta langkah-langkah yang harus dikerjakan pada setiap tahapan.

Metode penelitian dilakukan untuk melakukan pengerjaan Tugas Akhir sesuai dengan langkah-langkah yang diperlukan supaya pada pengerjaan Tugas Akhir dapat dilakukan secara terstruktur dan sistematis. Adapun model penelitian dapat dilihat pada gambar 3.1.



Gambar 3. 1 Diagram Model Penelitian

Tahapan pertama yang dilakukan adalah *Requirement Analysis*. Tahapan *Requirement Analysis* adalah tahapan pertama yang dilakukan dalam melakukan pembuatan Rancang Bangun Aplikasi Penentuan *Vendor* Pernikahan untuk Usaha *Wedding Organizer* Menggunakan Metode SMARTER Berbasis *Web*. Pada tahapan ini dilakukan pengumpulan data yang digunakan dalam pengembangan Aplikasi Penentuan *Vendor* Pernikahan Untuk Usaha *Wedding Organizer* Menggunakan Metode SMARTER Berbasis *Web* dari sumber yang terlibat. Tahap *Requirement Analysis* ini dilakukan dengan menggunakan dua cara untuk memperoleh data-data yang dibutuhkan yaitu:

1. Observasi

Pada kegiatan observasi ini dilakukan pengamatan secara langsung proses yang terjadi dalam melakukan penentuan *vendor* ketika membuat paket pada *Intimate Production*. Pengumpulan data dengan cara observasi ini dilakukan untuk mendapatkan informasi dan data yang berhubungan dalam penyelesaian masalah dan untuk mengetahui langkah-langkah yang dilakukan dalam melakukan penentuan *vendor*. Hasil observasi yang telah dilakukan yaitu IP dalam membuat paket pernikahan masih menggunakan cara yang manual dengan mencari atau menghubungi satu persatu *vendor* untuk melakukan kerjasama yang disesuaikan dengan kebutuhan *customer*.

2. Wawancara

Wawancara dilakukan dengan Pemilik *Intimate Production* pada bulan Agustus 2017. Wawancara ini bertujuan untuk mendapatkan informasi-informasi yang dibutuhkan. Hasil dari wawancara tersebut yaitu mengenai proses bisnis dimulai dari *customer* melakukan konsultasi dengan *Intimate*, memilih *vendor*

yang sesuai dengan kebutuhan *customer*, *customer* melakukan *Down Payment* untuk kepastian menggunakan jasa Intimate Production, menghubungi vendor terkait untuk memastikan ketersediaan jadwal, penyusunan *rundown* acara, pelunasan pembayaran dan eksekusi acara.

Pada tahap *Requirement Analysis* ini juga dilakukan terhadap beberapa analisis yang diperlukan dalam proses pengembangan sistem yaitu analisis proses bisnis, analisis kebutuhan pengguna, analisis kebutuhan fungsionalitas, analisis kebutuhan non fungsional, dan analisis sistem.

### 3.1.1 Analisis Proses Bisnis

Pada tahapan analisis proses bisnis dilakukan analisis terhadap proses bisnis yang terjadi pada proses penentuan *vendor* pernikahan. Tahapan yang dilakukan dalam analisis proses bisnis yaitu dengan melakukan identifikasi masalah, identifikasi pengguna dan identifikasi data.

#### a. Identifikasi Masalah

Identifikasi masalah didapatkan dari hasil wawancara dan observasi yang telah dilakukan. Berdasarkan hasil wawancara dan observasi, proses penentuan *vendor* pada Intimate Production terdapat dua sub proses yaitu sub proses pencatatan paket yang ada pada *vendor* dan sub proses pemilihan paket pada *vendor* sesuai budget. Sub proses pencatatan *vendor* dimulai dari pencatatan data *vendor*, lalu pencatatan paket yang dimiliki oleh *vendor*, kemudian mencatat fee yang diberikan.

Sub Proses selanjutnya yaitu sub proses pemilihan paket pada *vendor* sesuai budget. Pada sub proses ini dimulai dari mencatat *budget* dari *Customer*, lalu

memilih paket *vendor* yang sesuai dengan *budget Customer*. Selanjutnya dihitung total biaya dari paket yang terpilih.

b. Identifikasi Pengguna

Berdasarkan hasil wawancara dengan pemilik Intimate Production, pengguna yang akan terlibat pada Aplikasi Penentuan Vendor Pernikahan Untuk Usaha *Wedding Organizer* Menggunakan Metode Smarter Berbasis Web adalah Admin Sistem, *Vendor*, *Customer*.

c. Identifikasi Data

Aplikasi Penentuan *Vendor* Pernikahan Untuk Usaha *Wedding Organizer* Menggunakan Metode Smarter Berbasis Web membutuhkan data sebagai berikut: Data *Profile Vendor*, Data *Category Vendor*, Data *Package Vendor*, Data *Customer*, Data Transaksi *Customer*, Data *Rating Vendor*.

Berdasarkan analisis proses bisnis yang telah dilakukan, maka diberikan solusi yaitu membuat aplikasi dengan menggunakan metode SMARTER. Rumus SMARTER dapat diimplementasikan sebagai berikut:

Kriteria = *Mc*, *Catering*, *Dekorasi*, *Photography*, *Entertainment*

Sub Kriteria = Harga, rating, pengalaman

$W_k = (\text{Dekorasi} = 1, \text{Catering} = 2, \text{Mc} = 3, \text{Photography} = 4, \text{Entertainment} = 5)$

$K = 5$

$$W_1 = \frac{(1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5})}{5}$$

$$W_2 = \frac{(0 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5})}{5}$$

$$W_3 = \frac{(0 + 0 + \frac{1}{3} + \frac{1}{4} + \frac{1}{5})}{5}$$

$$W_4 = \frac{(0 + 0 + 0 + \frac{1}{4} + \frac{1}{5})}{5}$$

$$W_5 = \frac{(0 + 0 + 0 + 0 + \frac{1}{5})}{5}$$

**Kriteria:**

$$W_1 = 0,456$$

$$W_2 = 0,256$$

$$W_3 = 0,156$$

$$W_4 = 0,09$$

$$W_5 = 0,04$$

**Sub Kriteria:**

$$W_1 = \frac{(1 + \frac{1}{2} + \frac{1}{3})}{3}$$

$$W_2 = \frac{(0 + \frac{1}{2} + \frac{1}{3})}{3}$$

$$W_3 = \frac{(0 + 0 + \frac{1}{3})}{3}$$



$$W_1 = 0,611$$

$$W_2 = 0,278$$

$$W_3 = 0,111$$

Tabel 3. 1 Nilai Alternatif

Alternatif	Batas Bawah	Batas Atas	Nilai
A	0	50	50
A	50	100	80
B	0	50	60
B	50	100	85
C	0	50	40
C	50	100	75

### 3.1.2 Analisis kebutuhan pengguna

Analisis kebutuhan pengguna dilakukan untuk mengetahui kebutuhan-kebutuhan pengguna dalam proses penentuan vendor pernikahan dan kebutuhan informasi yang terlibat. Tiap pengguna memiliki fungsi yang berbeda sesuai dengan kebutuhan data dan informasi dari setiap pengguna. Untuk penerapan sistem pada aplikasi dapat dilihat dari kebutuhan pengguna dalam aplikasi penentuan vendor berikut:

#### A. Admin Sistem

Tabel 3. 2 Tabel Kebutuhan Pengguna Admin Sistem

Kebutuhan Fungsi	Kebutuhan Data	Kebutuhan Informasi
<i>Dashboard</i>	<ol style="list-style-type: none"> <li>1. Data Transaksi</li> <li>2. Data Vendor</li> <li>3. Data <i>Customer</i></li> <li>4. Data Kategori</li> <li>5. Data Kategori Vendor</li> <li>6. Data Paket Vendor</li> </ol>	<ol style="list-style-type: none"> <li>1. Laporan Member Vendor</li> <li>2. Laporan Member <i>Customer</i></li> <li>3. Laporan Transaksi</li> <li>4. Laporan Pendapatan</li> </ol>

<b>Kebutuhan Fungsi</b>	<b>Kebutuhan Data</b>	<b>Kebutuhan Informasi</b>
Pengelolaan Data Master Kategori	Data Kategori	5. Informasi Peringkat Vendor
Pengelolaan Data Master Sub Kriteria	1. Data Sub Kriteria 2. Data Kategori	
Pengelolaan Data Master Nilai	1. Data Sub Kriteria 2. Data Kategori 3. Data Nilai	
Pengelolaan Data Master <i>Bank</i>	Data Bank	
<i>Confirmation Payment</i>	1. Data Transaksi 2. Data Pembayaran 3. Data <i>Customer</i> 4. Data Bank 5. Data Notifikasi	
Histori Konfirmasi Pembayaran	1. Data Bank 2. Data Transaksi 3. Data Pembayaran	
<i>Legality Verification</i>	1. Data Vendor 2. Data Notifikasi	
<i>Confirmation Withdraw</i>	1. Data <i>withdraw</i> 2. Data <i>Bank</i> 3. Data <i>User</i> 4. Data Buku Besar 5. Data Notifikasi	
Pembuatan Laporan Periodik	1. Data Customer 2. Data Transaksi 3. Data Vendor 4. Data Kategori Vendor	

## B. Vendor

Tabel 3. 3 Tabel Kebutuhan Pengguna *Vendor*

<b>Kebutuhan Fungsi</b>	<b>Kebutuhan Data</b>	<b>Kebutuhan Informasi</b>
Registrasi Vendor	Data User	1. Laporan Pendapatan Perbulan 2. Laporan Transaksi 3. Informasi Total Pendapatan
<i>Legality Vendor</i>	Data Vendor	
<i>Dashboard</i>	1. Data Transaksi 2. Data Jadwal 3. Data Kategori Vendor	

<b>Kebutuhan Fungsi</b>	<b>Kebutuhan Data</b>	<b>Kebutuhan Informasi</b>
Pengelolaan Data Master Kategori Vendor	Data Kategori Vendor	4. Informasi Jadwal 5. Informasi Peringkat <i>Brand</i> /Kategori Vendor
Pengelolaan Data Master Paket Vendor	Data Paket Vendor	6. Informasi Notifikasi Order
Pengelolaan <i>Profile</i> Vendor	1. Data Vendor 2. Data <i>User</i>	7. Informasi Notifikasi Konfirmasi Penarikan
Konfirmasi <i>Order</i>	1. Data Transaksi 2. Data <i>Customer</i> 3. Data Kategori Vendor 4. Data Detail Transaksi 5. Data Notifikasi	
<i>Withdraw</i>	1. Data <i>Withdraw</i> 2. Data Buku Besar 3. Data <i>User</i>	
Jadwal Vendor	1. Data Vendor 2. Data Kategori 3. Data Jadwal	
Pembuatan Laporan Periodik	1. Data Transaksi 2. Data Kategori Vendor	

### C. Customer

Tabel 3. 4 Tabel Kebutuhan Pengguna *Customer*

<b>Kebutuhan Fungsi</b>	<b>Kebutuhan Data</b>	<b>Kebutuhan Informasi</b>
Registrasi <i>Customer</i>	<i>Data User</i>	1. Informasi <i>Vendor</i> 2. Informasi Paket yang didapat
Pengelolaan <i>Profile</i>	1. Data <i>Customer</i> 2. Data <i>User</i>	3. Informasi Transaksi
Pencarian Vendor	Data Kategori Vendor	4. Informasi Notifikasi Konfirmasi Pembayaran
<i>Customize Order</i> Vendor	1. Data Kategori Vendor 2. Data Paket Vendor 3. Data Transaksi 4. Data Detail Transaksi 5. Data <i>Rating</i> 6. Data Notifikasi	5. Informasi Notifikasi Konfirmasi Penarikan



Kebutuhan Fungsi	Kebutuhan Data	Kebutuhan Informasi
<i>Generate</i> Paket dan Pemesanan Paket	<ol style="list-style-type: none"> <li>1. Data Vendor</li> <li>2. Data Paket Vendor</li> <li>3. Data Kategori</li> <li>4. Data Sub Kriteria</li> <li>5. Data Kategori Vendor</li> <li>6. Data Nilai</li> <li>7. Data Rating</li> <li>8. Data Transaksi</li> <li>9. Data Detail Transaksi</li> <li>10. Data Jadwal</li> </ol>	6. Informasi Notifikasi Konfirmasi Order
Histori Transaksi	<ol style="list-style-type: none"> <li>1. Data Transaksi</li> <li>2. Data Pembayaran</li> <li>3. Data Detail Transaksi</li> </ol>	
Konfirmasi Pembayaran	<ol style="list-style-type: none"> <li>1. Data Transaksi</li> <li>2. Data Pembayaran</li> <li>3. Data <i>Bank</i></li> </ol>	
<i>Rating &amp; Review</i>	<ol style="list-style-type: none"> <li>1. Data <i>Category</i> Vendor</li> <li>2. Data <i>Customer</i></li> <li>3. Data <i>Rating</i></li> </ol>	
<i>Withdraw</i>	<ol style="list-style-type: none"> <li>1. Data <i>Withdraw</i></li> <li>2. Data Buku Besar</li> <li>3. Data <i>User</i></li> </ol>	

### 3.1.3 Analisis Kebutuhan Fungsional

Kebutuhan fungsionalitas merupakan suatu layanan sistem yang harus disediakan dan bagaimana sistem berinteraksi dengan pengguna sistem, seperti pengguna dapat melakukan apa saja di dalam sistem. Hasil Implementasi fungsi ini didapatkan dari hasil analisis kebutuhan pengguna.

## A. Analisis Kebutuhan Fungsional Admin

### 1. Fungsi *Dashboard*

Tabel 3. 5 Kebutuhan Fungsi *Dashboard* Pada Admin

<b>Nama Fungsi</b>	Fungsi <i>Dashboard</i>	
<b>Pengguna</b>	Admin	
<b>Deskripsi</b>	Fungsi ini merupakan kegiatan yang dilakukan untuk mengetahui dan mencetak laporan <i>vendor</i> , <i>customer</i> , transaksi, dan paket <i>vendor</i>	
<b>Kondisi Awal</b>	Data Transaksi, Data <i>Vendor</i> , Data <i>Customer</i> , Data Paket <i>Vendor</i>	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<i>Dashboard</i>	
	Pengguna menekan salah satu menu <i>dashboard</i> pada halaman <i>dashboard</i>	Sistem menampilkan <i>report dashboard</i> yang siap di <i>print</i> .
	Pengguna menekan tombol <i>print</i> jika ingin mencetak <i>report</i>	Sistem mencetak <i>report</i> yang dipilih
<b>Kondisi Akhir</b>	Fungsi dapat melihat dan mencetak <i>report</i> yang dipilih dari <i>dashboard</i>	

### 2. Fungsi Pengelolaan Data Master Kategori

Tabel 3. 6 Kebutuhan Fungsi Pengelolaan Data Master Kategori Pada Admin

<b>Nama Fungsi</b>	Fungsi Pengelolaan Data Master Kategori	
<b>Pengguna</b>	Admin	
<b>Deskripsi</b>	Fungsi ini merupakan kegiatan yang dilakukan untuk melakukan pengelolaan data master kategori.	
<b>Kondisi Awal</b>	Data Kategori	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<b>Menyimpan Data Master Kategori</b>	
	Pengguna memilih menu master kategori	Sistem akan menampilkan halaman <i>index</i> dari

		data halaman master kategori
	Kemudian pengguna memasukkan data master kedalam <i>form</i> tambah data dan menekan tombol tambah.	Sistem akan terlebih dahulu mengecek kelengkapan pengisian <i>form</i> dan kesesuaian pengisian <i>form</i> seperti <i>form</i> angka dan rentang data isian. Jika semua <i>field</i> terisi dan kesesuaian pengisian telah tepat, maka sistem akan menyimpan penambahan data dari <i>form</i> yang telah diisi kedalam tabel master. Sebaliknya, sistem akan menampilkan halaman <i>form</i> kembali dan peringatan sesuai dengan kesalahan pengisian yang telah dilakukan pengguna.
	<b>Mengubah Data Master Kategori</b>	
	Pengguna memilih data master yang ingin diubah.	Sistem akan menampilkan data master sesuai dengan pilihan pengguna pada <i>form</i> perubahan data master.
	Pengguna memasukkan perubahan data kedalam <i>form</i> perubahan data dan menekan tombol tambah.	Sistem akan terlebih dahulu mengecek kelengkapan pengisian <i>form</i> dan kesesuaian pengisian <i>form</i> seperti <i>form</i> angka dan rentang data isian. Jika semua <i>field</i> terisi dan kesesuaian pengisian

		telah tepat, maka sistem akan menyimpan penambahan data dari <i>form</i> yang telah diisi kedalam tabel master. Sebaliknya, sistem akan menampilkan halaman <i>form</i> kembali dan peringatan sesuai dengan kesalahan pengisian yang telah dilakukan pengguna.
	<b>Menghapus Data Master Kategori</b>	
	Pengguna memilih data master yang akan dihapus	Sistem akan menampilkan konfirmasi penghapusan data master untuk dikonfirmasi oleh pengguna.
	Pengguna melakukan konfirmasi penghapusan data dengan tombol hapus.	Sistem akan menghapus data pada tabel master dan menampilkan pesan berhasil menghapus data master.
<b>Kondisi Akhir</b>	Fungsi ini dapat melakukan pengelolaan data master kategori mulai dari penambahan, perubahan dan penghapusan data master.	

### 3. Fungsi Pengelolaan Data Master Sub Kriteria

Tabel 3. 7 Kebutuhan Fungsi Pengelolaan Data Master Sub Kriteria Pada Admin

<b>Nama Fungsi</b>	Fungsi Pengelolaan Data Master Sub Kriteria
<b>Pengguna</b>	Admin
<b>Deskripsi</b>	Fungsi ini merupakan kegiatan yang dilakukan untuk melakukan pengelolaan data master sub kriteria.

<b>Kondisi Awal</b>	Data Sub Kriteria, Data Kategori	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<b>Menyimpan Data Master Sub Kriteria</b>	
	Pengguna memilih menu master sub kriteria	Sistem akan menampilkan halaman <i>index</i> dari data halaman master sub kriteria
	Kemudian pengguna memasukkan data master kedalam <i>form</i> tambah data dan menekan tombol tambah.	Sistem akan terlebih dahulu mengecek kelengkapan pengisian <i>form</i> dan kesesuaian pengisian <i>form</i> seperti <i>form</i> angka dan rentang data isian. Jika semua <i>field</i> terisi dan kesesuaian pengisian telah tepat, maka sistem akan menyimpan penambahan data dari <i>form</i> yang telah diisi kedalam tabel master. Sebaliknya, sistem akan menampilkan halaman <i>form</i> kembali dan peringatan sesuai dengan kesalahan pengisian yang telah dilakukan pengguna.
	<b>Mengubah Data Master Sub Kriteria</b>	
	Pengguna memilih data master yang ingin diubah.	Sistem akan menampilkan data master sesuai dengan pilihan pengguna pada <i>form</i> perubahan data master.
	Pengguna memasukkan perubahan data kedalam <i>form</i>	Sistem akan terlebih dahulu mengecek kelengkapan pengisian <i>form</i> dan

	<p>pengubahan data dan menekan tombol tambah.</p>	<p>kesesuaian pengisian <i>form</i> seperti <i>form</i> angka dan rentang data isian. Jika semua <i>field</i> terisi dan kesesuaian pengisian telah tepat, maka sistem akan menyimpan penambahan data dari <i>form</i> yang telah diisi kedalam tabel master. Sebaliknya, sistem akan menampilkan halaman <i>form</i> kembali dan peringatan sesuai dengan kesalahan pengisian yang telah dilakukan pengguna.</p>
<b>Menghapus Data MasterSub Kriteria</b>		
	<p>Pengguna memilih data master yang akan dihapus</p>	<p>Sistem akan menampilkan konfirmasi penghapusan data master untuk dikonfirmasi oleh pengguna.</p>
	<p>Pengguna melakukan konfirmasi penghapusan data dengan tombol hapus.</p>	<p>Sistem akan menghapus data pada tabel master dan menampilkan pesan berhasil menghapus data master.</p>
<b>Kondisi Akhir</b>	<p>Fungsi ini dapat melakukan pengelolaan data master sub kriteria mulai dari penambahan, pengubahan dan penghapusan data master.</p>	

## 4. Fungsi Pengelolaan Data Master Nilai

Tabel 3. 8 Kebutuhan Fungsi Pengelolaan Data Master Nilai Pada Admin

<b>Nama Fungsi</b>	Fungsi Pengelolaan Data Master Nilai	
<b>Pengguna</b>	Admin	
<b>Deskripsi</b>	Fungsi ini merupakan kegiatan yang dilakukan untuk melakukan pengelolaan data master nilai	
<b>Kondisi Awal</b>	Data Sub Kriteria, Data Kategori, Data Nilai	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<b>Menyimpan Data Master Nilai</b>	
	Pengguna memilih menu master nilai	Sistem akan menampilkan halaman <i>index</i> dari data halaman master nilai
	Kemudian pengguna memasukkan data master kedalam <i>form</i> tambah data dan menekan tombol tambah.	Sistem akan terlebih dahulu mengecek kelengkapan pengisian <i>form</i> dan kesesuaian pengisian <i>form</i> seperti <i>form</i> angka dan rentang data isian. Jika semua <i>field</i> terisi dan kesesuaian pengisian telah tepat, maka sistem akan menyimpan penambahan data dari <i>form</i> yang telah diisi kedalam tabel master. Sebaliknya, sistem akan menampilkan halaman <i>form</i> kembali dan peringatan sesuai dengan kesalahan pengisian yang telah dilakukan pengguna.
<b>Mengubah Data Master Nilai</b>		

	Pengguna memilih data master yang ingin diubah.	Sistem akan menampilkan data master sesuai dengan pilihan pengguna pada <i>form</i> perubahan data master.
	Pengguna memasukkan perubahan data kedalam <i>form</i> perubahan data dan menekan tombol tambah.	Sistem akan terlebih dahulu mengecek kelengkapan pengisian <i>form</i> dan kesesuaian pengisian <i>form</i> seperti <i>form</i> angka dan rentang data isian. Jika semua <i>field</i> terisi dan kesesuaian pengisian telah tepat, maka sistem akan menyimpan penambahan data dari <i>form</i> yang telah diisi kedalam tabel master. Sebaliknya, sistem akan menampilkan halaman <i>form</i> kembali dan peringatan sesuai dengan kesalahan pengisian yang telah dilakukan pengguna.
<b>Menghapus Data Master Nilai</b>		
	Pengguna memilih data master yang akan dihapus	Sistem akan menampilkan konfirmasi penghapusan data master untuk dikonfirmasi oleh pengguna.
	Pengguna melakukan konfirmasi penghapusan data dengan tombol hapus.	Sistem akan menghapus data pada tabel master dan menampilkan pesan



		berhasil menghapus data master.
<b>Kondisi Akhir</b>	Fungsi ini dapat melakukan pengelolaan data master nilai mulai dari penambahan, perubahan dan penghapusan data master.	

#### 5. Fungsi Pengelolaan Data Master *Bank*

Tabel 3. 9 Kebutuhan Fungsi Pengelolaan Data Master *Bank*

<b>Nama Fungsi</b>	Fungsi Pengelolaan Data Master <i>Bank</i>	
<b>Pengguna</b>	Admin	
<b>Deskripsi</b>	Fungsi ini merupakan kegiatan yang dilakukan untuk menambah, mengubah dan menghapus data <i>bank</i>	
<b>Kondisi Awal</b>	Data <i>Bank</i>	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<b>Menyimpan Data Master <i>Bank</i></b>	
	Pengguna memilih menu <i>manage bank</i>	Sistem akan menampilkan halaman pengelolaan data master <i>bank</i> .
	Kemudian pengguna memasukkan data master kedalam <i>form</i> tambah data dan menekan tombol tambah.	Sistem akan terlebih dahulu mengecek kelengkapan pengisian <i>form</i> dan kesesuaian pengisian <i>form</i> seperti <i>form</i> angka dan rentang data isian. Jika semua <i>field</i> terisi dan kesesuaian pengisian telah tepat, maka sistem akan menyimpan penambahan data dari <i>form</i> yang telah diisi kedalam tabel master. Sebaliknya, sistem akan menampilkan halaman <i>form</i> kembali

		dan peringatan sesuai dengan kesalahan pengisian yang telah dilakukan pengguna.
<b>Mengubah Data Master Bank</b>		
	Pengguna memilih data master yang ingin diubah.	Sistem akan menampilkan data master sesuai dengan pilihan pengguna pada <i>form</i> perubahan data master.
	Pengguna memasukkan perubahan data kedalam <i>form</i> perubahan data dan menekan tombol tambah.	Sistem akan terlebih dahulu mengecek kelengkapan pengisian <i>form</i> dan kesesuaian pengisian <i>form</i> seperti <i>form</i> angka dan rentang data isian. Jika semua <i>field</i> terisi dan kesesuaian pengisian telah tepat, maka sistem akan menyimpan penambahan data dari <i>form</i> yang telah diisi kedalam tabel master. Sebaliknya, sistem akan menampilkan halaman <i>form</i> kembali dan peringatan sesuai dengan kesalahan pengisian yang telah dilakukan pengguna.
<b>Menghapus Data Master Bank</b>		
	Pengguna memilih data master yang akan dihapus	Sistem akan menampilkan konfirmasi penghapusan data master untuk dikonfirmasi oleh pengguna.

	Pengguna melakukan konfirmasi penghapusan data dengan tombol hapus.	Sistem akan menghapus data pada tabel master dan menampilkan pesan berhasil menghapus data master.
<b>Kondisi Akhir</b>	Fungsi ini dapat melakukan pengelolaan data master <i>bank</i> mulai dari penambahan, pengubahan dan penghapusan data master.	

## 6. Fungsi *Confirmation Payment*

Tabel 3. 10 Kebutuhan Fungsi *Confirmation Payment* Pada Admin

<b>Nama Fungsi</b>	Fungsi <i>Confirmation Payment</i>	
<b>Pengguna</b>	Admin	
<b>Deskripsi</b>	Fungsi ini merupakan kegiatan yang dilakukan untuk mengkonfirmasi pembayaran	
<b>Kondisi Awal</b>	Data Transaksi, Data Pembayaran, Data <i>Customer</i> , Data Bank	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<i>Confirmation Payment</i>	
	Pengguna memilih menu <i>confirmation payment</i>	Sistem menampilkan <i>list</i> pembayaran
	Pengguna melihat bukti transfer dengan menekan gambar bukti transfer	Sistem menampilkan gambar bukti transfer
Pengguna mengecek pembayaran pada internet banking. Jika ada, pengguna menekan tombol <i>confirmation</i> . Jika tidak ada maka pengguna menekan tombol <i>reject</i>	Sistem menampilkan modal/pop-up untuk memastikan konfirmasi (jika menekan tombol <i>confirmation</i> ). Sebaliknya sistem menampilkan modal/pop-up untuk memastikan <i>reject</i> atau tidak (jika menekan tombol	

		<i>reject</i> ). Setelah menekan tombol <i>confirmation</i> atau <i>reject</i> , maka sistem akan meng- <i>update</i> status pada tabel pembayaran.
<b>Kondisi Akhir</b>	Fungsi dapat melakukan konfirmasi pembayaran atau mereject dan sistem akan meng- <i>update</i> tabel dalam <i>database</i>	

## 7. Fungsi Histori Konfirmasi Pembayaran

Tabel 3. 11 Kebutuhan Fungsi Histori Pembayaran

<b>Nama Fungsi</b>	Fungsi Histori Konfirmasi Pembayaran	
<b>Pengguna</b>	Admin	
<b>Deskripsi</b>	Fungsi ini merupakan kegiatan yang dilakukan admin untuk melihat riwayat konfirmasi pembayaran dan statusnya.	
<b>Kondisi Awal</b>	Data Bank, Data Transaksi, Data Pembayaran	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<b>Histori Konfirmasi Pembayaran</b>	
	Pengguna menekan menu <i>confirmation payment</i>	Sistem menampilkan halaman <i>confirmation payment</i> dan <i>list</i> pembayaran
<b>Kondisi Akhir</b>	Fungsi ini dapat melihat riwayat pembayaran	

## 8. Fungsi *Legality Verification*

Tabel 3. 12 Kebutuhan Fungsi *Legality Verification*

<b>Nama Fungsi</b>	Fungsi <i>Legality Verification</i>
<b>Pengguna</b>	Admin
<b>Deskripsi</b>	Fungsi ini merupakan kegiatan yang dilakukan admin untuk mengecek data legal <i>vendor</i> yang mendaftar berupa nomor NPWP, nama badan usaha, alamat, KTP, nama pimpinan, NIK pimpinan, dan alamat pimpinan.

<b>Kondisi Awal</b>	Data <i>Vendor</i>	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<b>Legality Verification</b>	
	Pengguna menekan menu <i>legality verification</i>	Sistem menampilkan halaman <i>legality verification vendor</i>
	Pengguna mengecek data dengan <i>file</i> yang dilampirkan. Jika sesuai maka pengguna akan menekan tombol aksi konfirmasi. Sebaliknya jika tidak sesuai maka pengguna menekan tombol <i>reject</i>	Jika data sesuai, sistem akan <i>update</i> status <i>vendor</i> . Jika data tidak sesuai, maka sistem akan <i>update</i> status bahwa data di <i>reject</i> .
<b>Kondisi Akhir</b>	Fungsi dapat melakukan verifikasi data <i>legality vendor</i> atau <i>mereject</i> dan sistem akan <i>update</i> tabel dalam <i>database</i>	

9. Fungsi *Confirmation Withdraw*Tabel 3. 13 Kebutuhan Fungsi *Confirmation Withdraw*

<b>Nama Fungsi</b>	Fungsi <i>Confirmation Withdraw</i>	
<b>Pengguna</b>	Admin	
<b>Deskripsi</b>	Fungsi ini merupakan kegiatan yang dilakukan untuk melakukan konfirmasi setelah melakukan transfer dana pada pengguna	
<b>Kondisi Awal</b>	Data <i>Withdraw</i> , Data <i>User</i> , Data Bank	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<b>Confirmation Withdraw</b>	
	Pengguna menekan salah satu menu <i>dashboard</i> pada halaman <i>dashboard</i>	Sistem menampilkan halaman <i>confirmation withdraw</i> .
	Jika telah melakukan transfer, pengguna menekan tombol <i>confirmation</i> . Jika data penerima dana	Sistem menampilkan modal/pop-up untuk memastikan konfirmasi (jika menekan tombol

	tidak sesuai dengan data pada sistem pengguna menekan tombol <i>reject</i>	<i>confirmation</i> ). Sebaliknya sistem menampilkan modal/pop-up untuk memastikan <i>reject</i> atau tidak (jika menekan tombol <i>reject</i> ). Setelah menekan tombol <i>confirmation</i> atau <i>reject</i> , maka sistem akan meng- <i>update</i> status pada tabel <i>withdraw</i> dan menambahkan data kedalam tabel buku besar dengan tipe debit dan user yang bersangkutan.
<b>Kondisi Akhir</b>	Fungsi dapat melakukan konfirmasi <i>withdraw</i> atau <i>mereject</i> dan sistem akan meng- <i>update</i> tabel dalam <i>database</i>	

10. Fungsi Pembuatan Laporan Periodik (Member Vendor, Member *Customer*, Transaksi, Pendapatan)

Tabel 3. 14 Kebutuhan Fungsi Pembuatan Laporan Tahunan

<b>Nama Fungsi</b>	Fungsi Pembuatan Laporan Tahunan (Member Vendor, Member <i>Customer</i> , Transaksi, Pendapatan)	
<b>Pengguna</b>	Admin	
<b>Deskripsi</b>	Fungsi ini merupakan kegiatan membuat laporan periodik (Member Vendor, Member <i>Customer</i> , Transaksi, Pendapatan) yang dilakukan oleh admin dan kemudian di cetak	
<b>Kondisi Awal</b>	Data Vendor, Data <i>Customer</i> , Data Transaksi, Data Kategori Vendor	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<b>Halaman Pembuatan Laporan Tahunan</b>	

	Pengguna memilih jenis laporan (Member Vendor/ Member <i>Customer</i> / Transaksi/ Pendapatan/ Peringkat Vendor)	Sistem menampilkan halaman halaman laporan yang dipilih
	Pengguna memilih periode	Sistem menampilkan data laporan sesuai periode yang dipilih
	Pengguna menekan tombol cetak laporan	Sistem mencetak laporan
<b>Kondisi Akhir</b>	Fungsi ini memungkinkan pengguna untuk membuat laporan periodik	

## B. Analisis Kebutuhan Fungsional *Vendor*

### 1. Fungsi Registrasi *Vendor*

Tabel 3. 15 Kebutuhan Fungsi Registrasi *Vendor*

<b>Nama Fungsi</b>	Fungsi Registrasi <i>Vendor</i>	
<b>Pengguna</b>	Vendor	
<b>Deskripsi</b>	Fungsi ini merupakan kegiatan yang dilakukan untuk melakukan pendaftaran untuk bergabung menjadi vendor Intimate Production	
<b>Kondisi Awal</b>	Data user	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<b>Registrasi <i>Vendor</i></b>	
	Pengguna memasukkan nama, <i>email</i> dan <i>password</i> kemudian menekan tombol <i>Sign Up</i> pada halaman registrasi/ <i>create account</i> .	Sistem melakukan pengecekan teradap kesesuaian nama, <i>email</i> dan <i>password</i> serta mengecek ada atau tidaknya data dalam <i>database</i> . Jika data belum ada pada <i>database</i> dan data telah sesuai, maka sistem akan menampilkan notifikasi registrasi

		berhasil. Sebaliknya jika nama, <i>email</i> dan <i>password</i> tidak sesuai atau sudah terdaftar di database, maka sistem akan menampilkan notifikasi registrasi gagal.
<b>Kondisi Akhir</b>	Fungsi <i>registrasi/create account</i> memungkinkan pengguna untuk dapat menggunakan aplikasi.	

## 2. Fungsi *Legality Vendor*

Tabel 3. 16 Kebutuhan Fungsi *Legality Vendor*

<b>Nama Fungsi</b>	Fungsi <i>Legality Vendor</i>	
<b>Pengguna</b>	Vendor	
<b>Deskripsi</b>	Fungsi ini merupakan kegiatan yang dilakukan <i>vendor</i> untuk mengisi data legal <i>vendor</i> yang mendaftar berupa nomor NPWP, nama badan usaha, alamat, KTP, nama pimpinan, NIK pimpinan, dan alamat pimpinan setelah melakukan registrasi.	
<b>Kondisi Awal</b>	Data Vendor	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	Pengguna melakukan <i>login</i>	Sistem akan mengecek kelengkapan berkas <i>vendor</i> . Jika lengkap, maka sistem akan menampilkan halaman <i>vendor</i> . Jika belum lengkap, maka akan menampilkan <i>form legality</i> . Jika ditolak oleh admin, maka akan kembali ke halaman <i>login</i> dan



		diberi notifikasi berkas ditolak
<b>Kondisi Akhir</b>	Fungsi dapat menambah dan menyimpan data <i>legality</i> vendor	

### 3. Fungsi *Dashboard*

Tabel 3. 17 Kebutuhan Fungsi *Dashboard* Pada Admin

<b>Nama Fungsi</b>	Fungsi <i>Dashboard</i>	
<b>Pengguna</b>	Vendor	
<b>Deskripsi</b>	Fungsi ini merupakan kegiatan yang dilakukan untuk mengetahui dan mencetak laporan transaksi, dan pendapatan perbulan	
<b>Kondisi Awal</b>	Data Transaksi, Data Jadwal, Data Kategori Vendor	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<i>Dashboard</i>	
	Pengguna menekan salah satu menu <i>dashboard</i> pada halaman <i>dashboard</i>	Sistem menampilkan <i>report dashboard</i> yang siap di <i>print</i> .
	Pengguna menekan tombol <i>print</i> jika ingin mencetak <i>report</i>	Sistem mencetak <i>report</i> yang dipilih
<b>Kondisi Akhir</b>	Fungsi dapat melihat dan mencetak <i>report</i> yang dipilih dari <i>dashboard</i>	

### 4. Fungsi Pengelolaan Data Master Kategori Vendor

Tabel 3. 18 Kebutuhan Fungsi Pengelolaan Data Master Kategori Vendor

<b>Nama Fungsi</b>	Fungsi Pengelolaan Data Master Kategori Vendor	
<b>Pengguna</b>	Vendor	
<b>Deskripsi</b>	Fungsi ini merupakan kegiatan yang dilakukan untuk melakukan pengelolaan data master kategori vendor.	
<b>Kondisi Awal</b>	Data Kategori Vendor	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<b>Menyimpan Data Master</b>	

	Pengguna memilih menu master kategori vendor	Sistem akan menampilkan halaman <i>index</i> dari data halaman master kategori vendor
	Kemudian pengguna memasukkan data master kedalam <i>form</i> tambah data dan menekan tombol tambah.	Sistem akan terlebih dahulu mengecek kelengkapan pengisian <i>form</i> dan kesesuaian pengisian <i>form</i> seperti <i>form</i> angka dan rentang data isian. Jika semua <i>field</i> terisi dan kesesuaian pengisian telah tepat, maka sistem akan menyimpan penambahan data dari <i>form</i> yang telah diisi kedalam tabel master. Sebaliknya, sistem akan menampilkan halaman <i>form</i> kembali dan peringatan sesuai dengan kesalahan pengisian yang telah dilakukan pengguna.
<b>Mengubah Data Master</b>		
	Pengguna memilih data master yang ingin diubah.	Sistem akan menampilkan data master sesuai dengan pilihan pengguna pada <i>form</i> pengubahan data master.
	Pengguna memasukkan perubahan data kedalam <i>form</i> pengubahan data dan menekan tombol tambah.	Sistem akan terlebih dahulu mengecek kelengkapan pengisian <i>form</i> dan kesesuaian pengisian <i>form</i> seperti <i>form</i> angka dan rentang

		data isian. Jika semua <i>field</i> terisi dan kesesuaian pengisian telah tepat, maka sistem akan menyimpan penambahan data dari <i>form</i> yang telah diisi kedalam tabel master. Sebaliknya, sistem akan menampilkan halaman <i>form</i> kembali dan peringatan sesuai dengan kesalahan pengisian yang telah dilakukan pengguna.
	<b>Menghapus Data Master</b>	
	Pengguna memilih data master yang akan dihapus	Sistem akan menampilkan konfirmasi penghapusan data master untuk dikonfirmasi oleh pengguna.
	Pengguna melakukan konfirmasi penghapusan data dengan tombol hapus.	Sistem akan menghapus data pada tabel master dan menampilkan pesan berhasil menghapus data master.
<b>Kondisi Akhir</b>	Fungsi ini dapat melakukan pengelolaan data master kategori <i>vendor</i> mulai dari penambahan, perubahan dan penghapusan data master.	

#### 5. Fungsi Pengelolaan Data Master Paket Vendor

Tabel 3. 19 Kebutuhan Fungsi Pengelolaan Data Master Paket Vendor

<b>Nama Fungsi</b>	Fungsi Pengelolaan Data Master Paket Vendor
<b>Pengguna</b>	Vendor

<b>Deskripsi</b>	Fungsi ini merupakan kegiatan yang dilakukan untuk melakukan pengelolaan data master paket vendor.	
<b>Kondisi Awal</b>	Data Paket Vendor	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<b>Menyimpan Data Master</b>	
	Pengguna memilih menu master paket vendor	Sistem akan menampilkan halaman <i>index</i> dari data halaman master paket vendor\
	Kemudian pengguna memasukkan data master kedalam <i>form</i> tambah data dan menekan tombol tambah.	Sistem akan terlebih dahulu mengecek kelengkapan pengisian <i>form</i> dan kesesuaian pengisian <i>form</i> seperti <i>form</i> angka dan rentang data isian. Jika semua <i>field</i> terisi dan kesesuaian pengisian telah tepat, maka sistem \ akan menyimpan penambahan data dari <i>form</i> yang telah diisi kedalam tabel master. Sebaliknya, sistem akan menampilkan halaman <i>form</i> kembali dan peringatan sesuai dengan kesalahan pengisian yang telah dilakukan pengguna.
	<b>Mengubah Data Master</b>	
Pengguna memilih data master yang ingin diubah.	Sistem akan menampilkan data master sesuai dengan pilihan pengguna pada <i>form</i> pengubahan data master.	

	<p>Pengguna memasukkan perubahan data kedalam <i>form</i> pengubahan data dan menekan tombol tambah.</p>	<p>Sistem akan terlebih dahulu mengecek kelengkapan pengisian <i>form</i> dan kesesuaian pengisian <i>form</i> seperti <i>form</i> angka dan rentang data isian. Jika semua <i>field</i> terisi dan kesesuaian pengisian telah tepat, maka sistem akan menyimpan penambahan data dari <i>form</i> yang telah diisi kedalam tabel master. Sebaliknya, sistem akan menampilkan halaman <i>form</i> kembali dan peringatan sesuai dengan kesalahan pengisian yang telah dilakukan pengguna.</p>
<b>Menghapus Data Master</b>		
	<p>Pengguna memilih data master yang akan dihapus</p>	<p>Sistem akan menampilkan konfirmasi penghapusan data master untuk dikonfirmasi oleh pengguna.</p>
	<p>Pengguna melakukan konfirmasi penghapusan data dengan tombol hapus.</p>	<p>Sistem akan menghapus data pada tabel master dan menampilkan pesan berhasil menghapus data master.</p>
<p><b>Kondisi Akhir</b></p>	<p>Fungsi ini dapat melakukan pengelolaan data master paket <i>vendor</i> mulai dari penambahan, pengubahan dan penghapusan data master.</p>	

6. Fungsi Pengelolaan *Profile Vendor*Tabel 3. 20 Kebutuhan Fungsi Pengelolaan *Profile Vendor*

<b>Nama Fungsi</b>	Fungsi Pengelolaan Data Master <i>Profile Vendor</i>	
<b>Pengguna</b>	Vendor	
<b>Deskripsi</b>	Fungsi ini merupakan kegiatan yang dilakukan untuk melakukan pengelolaan data <i>profile vendor</i> .	
<b>Kondisi Awal</b>	Data Vendor, Data <i>User</i>	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<b>Menyimpan/Mengubah Data <i>Profile</i></b>	
	Pengguna memilih menu <i>view profile</i>	Sistem akan menampilkan halaman <i>index</i> dari data halaman <i>view profile</i>
Kemudian pengguna memasukkan data <i>profile</i> kedalam <i>form</i> dan menekan tombol tambah.	Sistem akan terlebih dahulu mengecek kelengkapan pengisian <i>form</i> dan kesesuaian pengisian <i>form</i> seperti <i>form</i> angka dan rentang data isian. Jika semua <i>field</i> terisi dan kesesuaian pengisian telah tepat, maka sistem akan menyimpan penambahan data dari <i>form</i> yang telah diisi kedalam tabel <i>vendor</i> . Sebaliknya, sistem akan menampilkan halaman <i>form</i> kembali dan peringatan sesuai dengan kesalahan pengisian yang telah dilakukan pengguna.	

<b>Kondisi Akhir</b>	Fungsi ini dapat menyimpan dan mengubah data <i>profile vendor</i>
----------------------	--

7. Fungsi Konfirmasi *Order*Tabel 3. 21 Kebutuhan Fungsi Konfirmasi *Order* Pada Vendor

<b>Nama Fungsi</b>	Fungsi Konfirmasi <i>Order</i>	
<b>Pengguna</b>	Vendor	
<b>Deskripsi</b>	Fungsi ini merupakan kegiatan yang dilakukan untuk mengkonfirmasi pemesanan	
<b>Kondisi Awal</b>	Data transaksi, Data <i>customer</i> , Data kategori vendor, Data detail transaksi	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<b>Konfirmasi <i>Order</i></b>	
	Pengguna memilih menu <i>confirmation order</i>	Sistem menampilkan <i>list</i> pemesanan
Pengguna menekan tombol <i>confirmation</i> . Jika tidak ada maka pengguna menekan tombol <i>reject</i>	Sistem menampilkan modal/pop-up untuk memastikan konfirmasi (jika menekan tombol <i>confirmation</i> ). Sebaliknya sistem menampilkan modal/pop-up untuk memastikan <i>reject</i> atau tidak (jika menekan tombol <i>reject</i> ). Setelah menekan tombol <i>confirmation</i> atau <i>reject</i> , maka sistem akan meng- <i>update</i> status pada tabel pemesanan.	
<b>Kondisi Akhir</b>	Fungsi dapat melakukan konfirmasi pemesanan atau <i>mereject</i> dan sistem akan meng- <i>update</i> tabel dalam <i>database</i>	

8. Fungsi *Withdraw*Tabel 3. 22 Kebutuhan Fungsi *Withdraw*

<b>Nama Fungsi</b>	Fungsi <i>Withdraw</i>	
<b>Pengguna</b>	Vendor, <i>Customer</i>	
<b>Deskripsi</b>	Fungsi ini merupakan kegiatan yang dilakukan untuk melakukan penarikan saldo	
<b>Kondisi Awal</b>	Data <i>Withdraw</i> , Data Buku Besar, Data User	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<b><i>Withdraw</i></b>	
	Pengguna memilih menu <i>withdraw</i>	Sistem menampilkan halaman <i>withdraw</i>
	Pengguna memilih <i>tab withdraw</i>	Sistem menampilkan <i>form withdraw</i>
	Pengguna mengisi <i>form withdraw</i> . Kemudian menekan tombol kirim	Sistem menampilkan notifikasi <i>withdraw</i> sedang diproses dan sistem menampilkan rincian <i>withdraw</i> pada histori.
<b>Kondisi Akhir</b>	Fungsi ini memungkinkan untuk melakukan penarikan saldo	

## 9. Fungsi Jadwal Vendor

Tabel 3. 23 Kebutuhan Fungsi Jadwal Vendor

<b>Nama Fungsi</b>	Fungsi Jadwal Vendor	
<b>Pengguna</b>	Vendor	
<b>Deskripsi</b>	Fungsi ini merupakan kegiatan pencatatan jadwal yang dimiliki oleh vendor	
<b>Kondisi Awal</b>	Data User, Data Jadwal	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<b>Pencatatan Jadwal Vendor</b>	
	Pengguna memilih menu <i>category vendor</i>	Sistem menampilkan halaman <i>category vendor</i>
	Pengguna memilih <i>category bisnis</i> , kemudian menekan tombol jadwal	Sistem menampilkan <i>form</i> jadwal



	Pengguna mengisi data jadwal. Kemudian menekan tombol tambah jadwal	Sistem menyimpan data jadwal
<b>Kondisi Akhir</b>	Fungsi ini memungkinkan untuk melakukan pencatatan dan menyimpan jadwal vendor	

#### 10. Fungsi Pembuatan Laporan Periodik (Transaksi, Pendapatan Perbulan)

Tabel 3. 24 Kebutuhan Fungsi Pembuatan Laporan Tahunan

<b>Nama Fungsi</b>	Fungsi Pembuatan Laporan Tahunan (Member Vendor, Member <i>Customer</i> , Transaksi, Pendapatan)	
<b>Pengguna</b>	Vendor	
<b>Deskripsi</b>	Fungsi ini merupakan kegiatan membuat laporan periodik (Transaksi, Pendapatan) yang dilakukan oleh admin dan kemudian di cetak	
<b>Kondisi Awal</b>	Data Transaksi, Data Kategori Vendor	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<b>Halaman Pembuatan Laporan Tahunan</b>	
	Pengguna memilih jenis laporan (Transaksi/ Pendapatan)	Sistem menampilkan halaman halaman laporan yang dipilih
	Pengguna memilih periode	Sistem menampilkan data laporan sesuai periode yang dipilih
	Pengguna menekan tombol cetak laporan	Sistem mencetak laporan
<b>Kondisi Akhir</b>	Fungsi ini memungkinkan pengguna untuk membuat laporan periodik	

### C. Analisis Kebutuhan Fungsional *Customer*

#### 1. Fungsi Registrasi *Customer*

Tabel 3. 25 Kebutuhan Fungsi Registrasi *Customer*

<b>Nama Fungsi</b>	Fungsi Registrasi <i>Customer</i>
<b>Pengguna</b>	<i>Customer</i>

<b>Deskripsi</b>	Fungsi ini merupakan kegiatan yang dilakukan untuk melakukan pendaftaran untuk mencari <i>vendor</i> dan membuat paket <i>vendor Intimate Production</i>	
<b>Kondisi Awal</b>	Data <i>user</i>	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<b>Registrasi</b>	
	Pengguna memasukkan nama, <i>email</i> dan <i>password</i> kemudian menekan tombol <i>Sign Up</i> pada halaman registrasi/ <i>create account</i> .	Sistem melakukan pengecekan teradap kesesuaian nama, <i>email</i> dan <i>password</i> serta mengecek ada atau tidaknya data dalam <i>database</i> . Jika data belum ada pada <i>database</i> dan data telah sesuai, maka sistem akan menampilkan notifikasi registrasi berhasil. Sebaliknya jika nama, <i>email</i> dan <i>password</i> tidak sesuai atau sudah terdaftar di <i>database</i> , maka sistem akan menampilkan notifikasi registrasi gagal.
<b>Kondisi Akhir</b>	Fungsi registrasi/ <i>create account</i> memungkinkan pengguna untuk dapat menggunakan aplikasi.	

## 2. Fungsi Pengelolaan *Profile Customer*

Tabel 3. 26 Kebutuhan Fungsi Pengelolaan *Profile Customer*

<b>Nama Fungsi</b>	Fungsi Pengelolaan Data <i>Profile Customer</i>
<b>Pengguna</b>	<i>Customer</i>
<b>Deskripsi</b>	Fungsi ini merupakan kegiatan yang dilakukan untuk melakukan pengelolaan data <i>profile customer</i> .

<b>Kondisi Awal</b>	Data <i>customer</i> , Data <i>User</i>	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<b>Menyimpan/Mengubah Data <i>Profile</i></b>	
	Pengguna memilih menu <i>view profile</i>	Sistem akan menampilkan halaman <i>index</i> dari data halaman <i>view profile</i>
	Kemudian pengguna memasukkan data <i>profile</i> kedalam <i>form</i> dan menekan tombol tambah.	Sistem akan terlebih dahulu mengecek kelengkapan pengisian <i>form</i> dan kesesuaian pengisian <i>form</i> seperti <i>form</i> angka dan rentang data isian. Jika semua <i>field</i> terisi dan kesesuaian pengisian telah tepat, maka sistem akan menyimpan penambahan data dari <i>form</i> yang telah diisi kedalam tabel <i>customer</i> . Sebaliknya, sistem akan menampilkan halaman <i>form</i> kembali dan peringatan sesuai dengan kesalahan pengisian yang telah dilakukan pengguna.
<b>Kondisi Akhir</b>	Fungsi ini dapat menyimpan dan mengubah data <i>profile customer</i>	

### 3. Fungsi Pencarian Vendor

Tabel 3. 27 Kebutuhan Fungsi Pencarian Vendor

<b>Nama Fungsi</b>	Fungsi Pencarian Vendor
<b>Pengguna</b>	<i>Customer</i>

<b>Deskripsi</b>	Fungsi ini merupakan kegiatan yang dilakukan oleh <i>customer</i> untuk mencari vendor yang diinginkan.	
<b>Kondisi Awal</b>	Data Kategori <i>Vendor</i>	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<b>Pencarian Vendor</b>	
	Pengguna memilih menu <i>home</i>	Sistem menampilkan daftar <i>brand</i> vendor/kategori vendor
	Pengguna memasukkan nama <i>brand</i> yang akan dicari, kemudian menekan tombol submit	Sistem menampilkan <i>brand</i> yang dicari
	Pengguna menekan tombol lihat <i>profile</i>	Sistem menampilkan <i>profile brand</i> /kategori vendor
<b>Kondisi Akhir</b>	Fungsi dapat melakukan pencarian <i>brand</i> /kategori vendor sesuai yang diinginkan <i>customer</i>	

#### 4. Fungsi *Customize Order Vendor*

Tabel 3. 28 Kebutuhan Fungsi *Customize Order Vendor*

<b>Nama Fungsi</b>	Fungsi <i>Custom Order Vendor</i>	
<b>Pengguna</b>	<i>Customer</i>	
<b>Deskripsi</b>	Fungsi ini merupakan kegiatan yang dilakukan oleh <i>customer</i> untuk memesan paket vendor secara <i>customize</i>	
<b>Kondisi Awal</b>	Data Paket <i>Vendor</i> , Data Kategori <i>Vendor</i> , Data Rating, Data Transaksi, Data Detail Transaksi, Data Notifikasi	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<b><i>Customize Order Vendor</i></b>	
	Pengguna menekan tombol <i>add to cart</i> untuk menambahkan	Sistem menambahkan paket yang dipilih ke keranjang belanja

	paket vendor yang diinginkan	
	Pengguna menekan tombol keranjang di pojok kanan atas	Sistem menampilkan paket yang dipilih
	Pengguna mengisi data transaksi, kemudian menekan tombol bayar	Sistem menampilkan halaman pembayaran
<b>Kondisi Akhir</b>	Fungsi dapat melakukan pembuatan paket secara <i>customize</i>	

### 5. Fungsi *Generate* Paket dan Pemesanan Paket

Tabel 3. 29 Kebutuhan Fungsi *Generate* Paket dan Pemesanan Paket Pada *Customer*

<b>Nama Fungsi</b>	Fungsi <i>Generate</i> Paket dan Pemesanan Paket	
<b>Pengguna</b>	<i>Customer</i>	
<b>Deskripsi</b>	Fungsi ini merupakan kegiatan yang dilakukan untuk membuat paket yang kemudian dipakai menentukan paket pernikahan.	
<b>Kondisi Awal</b>	Data <i>Vendor</i> , Data Paket <i>Vendor</i> , Data Kategori, Data Sub Kriteria, Data Kategori <i>Vendor</i> , Data Nilai, Data Rating, Data Transaksi	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<b><i>Generate</i> Paket</b>	
	Pengguna memilih menu <i>generate</i>	Sistem menampilkan <i>form generate</i> paket
	Pengguna memasukan <i>budget</i> dan kategori, kemudian menekan tombol submit.	Sistem menampilkan pilihan kriteria yang telah dipilih.
Pengguna memilih prioritas kriteria dan sub kriteria pada tiap kriteria kemudian menekan tombol <i>Generate</i>	Sistem menampilkan pilihan paket sesuai kriteria yang telah dipilih.	

	Pengguna memilih salah satu paket dengan menekan tombol <i>Choose</i> pada salah satu pilihan paket	Sistem menampilkan <i>form</i> transaksi
	Pengguna memasukkan tanggal acara, tema acara, jumlah acara kemudian menekan tombol <i>Order</i>	Sistem menampilkan notifikasi <i>order</i> akan di proses.
<b>Kondisi Akhir</b>	Fungsi dapat melakukan <i>generate</i> paket untuk menentukan paket yang dipilih sesuai <i>budget customer</i>	

## 6. Fungsi Histori Transaksi

Tabel 3. 30 Kebutuhan Fungsi Histori Transaksi Pada *Customer*

<b>Nama Fungsi</b>	Fungsi Histori Transaksi	
<b>Pengguna</b>	<i>Customer</i>	
<b>Deskripsi</b>	Fungsi ini merupakan kegiatan yang dilakukan untuk melihat status pemesanan/transaksi yang telah dilakukan	
<b>Kondisi Awal</b>	Data transaksi, Data Detail Transaksi, Data Pembayaran	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<b>Histori Transaksi</b>	
	Pengguna memilih menu histori transaksi	Sistem menampilkan halaman histori transaksi
	Pengguna menekan tombol <i>view</i> untuk melihat detail transaksi	Sistem menampilkan detail transaksi
<b>Kondisi Akhir</b>	Fungsi dapat melihat status pembayaran, transaksi dan melihat detail transaksi	

## 7. Fungsi Konfirmasi Pembayaran

Tabel 3. 31 Kebutuhan Fungsi Konfirmasi Pembayaran

<b>Nama Fungsi</b>	Fungsi Konfirmasi Pembayaran	
<b>Pengguna</b>	<i>Customer</i>	
<b>Deskripsi</b>	Fungsi ini merupakan kegiatan yang dilakukan <i>customer</i> untuk melakukan pembayaran dengan menginputkan nama <i>bank</i> , nomor rekening, jumlah transfer, dan berita transfer	
<b>Kondisi Awal</b>	Data Bank, Data Transaksi, Data Pembayaran	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<b>Histori Konfirmasi Pembayaran</b>	
	Pengguna menekan menu <i>history transaction</i>	Sistem menampilkan halaman <i>history transaction</i>
	Pengguna menekan tombol konfirmasi pembayaran	Sistem menampilkan <i>form</i> konfirmasi pembayaran
	Pengguna mengisi data konfirmasi pembayaran dan menekan tombol submit	Sistem akan mengecek kelengkapan data. Jika lengkap maka sistem akan menampilkan notifikasi mohon menunggu konfirmasi penerimaan pembayaran oleh admin. Jika tidak lengkap maka sistem tidak akan menyimpan data, dan sistem akan tetap dihalaman tersebut sampai <i>form</i> diisi dengan lengkap kemudian pengguna menekan tombol submit.
<b>Kondisi Akhir</b>	Fungsi ini dapat melakukan konfirmasi pembayaran dan melihat status pembayaran	

8. Fungsi *Rating* dan *Review*Tabel 3. 32 Kebutuhan Fungsi *Rating* dan *Review*

<b>Nama Fungsi</b>	Fungsi <i>Rating</i> dan <i>Review</i>	
<b>Pengguna</b>	<i>Customer</i>	
<b>Deskripsi</b>	Fungsi ini merupakan kegiatan yang dilakukan <i>customer</i> untuk memberikan rating atau ulasan mengenai vendor	
<b>Kondisi Awal</b>	Data <i>Category Vendor</i> , Data <i>Customer</i>	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<b><i>Rating dan Review</i></b>	
	Pengguna mencari vendor	Sistem menampilkan halaman vendor pilihan
	Pengguna memilih <i>review</i> sekarang	Sistem menampilkan <i>form rating</i> dan <i>review</i>
	Pengguna mengisi data <i>rating</i> dan <i>review</i> . Kemudian menekan tombol simpan.	Sistem menyimpan data <i>rating</i> dan <i>review</i>
<b>Kondisi Akhir</b>	Fungsi ini dapat melakukan pencatatan dan menyimpan data <i>rating</i> dan <i>review</i>	

9. Fungsi *Withdraw*Tabel 3. 33 Kebutuhan Fungsi *Withdraw*

<b>Nama Fungsi</b>	Fungsi <i>Withdraw</i>	
<b>Pengguna</b>	<i>Customer</i>	
<b>Deskripsi</b>	Fungsi ini merupakan kegiatan yang dilakukan <i>customer</i> untuk menarik saldo	
<b>Kondisi Awal</b>	Data <i>Withdraw</i> , Data Buku Besar, Data <i>User</i>	
<b>Alur Normal</b>	<b>Aksi Pengguna</b>	<b>Respon Sistem</b>
	<b><i>Withdraw</i></b>	
	Pengguna memilih menu <i>withdraw</i>	Sistem menampilkan halaman <i>withdraw</i>



		dan data <i>history withdraw</i>
	Pengguna memilih <i>tab withdraw</i> . Kemudian mengisi data <i>bank</i> , rekening, dan jumlah penarikan. Kemudian menekan tombol kirim	Sistem menyimpan data <i>withdraw</i>
<b>Kondisi Akhir</b>	Fungsi ini dapat melakukan penarikan saldo yang dilakukan oleh <i>customer</i>	

### 3.1.4 Analisis kebutuhan non fungsional

Analisis kebutuhan non fungsional dilakukan untuk mengetahui spesifikasi kebutuhan sistem beserta dengan hak akses pada sistem penentuan vendor pernikahan. Setiap entitas memiliki hak akses yang berbeda dalam menggunakan fungsi-fungsi di dalam sistem.

Tabel 3. 34 Tabel Analisis Kebutuhan Non Fungsional

Kriteria	Kebutuhan Non Fungsional
Keamanan	<ol style="list-style-type: none"> <li>1. <i>Login &amp; Logout User</i> menggunakan enkripsi <i>password "md5"</i></li> <li>2. Pembatasan hak akses tiap pengguna menggunakan konsep <i>Role Based Access Control</i></li> </ol>
<i>Usability</i>	Digunakan <i>AJAX Partiall View</i> guna mempermudah pengguna dalam mengakses dan menggunakan aplikasi penentuan vendor pernikahan

### 3.1.5 Analisis kebutuhan sistem

Analisis kebutuhan sistem dilakukan untuk mengetahui spesifikasi kebutuhan sistem dalam membangun aplikasi penentuan vendor pernikahan pada

Intimate Production. Spesifikasi kebutuhan sistem melibatkan analisis perangkat keras/*hardware* dan analisis perangkat lunak/*software*.

### A. Kebutuhan Perangkat Keras

Kebutuhan perangkat keras adalah peralatan fisik yang membentuk suatu sistem komputer terstruktur, serta perangkat keras lain yang mendukung komputer dalam menjalankan fungsinya. Perangkat keras yang digunakan setidaknya memiliki spesifikasi dan kinerja yang baik untuk dapat menjalankan sistem tanpa ada suatu masalah. Kebutuhan hardware yang dibutuhkan adalah sebagai berikut:

#### 1. *Client Side*

- *Processor* Intel Core Duo
- Ram 2 GB
- Screen

#### 2. *Server Side*

- *Processor* Intel Core i5
- RAM 8 GB dan SSD 500 GB
- Monitor dengan resolusi 1024 x 768
- *Mouse & Keyboard*

### B. Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak atau *software* adalah suatu program yang digunakan untuk mengembangkan dan membangun aplikasi penentuan vendor pernikahan untuk usaha *wedding organizer* menggunakan metode smarter berbasis *web*. Perangkat lunak yang digunakan memiliki fungsi masing-masing yang berbeda. Perbedaan tersebut mulai dari pembuatan rancangan *document flow*,

*system flow*, *data flow diagram*, struktur *database*, hingga sistem itu sendiri.

Adapun perangkat lunak yang digunakan adalah sebagai berikut:

1. Microsoft Visio 2013 untuk membuat rancangan *document flow* dan *system flow*.
2. Power Designer 15.2 untuk membuat context diagram, *data flow diagram*, *conceptual data model* dan *physical data model*.
3. XAMPP untuk *web server*.
4. MySQL sebagai *database management System* (DBMS).
5. *Codeigniter* versi 3.0.0 sebagai *framework* aplikasi.
6. Mockingbot untuk membuat perancangan *interface* aplikasi.

### 3.2 Perancangan Sistem

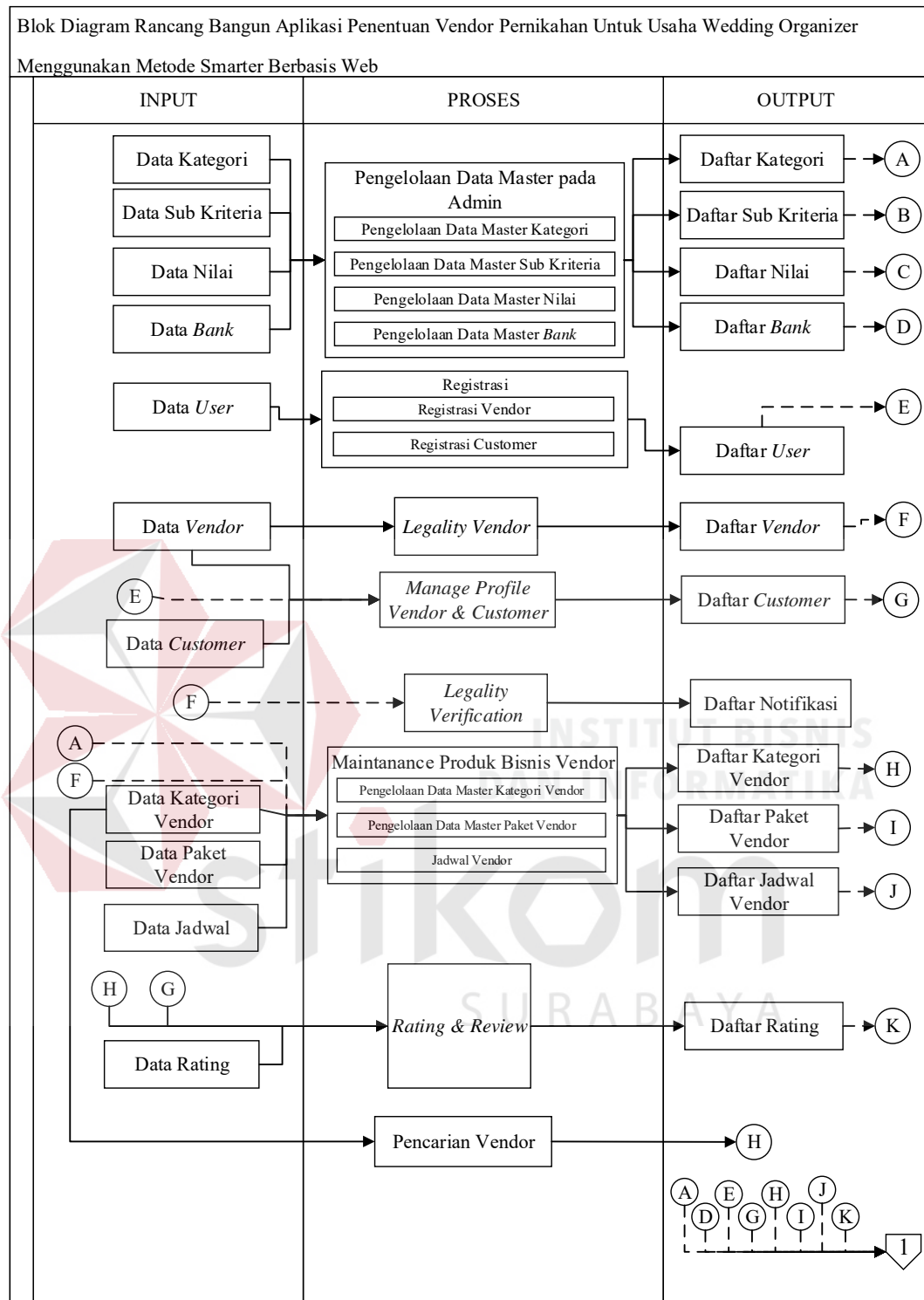
Tahapan sebelum melakukan perancangan sistem atau *modelling* adalah menentukan rencana kerja pengembangan aplikasi. Rencana kerja pengembangan aplikasi. Rencana kerja pengembangan aplikasi penentuan vendor pernikahan untuk usaha *wedding organizer* menggunakan metode *smarter* berbasis *web* ini adalah selama empat bulan melakukan pengumpulan data, analisis, perancangan sistem, pembuatan sistem, uji coba sistem dan pembuatan laporan.

Berdasarkan analisis permasalahan yang telah dilakukan, maka tahap selanjutnya adalah menentukan desain dan perancangan sesuai dengan analisis yang telah dilakukan. Tujuan dari desain dan perancangan sistem ini adalah untuk membuat kerangka dasar dalam melakukan implementasi sistem ini adalah untuk membuat kerangka dasar dalam melakukan implementasi ke sistem yang akan dibuat atau dibangun.

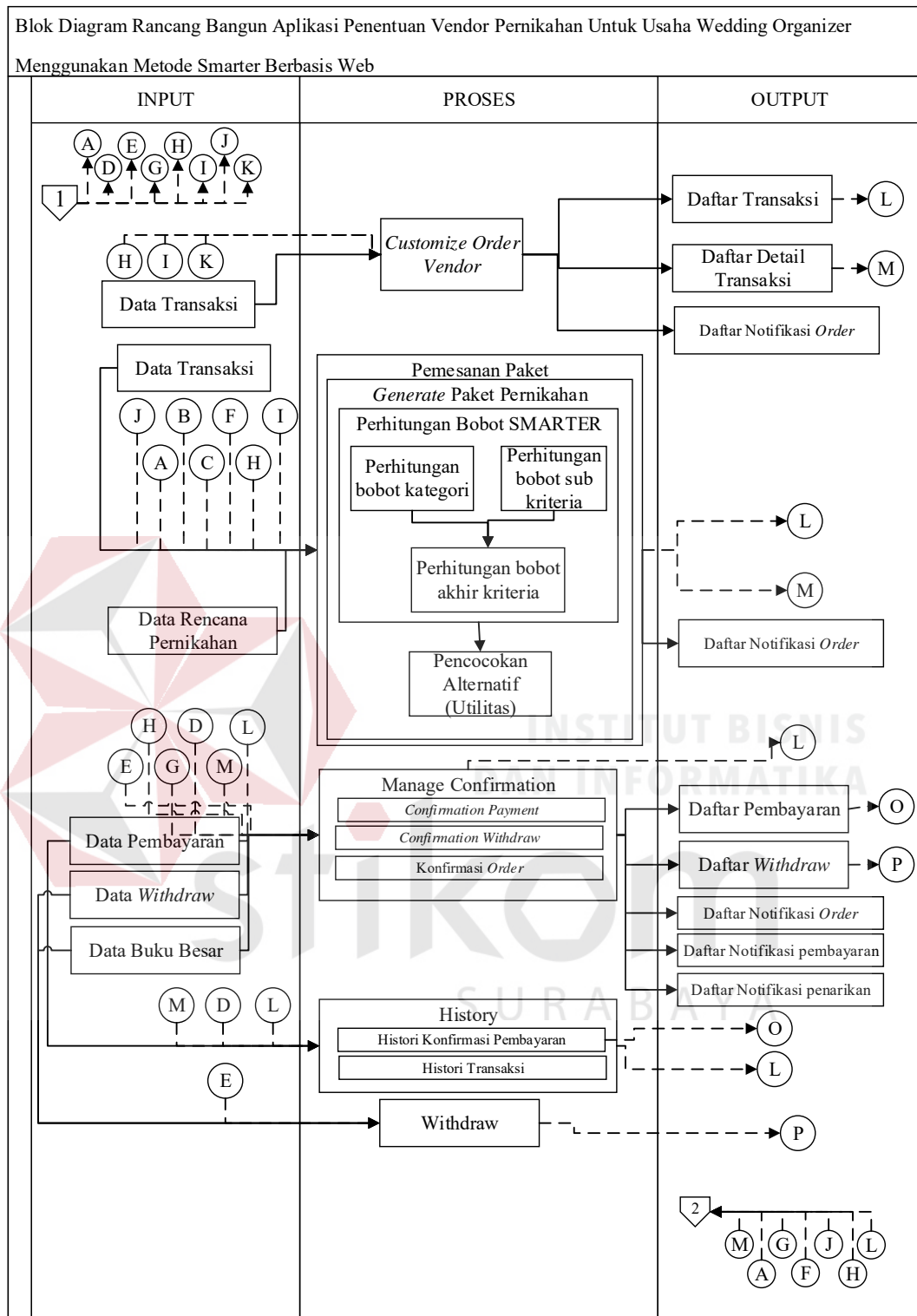
### 3.2.1 Blok Diagram

Aplikasi Penentuan *Vendor* Pernikahan Untuk Usaha *Wedding Organizer* Menggunakan Metode *Smarter Berbasis Web*. Aplikasi dapat mengelola transaksi *Customer* maupun mengelola paket yang dimiliki oleh *vendor*, sehingga memudahkan *vendor* apabila bergerak pada lebih dari satu kategori *vendor*. Berikut bentuk Block Diagram aplikasi penentuan vendor pernikahan untuk usaha *wedding organizer* menggunakan metode *SMARTER* berbasis *web* yang akan dibuat :

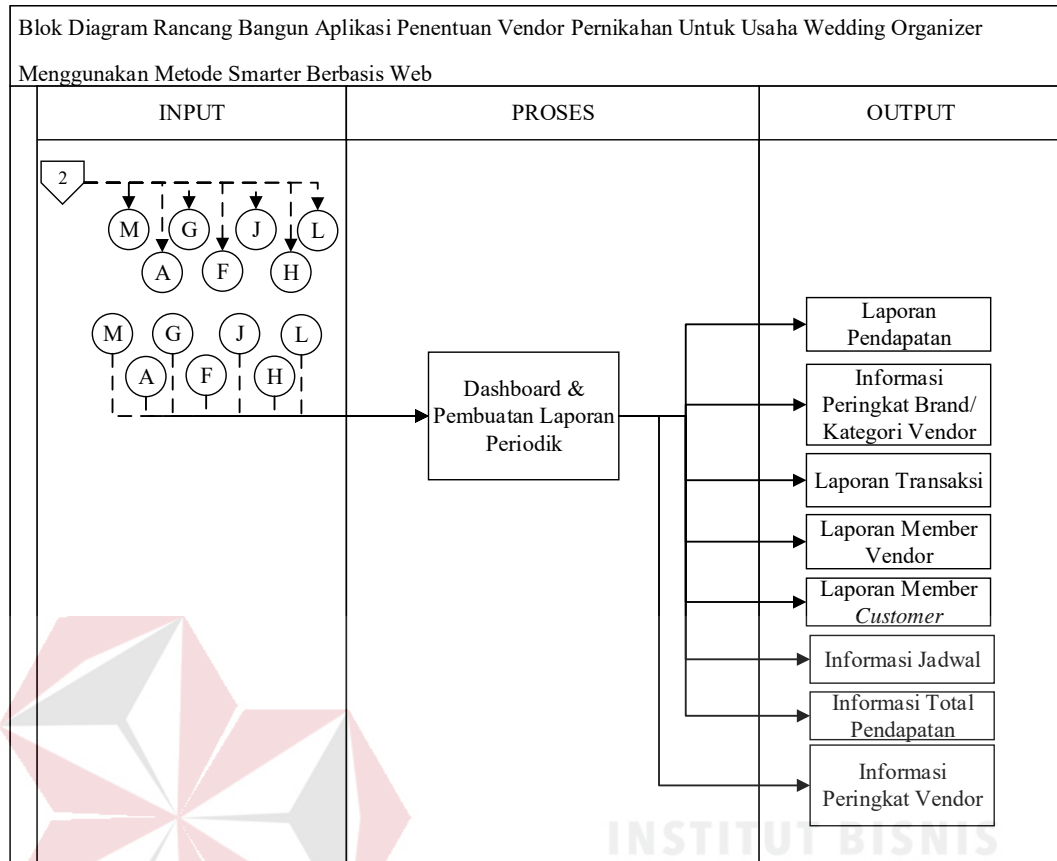




Gambar 3. 2 Blok Diagram Aplikasi Penentuan Vendor Pernikahan 1



Gambar 3. 3 Blok Diagram Aplikasi Penentuan Vendor Pernikahan 2



Gambar 3. 4 Blok Diagram Aplikasi Penentuan Vendor Pernikahan 3

### 3.2.2 Proses Modelling

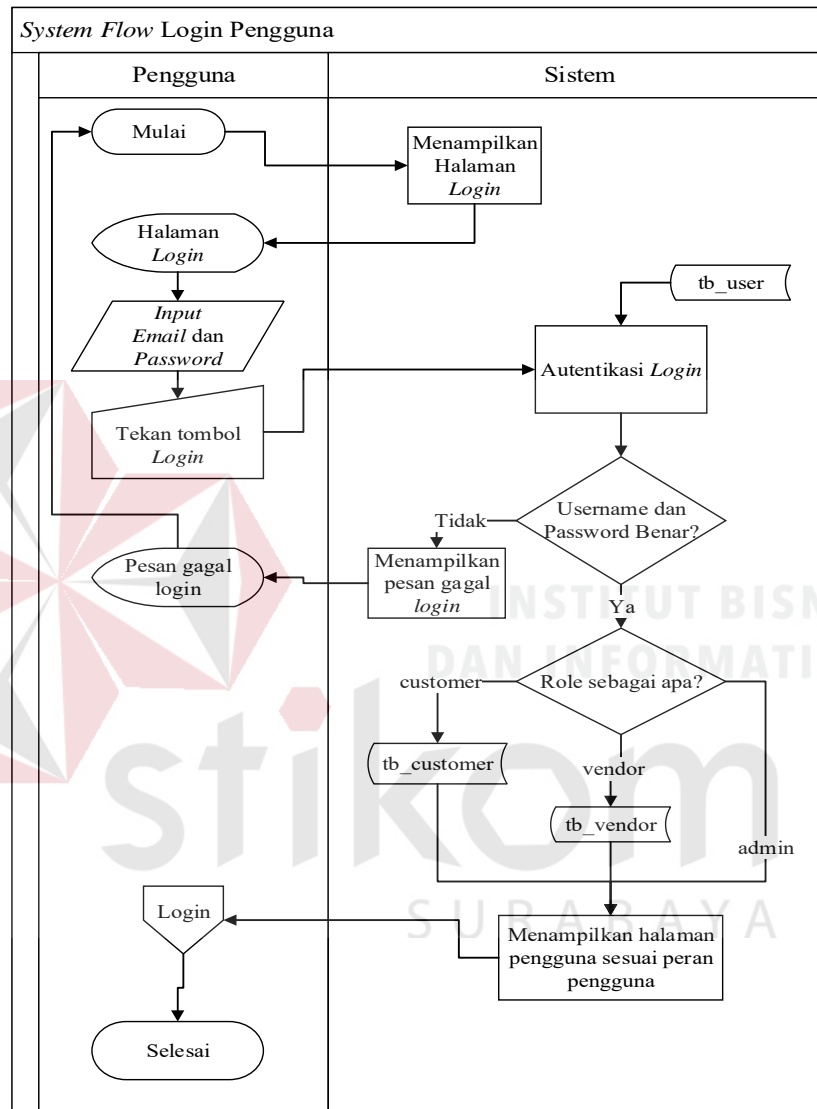
Proses *Modelling* menggambarkan bagaimana suatu sistem dan bisnis beroperasi serta mengilustrasikan aktivitas-aktivitas yang dilakukan dan bagaimana data berpindah. Pada proses *modelling* akan dilakukan perancangan *System Flow* dan perancangan *Data Flow Diagram* (DFD).

#### A. System Flow

*System Flow* merupakan atau perangkat diagram grafik yang mengkomunikasi aliran data dan memiliki arus pekerjaan dari suatu sistem yang menjelaskan urutan prosedur yang terdapat didalam sistem.

### a. System Flow Login Pengguna

Berikut adalah *system flow* dari *login* untuk setiap pengguna yang dapat dilihat pada gambar 3.5.



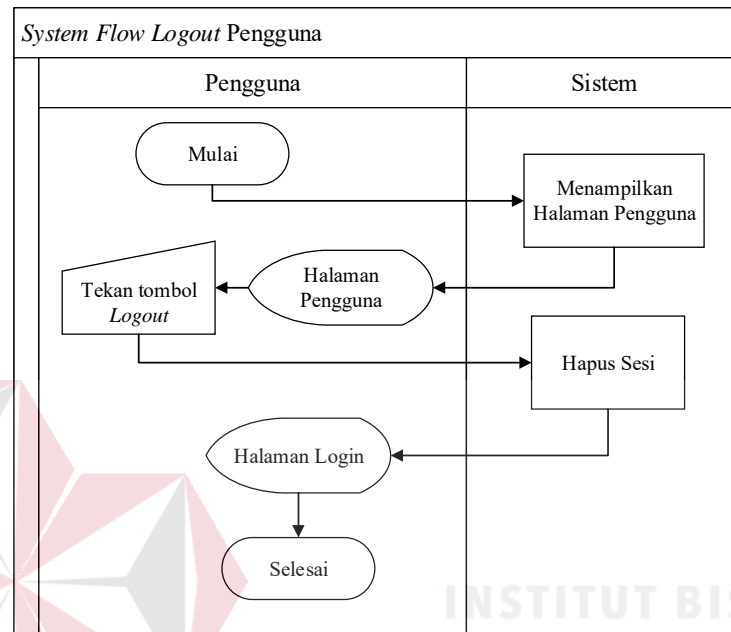
Gambar 3. 5 *System Flow Login Pengguna*

Gambar 3.5 menjelaskan proses *login* yang dapat dilakukan oleh pengguna sistem yaitu *admin*, *customer*, dan *vendor* dengan memasukkan *email* dan *password*. Sistem akan melakukan pengecekan terhadap kesesuaian *email* dan *password* serta mengecek *role* pengguna.



### b. System Flow Logout

Berikut adalah *system flow* untuk *logout* dari aplikasi yang dilakukan oleh setiap pengguna yang dapat dilihat pada Gambar 3.6.

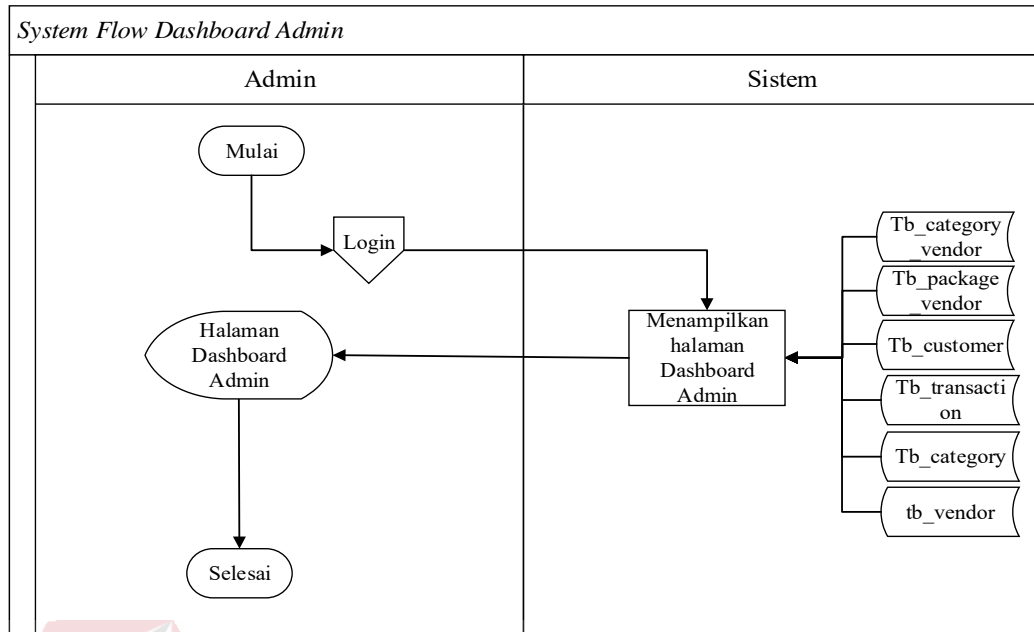


Gambar 3. 6 System Flow Logout

Gambar 3.6 menjelaskan proses *logout* yang dapat dilakukan oleh pengguna sistem yaitu admin, *customer*, dan *vendor* dengan menekan tombol *logout* pada halaman, kemudian sistem akan menghapus sesi dan proses kembali ke halaman *login*.

### c. System Flow Dashboard Pada Admin

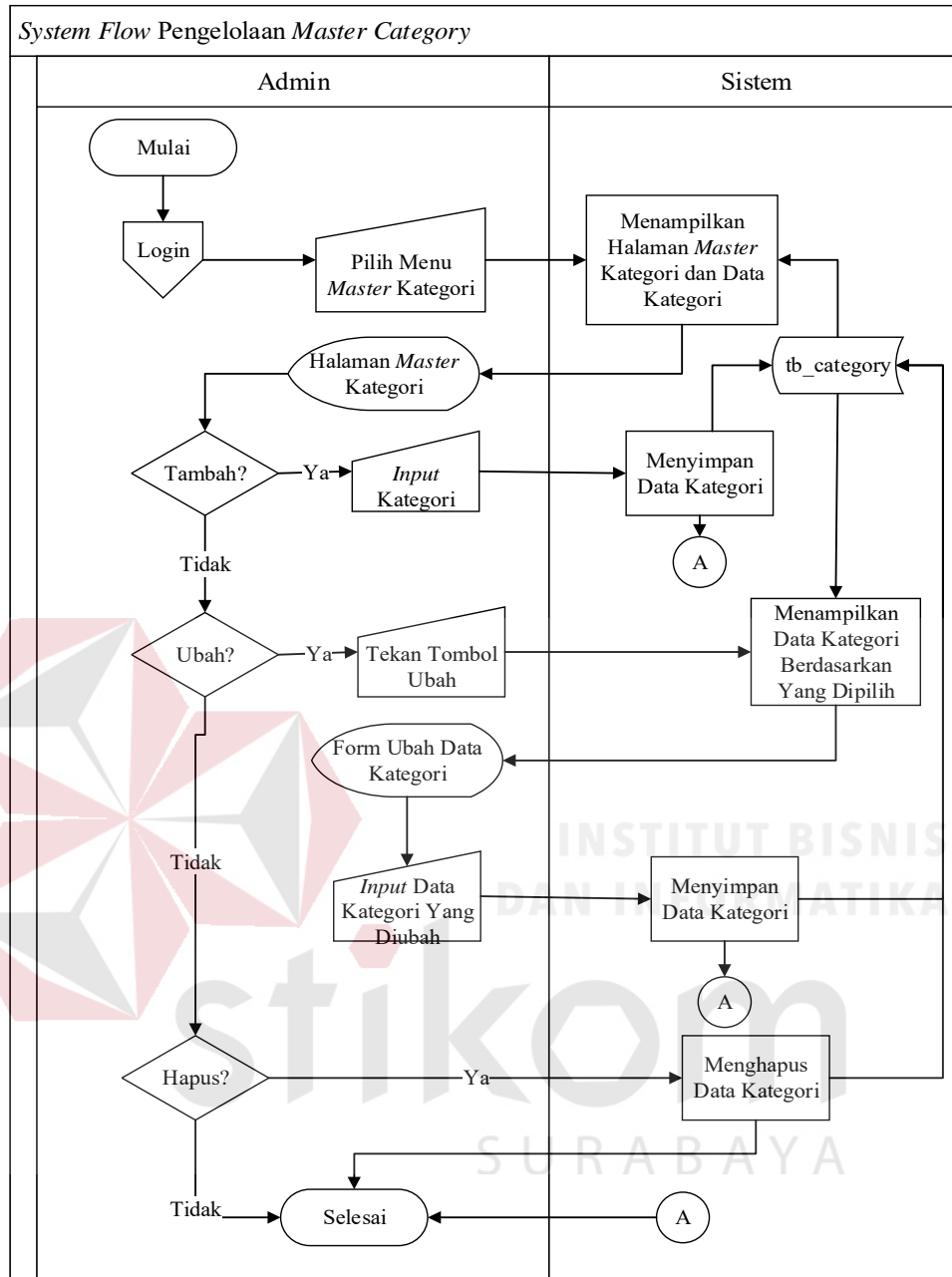
*System flow Dashboard* pada tampilan admin menggambarkan tentang informasi peringkat vendor, laporan member vendor, laporan member *customer*, laporan transaksi, dan laporan pendapatan. *Dashboard* pada admin dapat dilihat pada Gambar 3.7.



Gambar 3. 7 *System Flow Dashboard* Pada Admin

#### d. *System Flow* Master Kategori

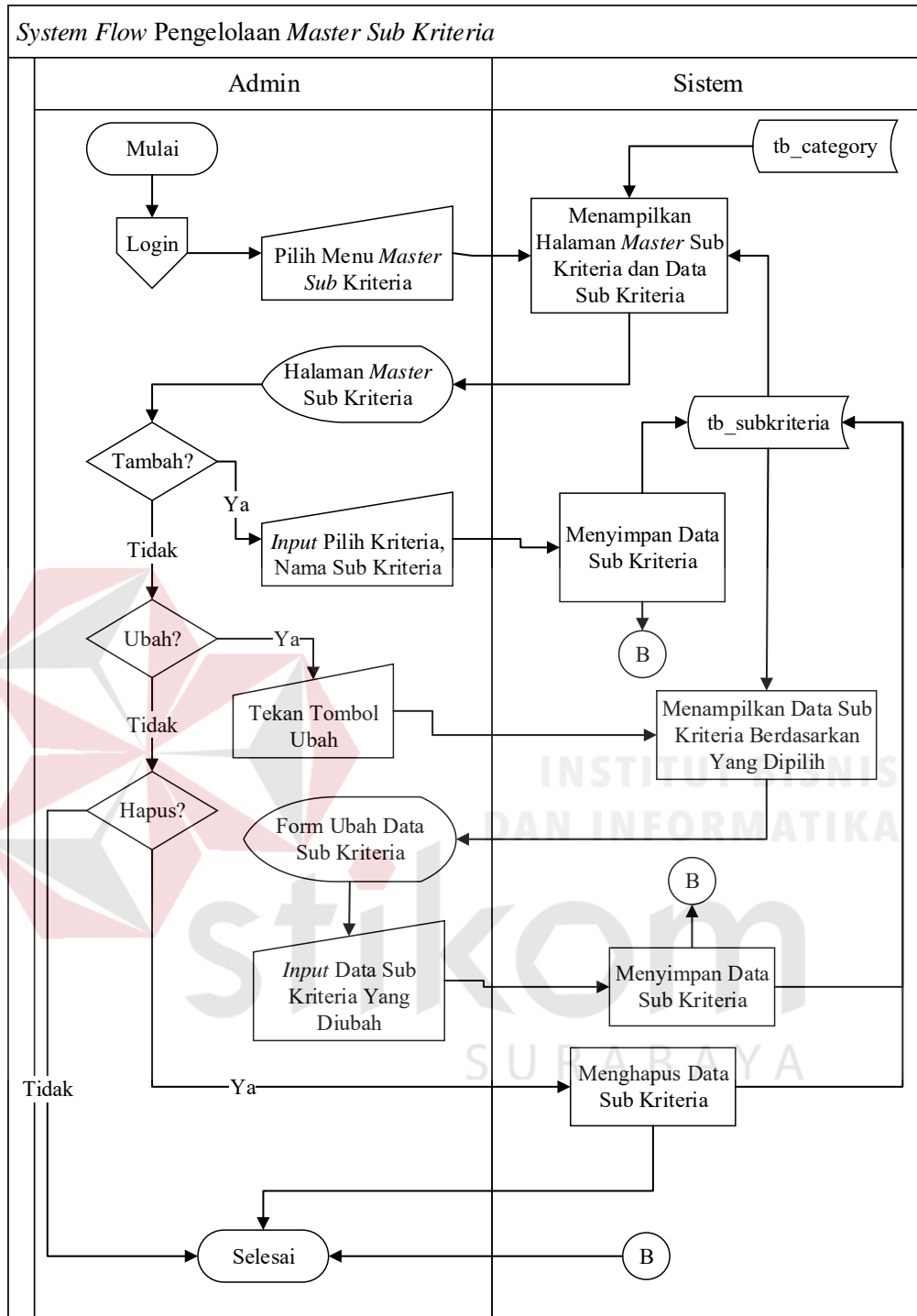
*System Flow* Master Kategori merupakan proses pengelolaan master yang dilakukan oleh admin sistem yang didalamnya terdapat pengelolaan data yang meliputi tambah data, ubah data dan hapus data. *System Flow* untuk Master Kategori dapat dilihat pada Gambar 3.8.



Gambar 3. 8 *System Flow* Pengelolaan Master Kategori

#### e. *System Flow* Master Sub Kriteria

*System Flow* Master Sub Kriteria merupakan proses pengelolaan master yang dilakukan oleh admin sistem yang didalamnya terdapat pengelolaan data yang meliputi tambah data, ubah data dan hapus data. *System Flow* untuk Master Sub Kriteria dapat dilihat pada Gambar 3.9.

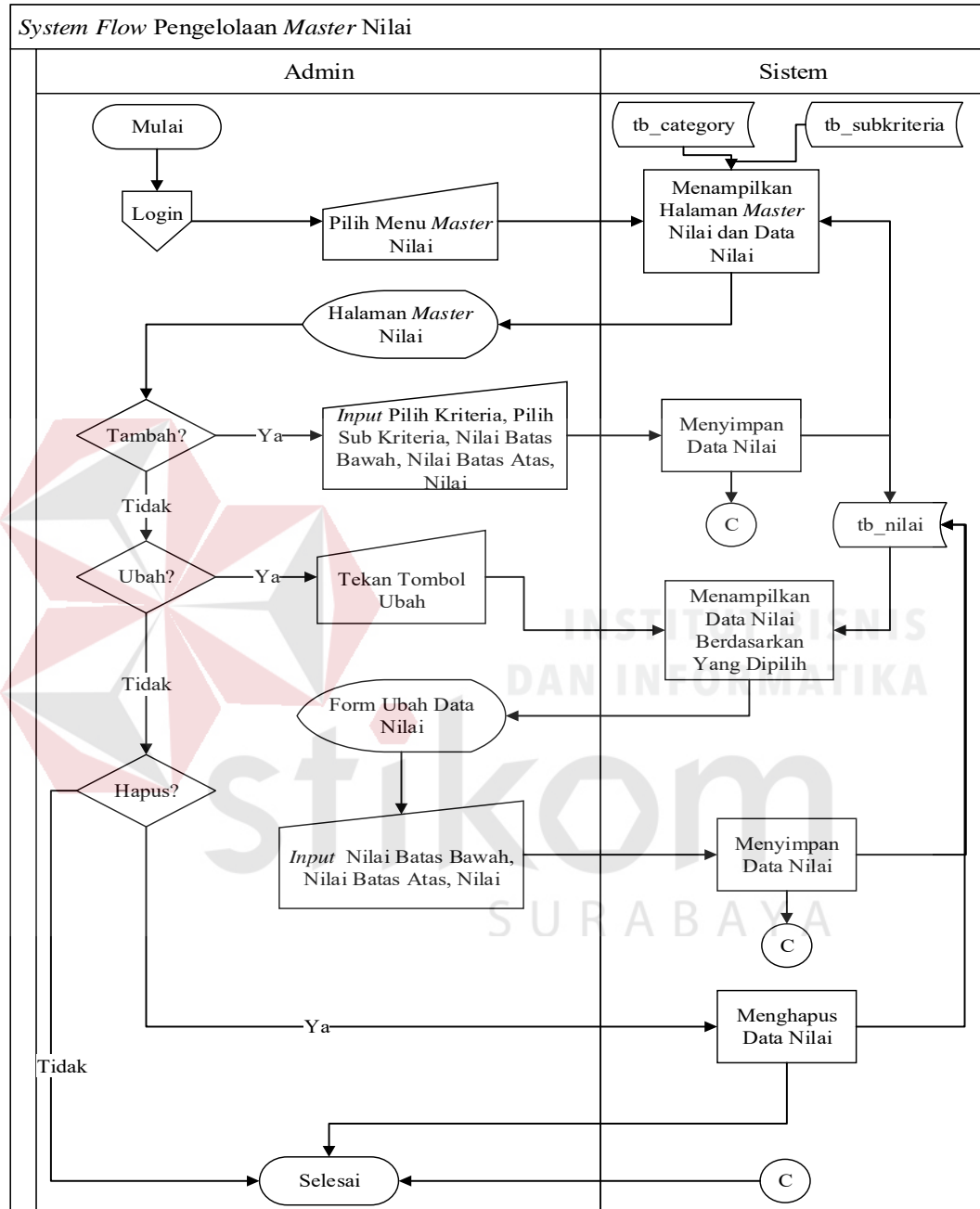


Gambar 3. 9 *System Flow* Master Sub Kriteria

#### f. *System Flow* Master Nilai

*System Flow* Master Nilai merupakan proses pengelolaan master yang dilakukan oleh admin sistem yang didalamnya terdapat pengelolaan data yang

meliputi tambah data, ubah data dan hapus data. *System Flow* untuk Master Nilai dapat dilihat pada Gambar 3.10.

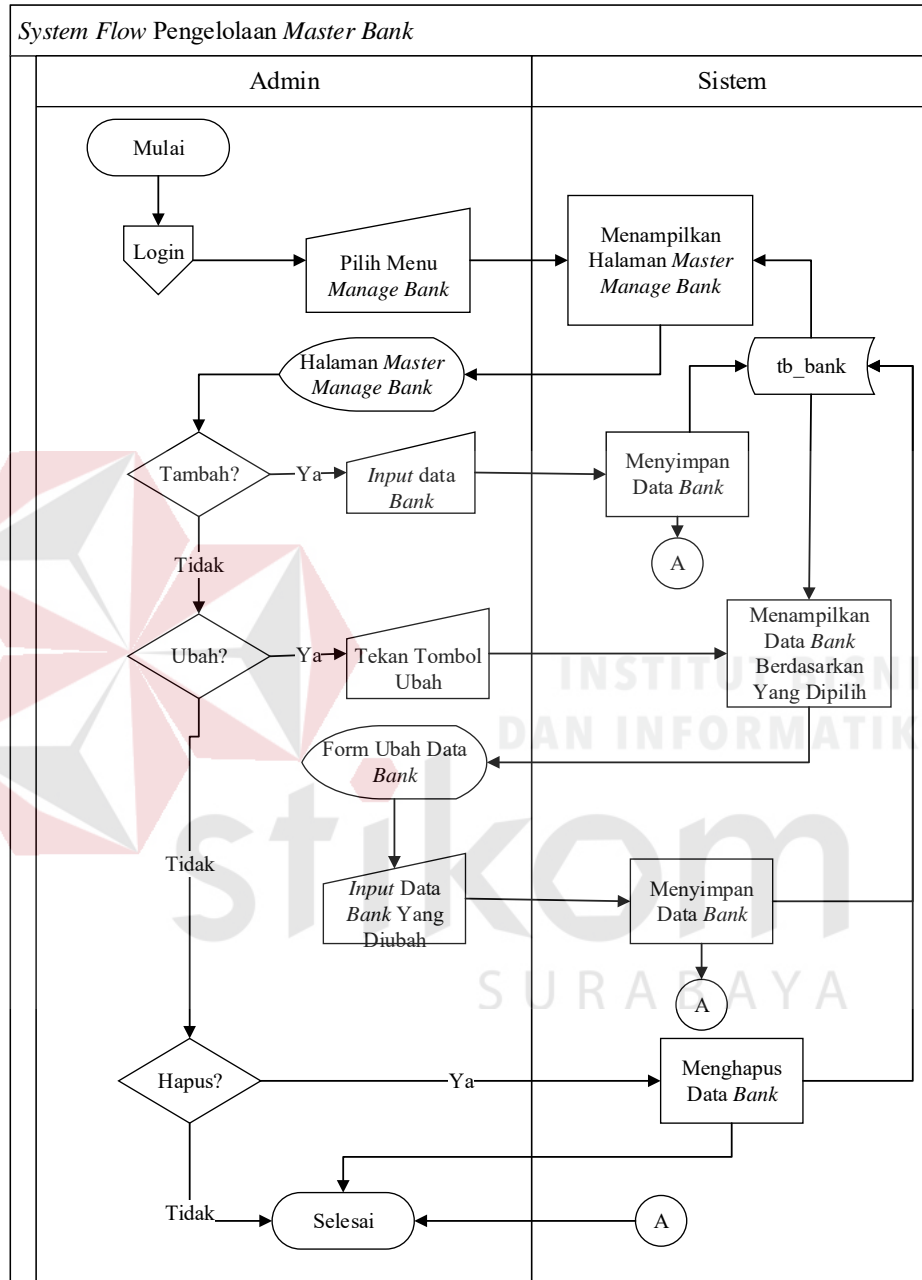


Gambar 3. 10 *System Flow* Pengelolaan Master Nilai

#### g. *System Flow* Pengelolaan Master Bank

*System Flow* Master Bank merupakan proses pengelolaan master yang dilakukan oleh admin sistem yang didalamnya terdapat pengelolaan data yang

meliputi tambah data, ubah data dan hapus data. Data yang disimpan meliputi kode *bank* dan nama *bank*. *System Flow Master Bank* dapat dilihat pada Gambar 3.11.

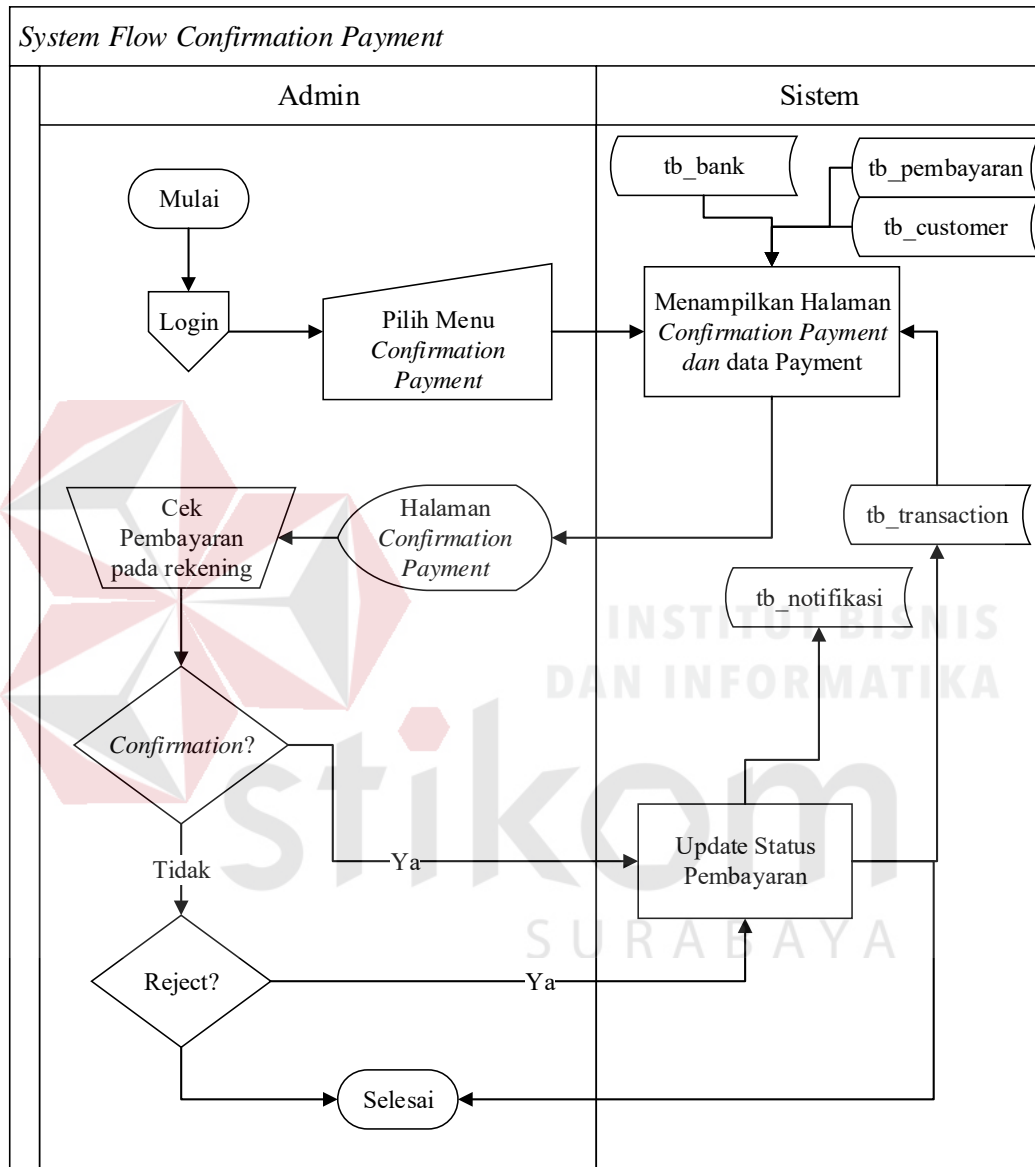


Gambar 3. 11 *System Flow* Pengelolaan Master Bank

#### h. *System Flow* Konfirmasi Pembayaran

Konfirmasi pembayaran merupakan proses yang dilakukan oleh admin untuk mengecek pembayaran di rekening setelah *customer* mengkonfirmasi

pembayaran. Jika *customer* telah membayar maka admin akan mengkonfirmasi dan jika belum, maka admin akan *me-reject*. Berikut adalah *system flow* konfirmasi pembayaran yang dapat dilihat pada Gambar 3.12.

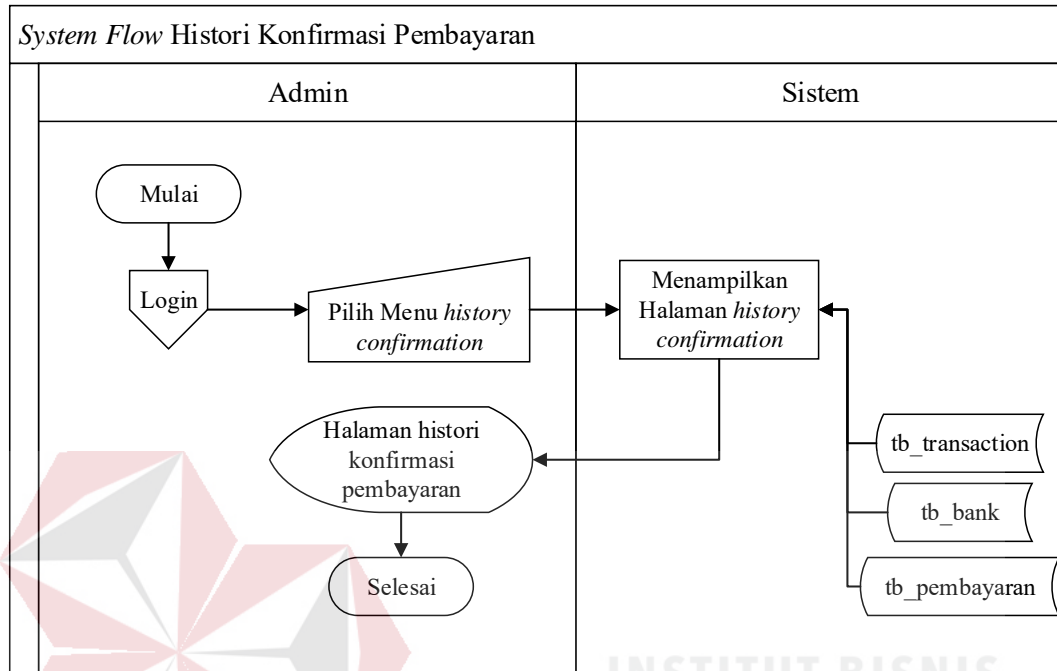


Gambar 3. 12 *System Flow Confirmation Payment*

#### i. *System Flow* Histori Konfirmasi Pembayaran

Histori konfirmasi pembayaran merupakan proses yang dilakukan oleh admin untuk melihat riwayat dan status pembayaran yang telah di konfirmasi

maupun di *reject*. Berikut adalah *system flow* histori konfirmasi pembayaran yang dapat dilihat pada Gambar 3.13.

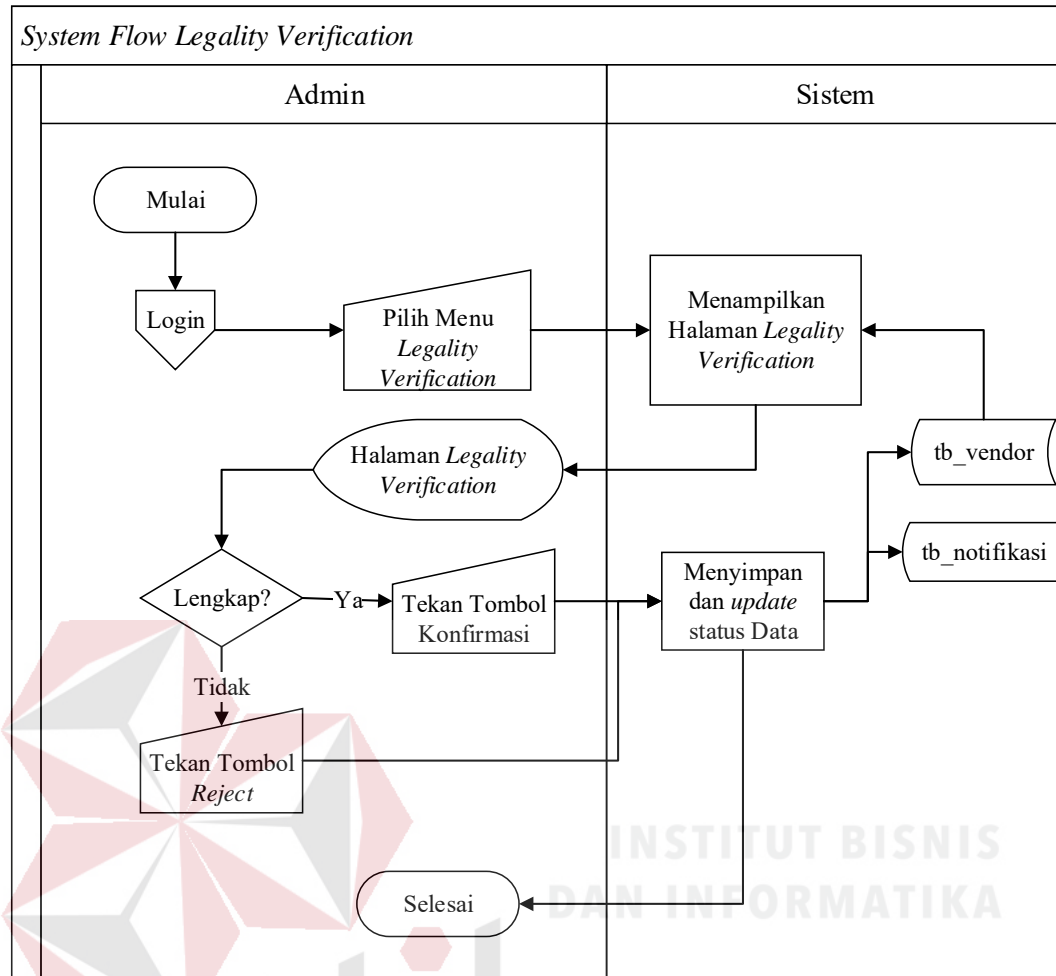


Gambar 3. 13 *System Flow* Histori Konfirmasi Pembayaran

#### j. *System Flow Legality Verification*

*Legality Verification* merupakan proses yang dilakukan oleh admin untuk mengecek data legal *vendor* yang mendaftar berupa nomor NPWP, nama badan usaha, alamat, KTP, nama pimpinan, NIK pimpinan, dan alamat pimpinan. Berikut adalah *system flow legality verification* yang dapat dilihat pada Gambar 3.14.

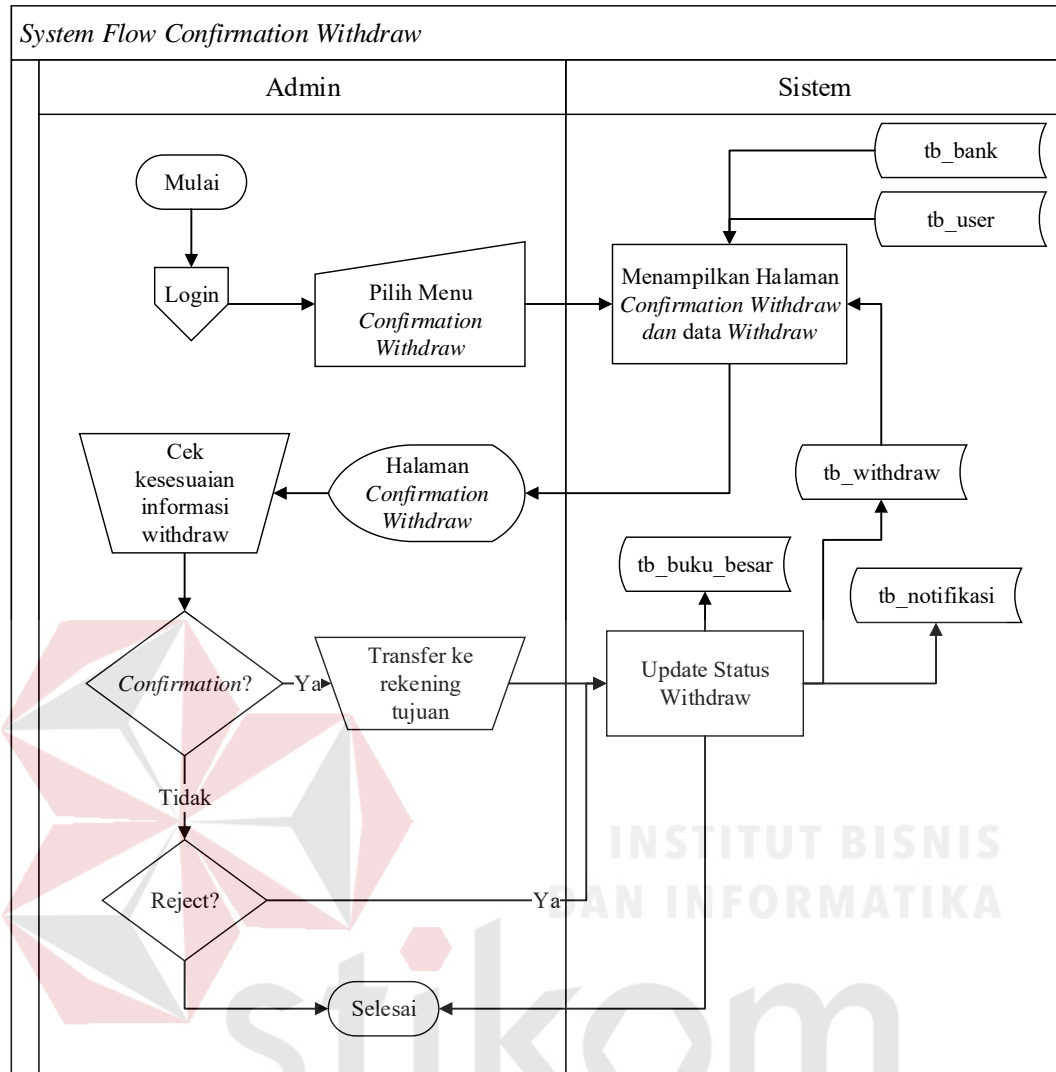




Gambar 3. 14 *System Flow Legality Verification*

#### k. *System Flow Confirmation Withdraw*

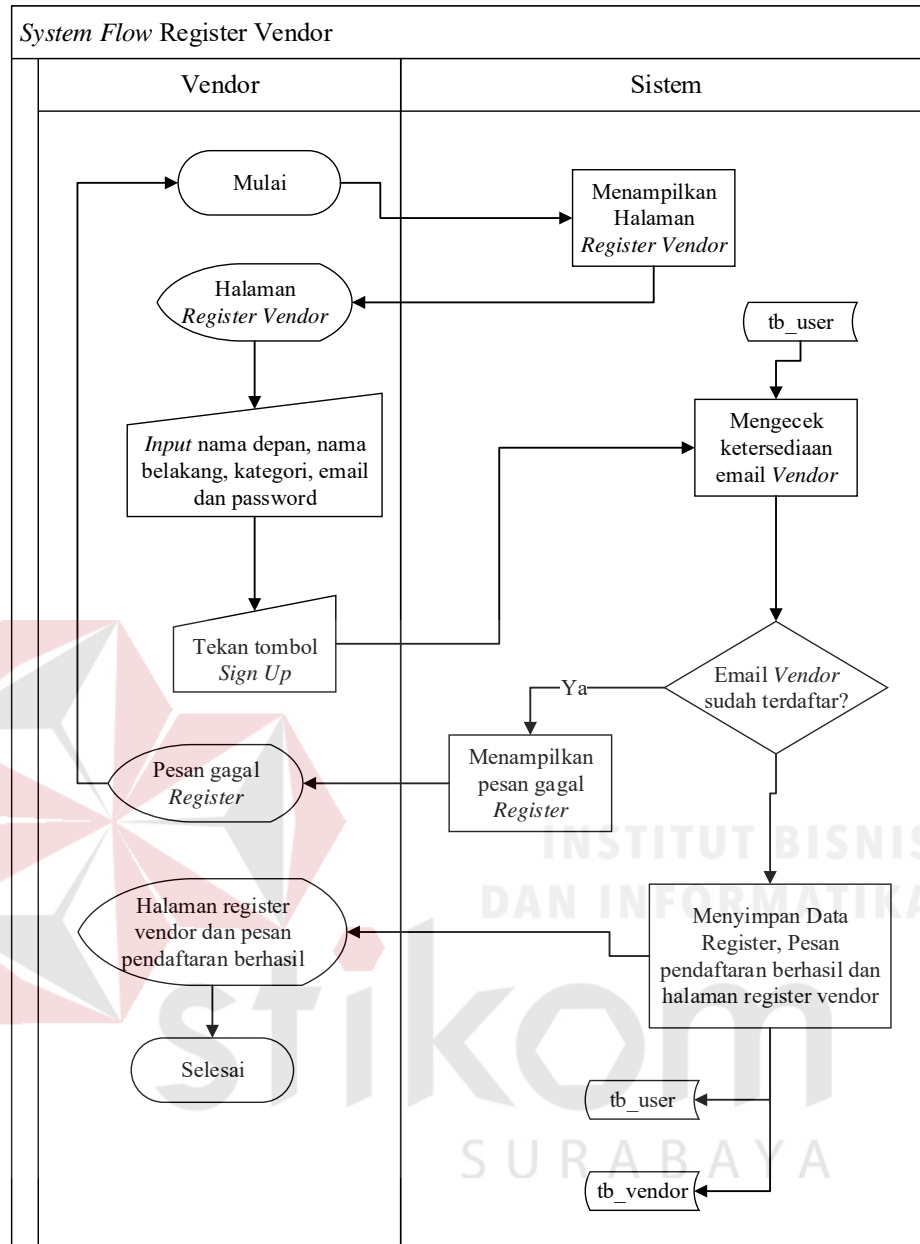
Konfirmasi *Withdraw* merupakan proses yang dilakukan oleh admin untuk melakukan pengecekan dan pengiriman dana yang ditarik oleh *customer* dan vendor. *System Flow Confirmation Withdraw* dapat dilihat pada Gambar 3.15.



Gambar 3. 15 *System Flow Confirmation Withdraw*

### 1. *System Flow Registrasi Vendor*

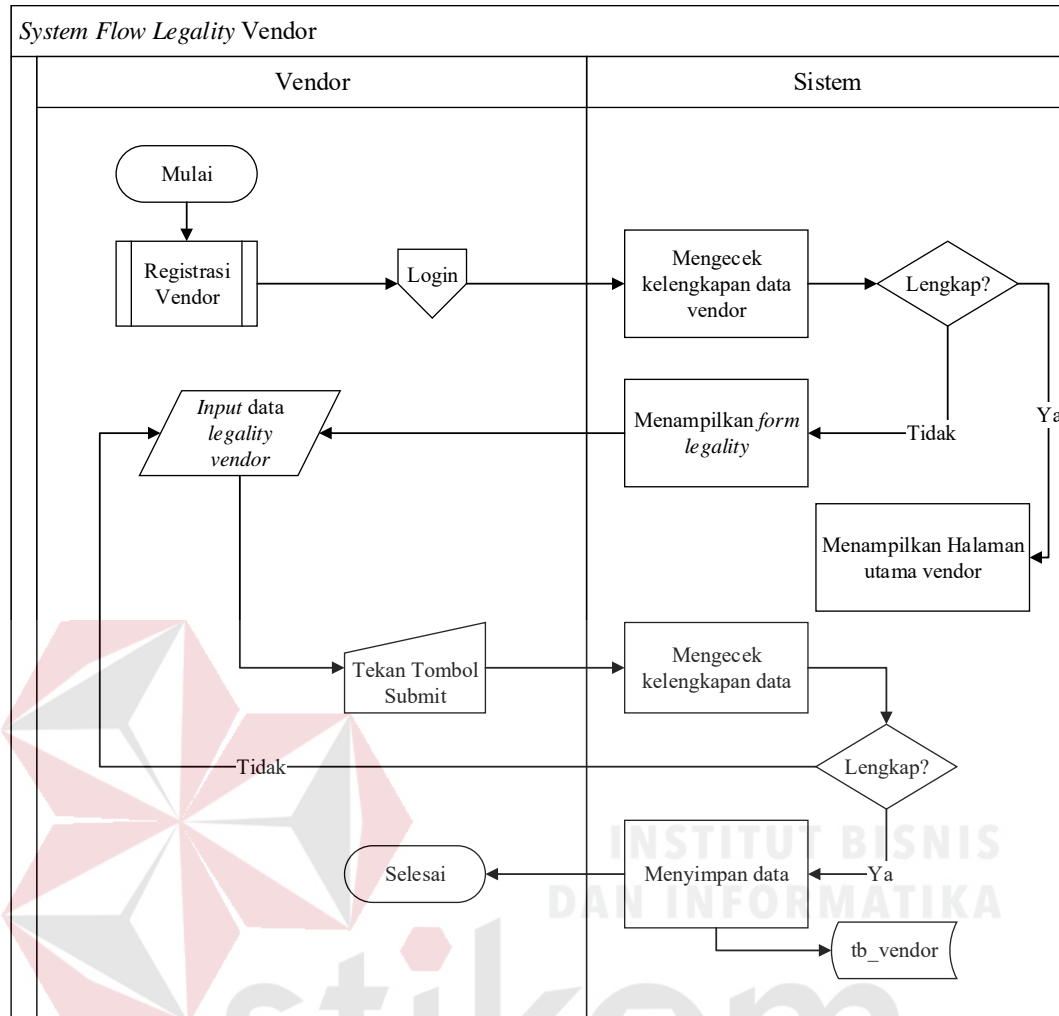
*System Flow Registrasi Vendor* merupakan proses yang dilakukan oleh *vendor* yang mendaftar melalui sistem dengan menginputkan identitas *vendor*, *email*, *password* dan kategori. Data yang telah diisi akan disimpan kedalam database. *System Flow* untuk *Registrasi Vendor* dapat dilihat pada Gambar 3.16.



Gambar 3. 16 *System Flow* Registrasi Vendor

### m. *System Flow Legality Vendor*

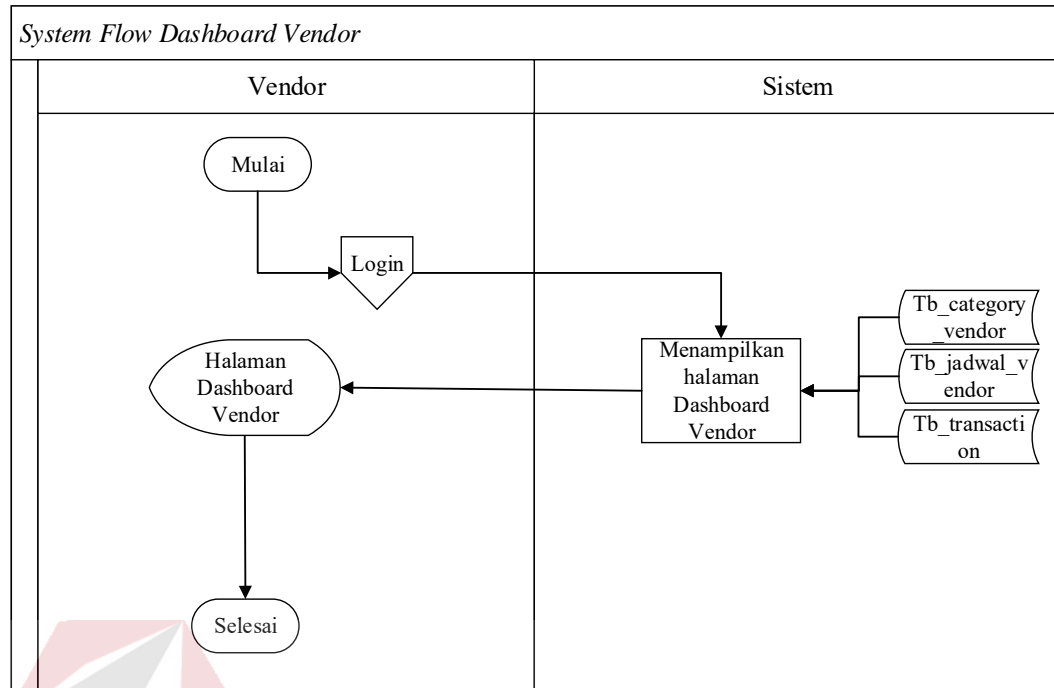
*Legality Vendor* merupakan proses yang dilakukan oleh *vendor* untuk menginputkan data legal *vendor* sebagai syarat mendaftar di aplikasi. Data tersebut berupa nomor NPWP, nama badan usaha, alamat, KTP, nama pimpinan, NIK pimpinan, dan alamat pimpinan. Berikut adalah *system flow legality vendor* yang dapat dilihat pada Gambar 3.17.



Gambar 3. 17 *System Flow Legality Vendor*

#### n. *System Flow Dashboard Pada Vendor*

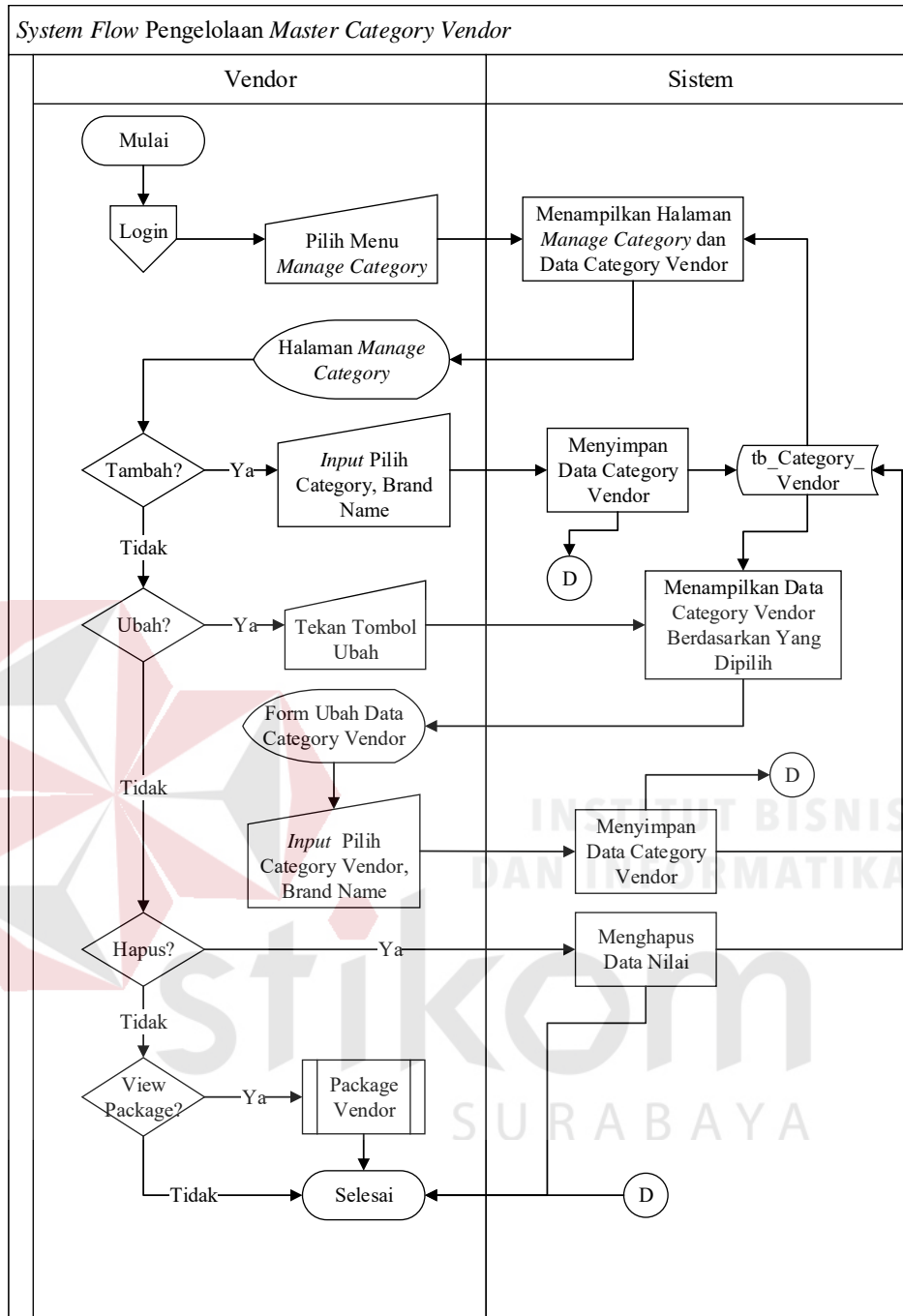
*System flow Dashboard* pada tampilan vendor menggambarkan tentang informasi peringkat vendor, laporan member vendor, laporan member *customer*, laporan transaksi, dan laporan pendapatan. *Dashboard* pada vendor dapat dilihat pada Gambar 3.18.



Gambar 3. 18 *System Flow Dashboard* Pada Vendor

#### o. *System Flow Master Kategori Vendor*

*System Flow Master Category Vendor* merupakan proses pengelolaan master yang dilakukan oleh *vendor* untuk memilih kategori *vendor* yang didalamnya terdapat pengelolaan data yang meliputi tambah data, ubah data dan hapus data. *System Flow* untuk *Master Category Vendor* dapat dilihat pada Gambar 3.19.

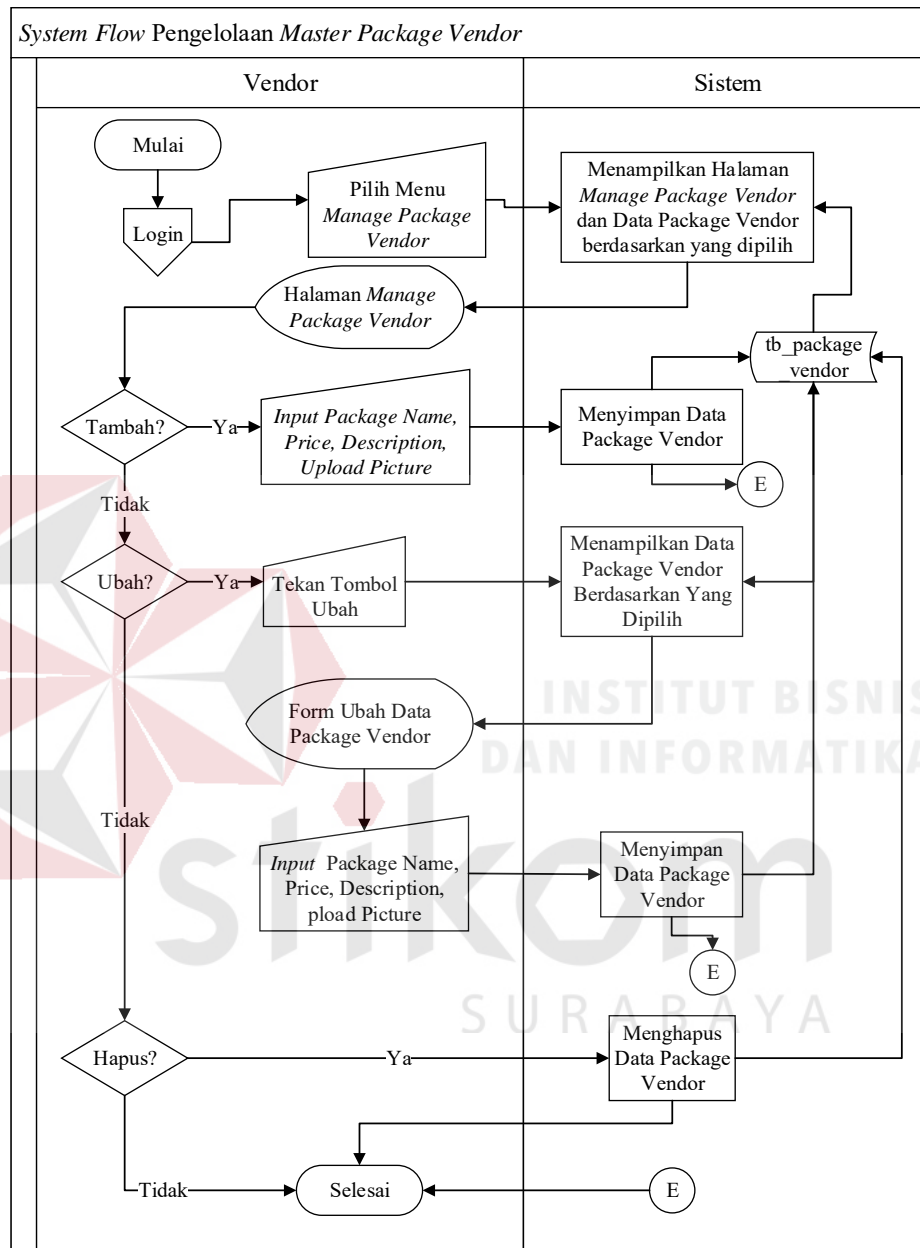


Gambar 3. 19 *System Flow* Pengelolaan Mastrer Kategori Vendor

**p. *System Flow* Master Paket Vendor**

*System Flow* Master Paket Vendor merupakan proses pengelolaan master yang dilakukan oleh *vendor* untuk membuat paket *vendor* yang didalamnya terdapat

pengelolaan data yang meliputi tambah data, ubah data dan hapus data. *System Flow* untuk *Master Package Vendor* dapat dilihat pada Gambar 3.20.



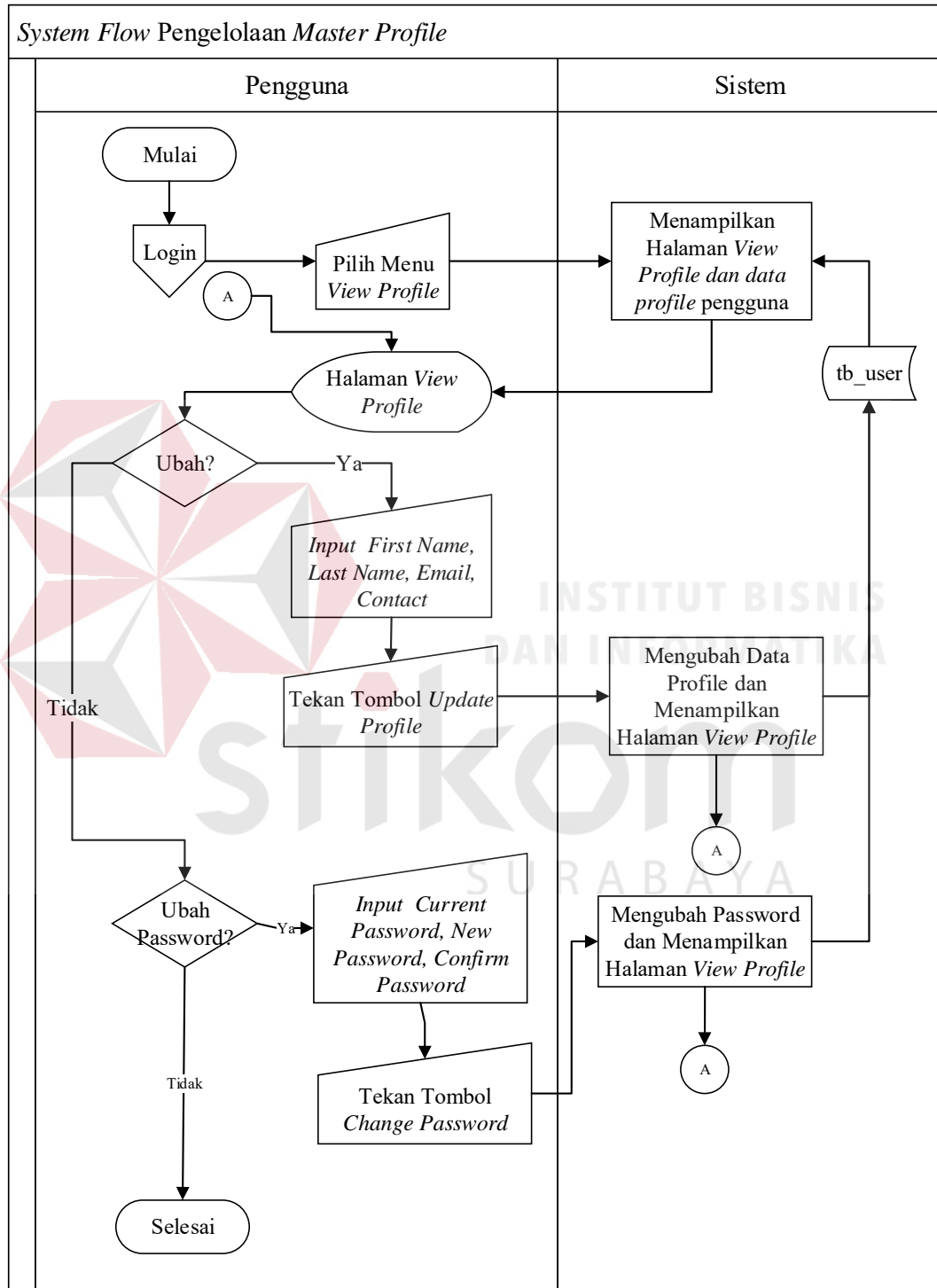
Gambar 3. 20 *System Flow* Master Package Vendor

#### q. *System Flow* Pengelolaan Profile Vendor

*System Flow* Pengelolaan *Profile Vendor* merupakan proses pengelolaan master yang dilakukan oleh *vendor* untuk menambah, mengubah dan menghapus

profil *vendor* baik nama, email, kontak *vendor* serta untuk mengubah *password*.

*System Flow* untuk Master Profile dapat dilihat pada Gambar 3.21.

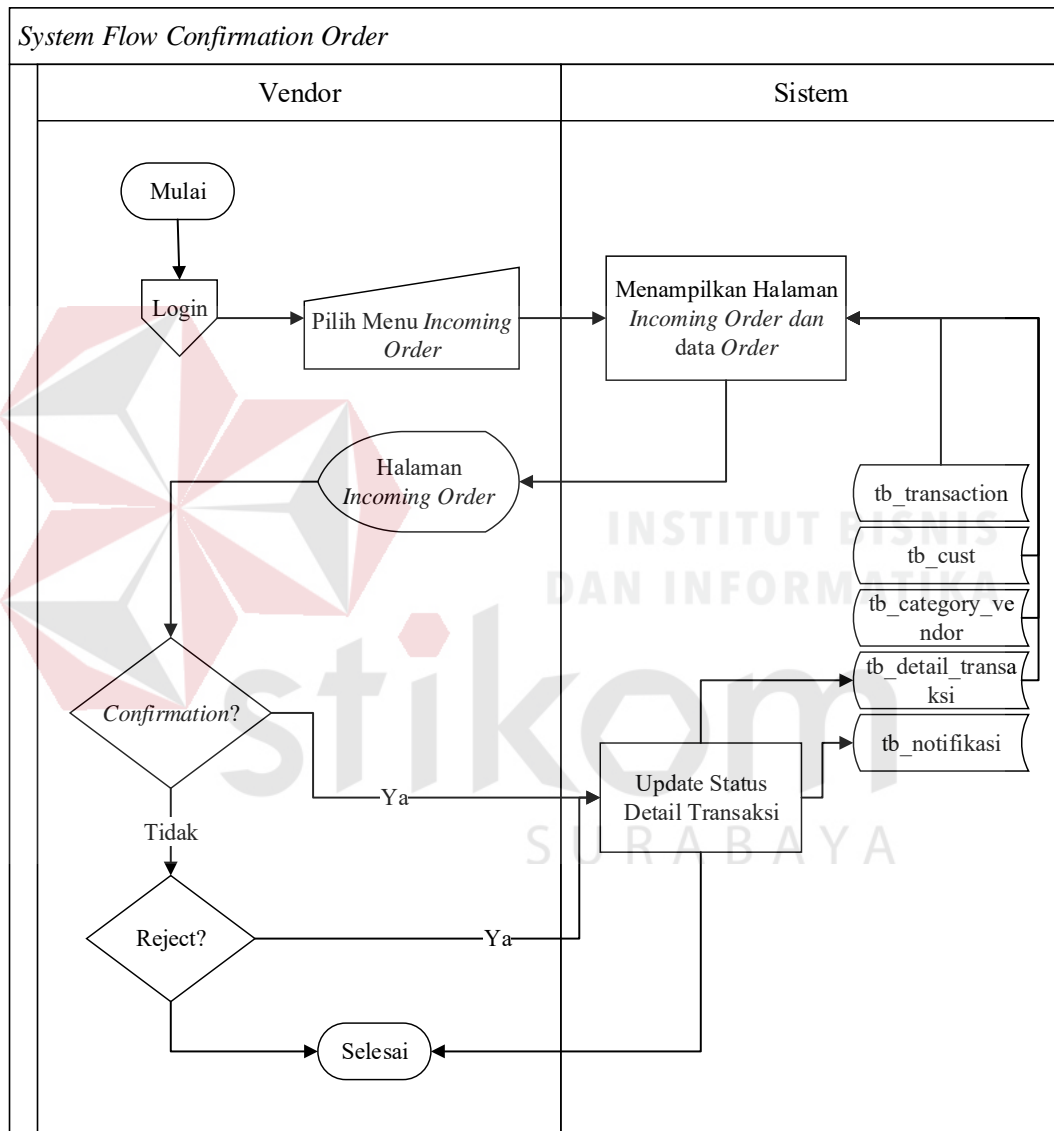


Gambar 3. 21 *System Flow* Pengelolaan Profile Vendor



### r. System Flow Konfirmasi Order

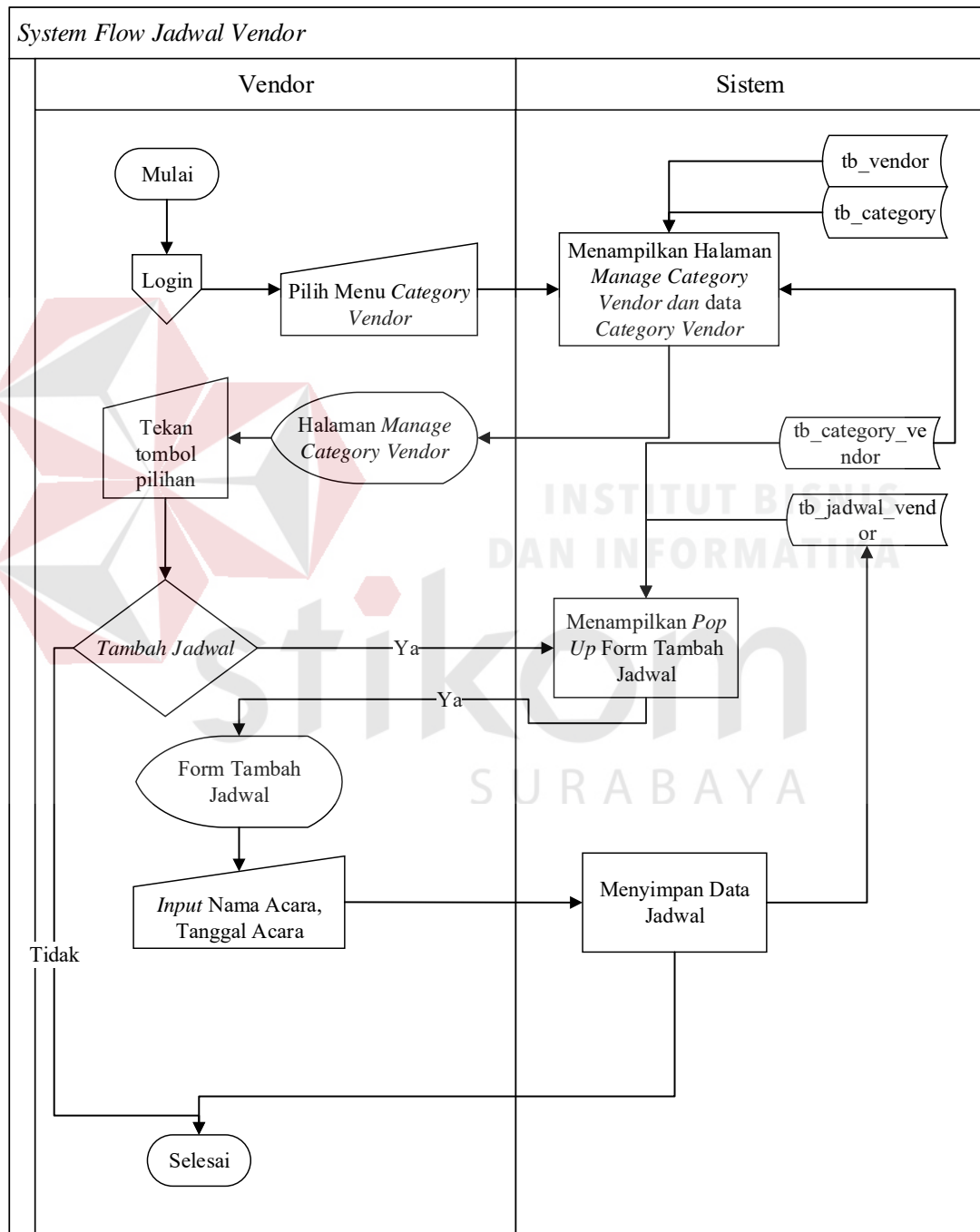
Konfirmasi *order* merupakan proses melakukan konfirmasi pemesanan yang dilakukan oleh *customer*. Berikut adalah *system flow* konfirmasi *order* yang dapat dilihat pada Gambar 3.22.



Gambar 3. 22 *System Flow* Konfirmasi *Order*

### s. *System Flow Jadwal Vendor*

*System Flow Jadwal Vendor* merupakan proses yang dilakukan vendor untuk mencatat/menambahkan jadwal vendor dengan menginputkan nama acara dan tanggal acara. *System Flow Jadwal Vendor* dapat dilihat pada Gambar 3.23.

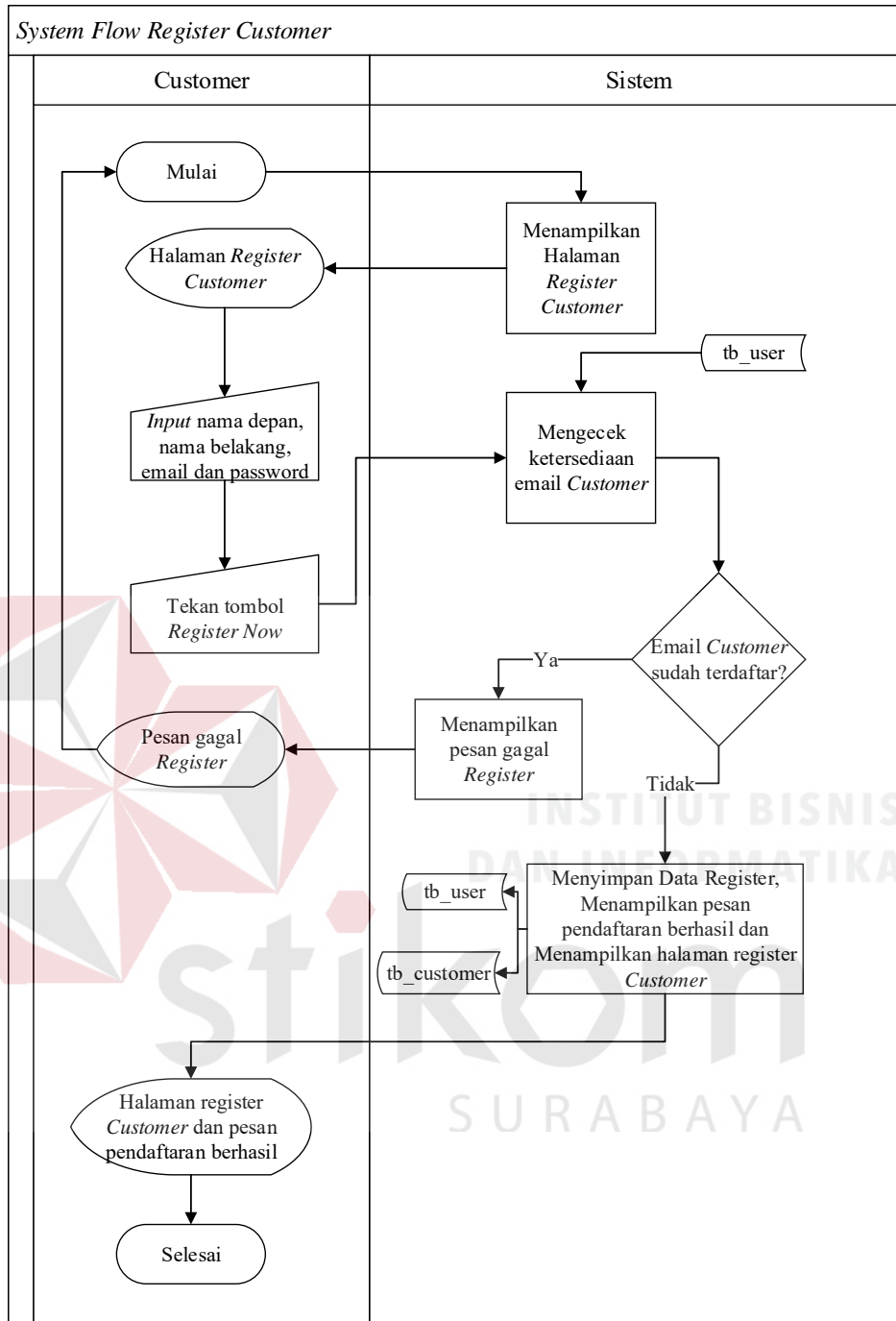


Gambar 3. 23 *System Flow Jadwal Vendor*

**t. *System Flow Registrasi Customer***

*System Flow Registrasi Customer* merupakan proses yang dilakukan oleh *customer* yang mendaftar melalui sistem untuk memilih *vendor* atau paket *vendor* yang diinginkan berdasarkan kriteria yang dipilih dengan menginputkan identitas *customer*, *email*, dan *password*. Data yang telah diisi akan disimpan kedalam *database*. *System Flow* untuk *Registrasi Customer* dapat dilihat pada Gambar 3.24.





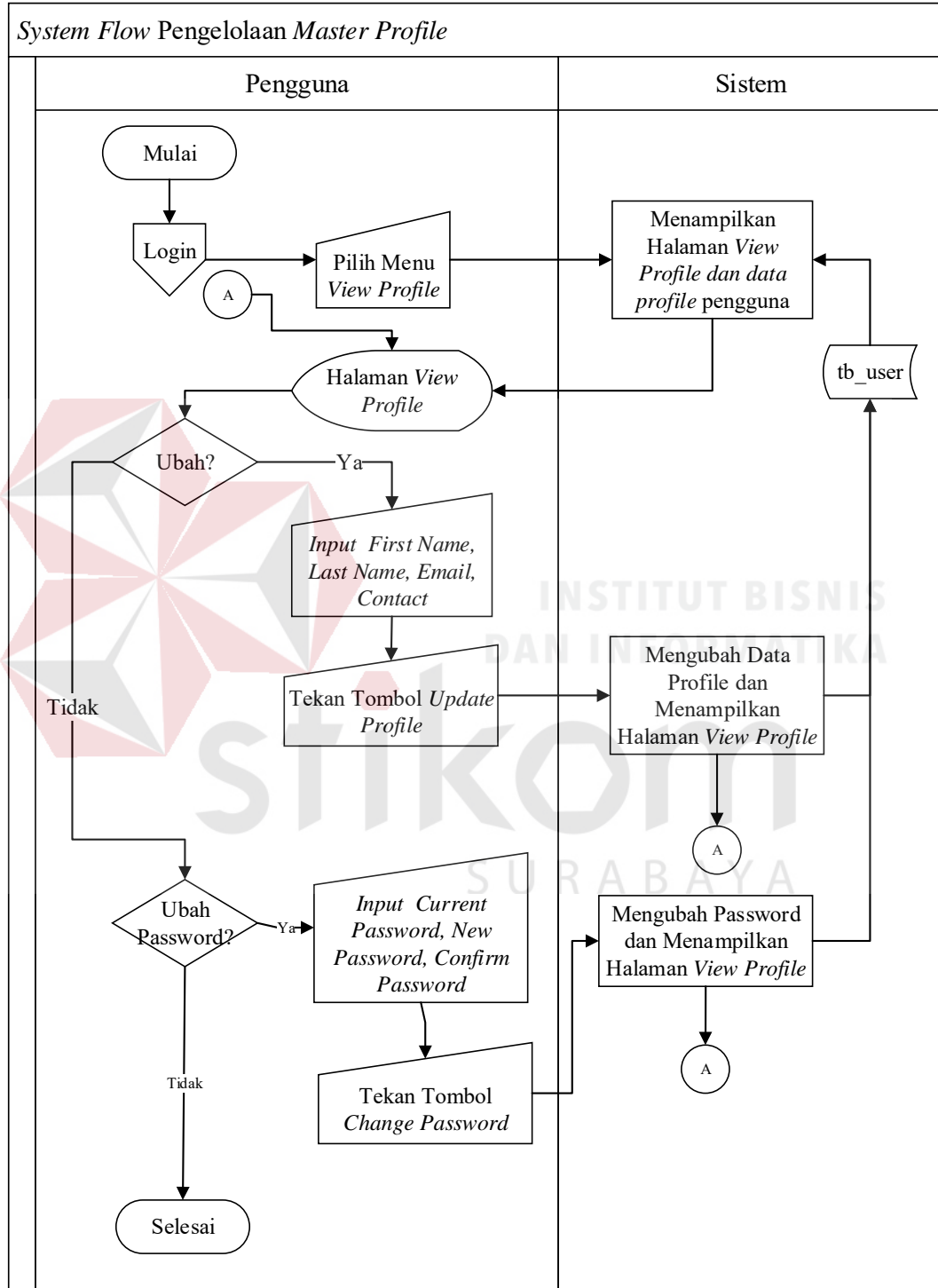
Gambar 3. 24 *System Flow Register Customer*

**u. *System Flow Pengelolaan Profile Customer***

*System Flow Profile Customer* merupakan proses pengelolaan master yang dilakukan oleh pengguna untuk menambah, mengubah dan menghapus profil

customer baik nama, email, kontak customer serta untuk mengubah password.

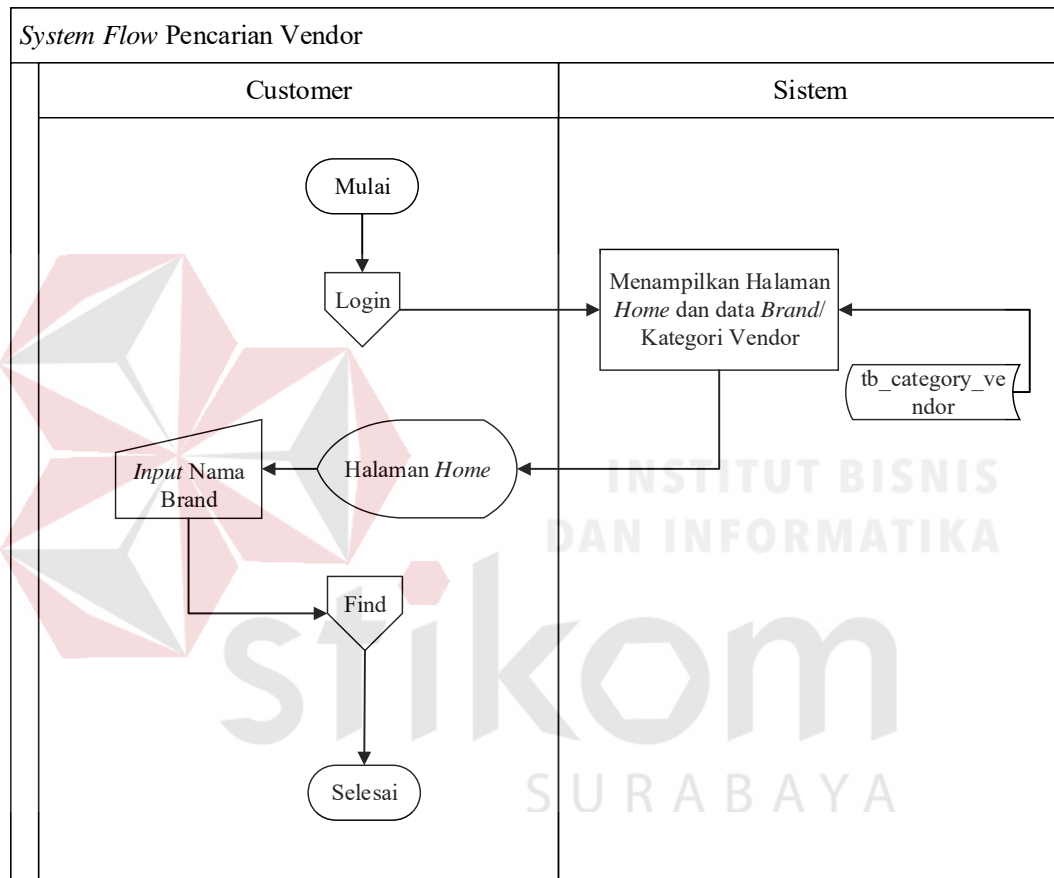
System Flow untuk Master Profile dapat dilihat pada Gambar 3.25.



Gambar 3. 25 System Flow Pengelolaan Profile Customer

#### v. *System Flow Pencarian Vendor*

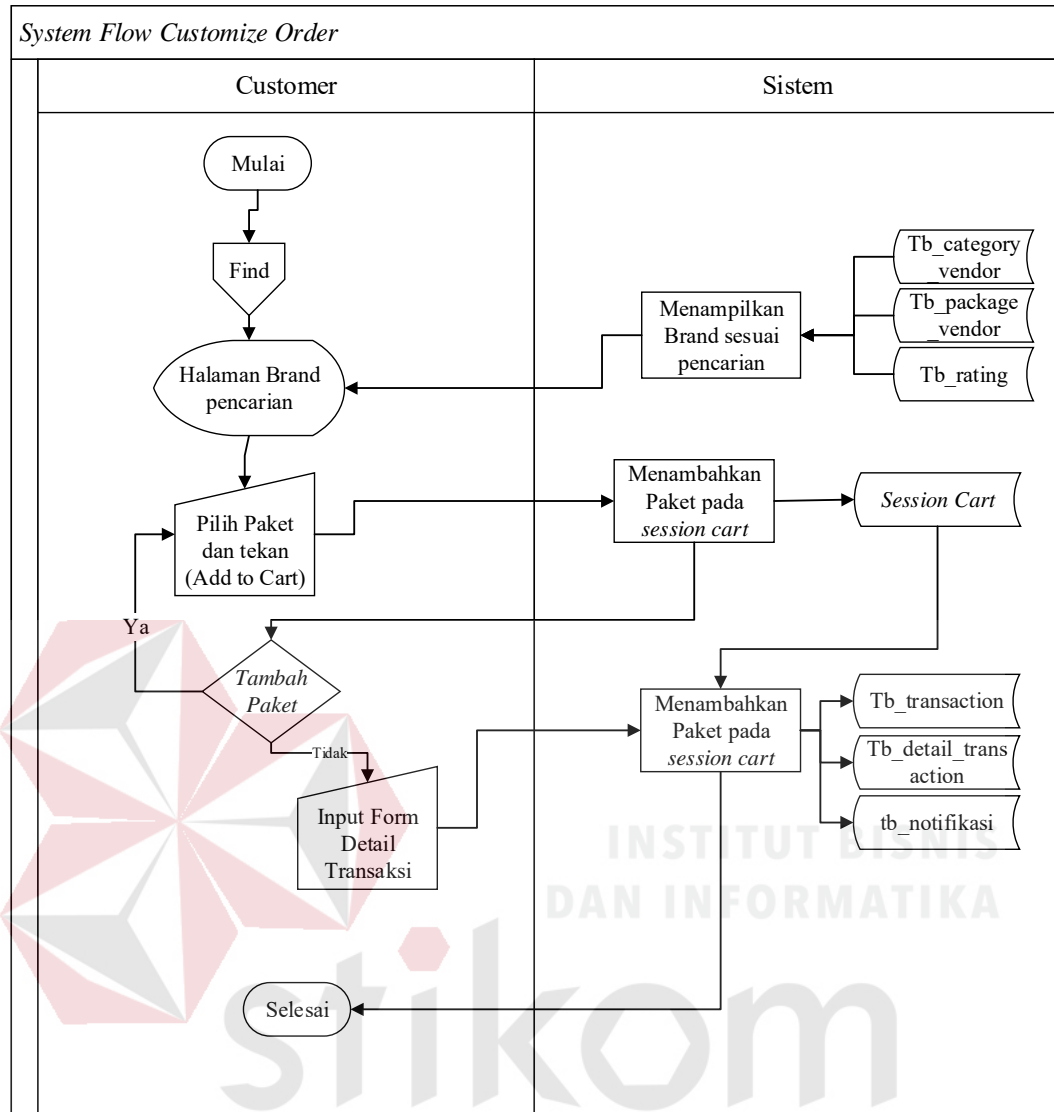
*System Flow* Pencarian Vendor merupakan proses yang dilakukan oleh *customer* untuk mencari vendor yang diinginkan dengan menginputkan *brand/kategori* vendor. *System Flow Pencarian Vendor* dapat dilihat pada Gambar 3.26.



Gambar 3. 26 *System Flow* Pencarian Vendor

#### w. *System Flow Customize Order Vendor*

*System Flow Customize Order* Vendor merupakan proses yang dilakukan oleh *customer* untuk memesan paket vendor secara *customize* sesuai dengan keinginan *customer* atau bukan rekomendasi dari sistem. *System Flow Customize Order* Vendor dapat dilihat pada Gambar 3.27.



Gambar 3. 27 *System Flow Customize Order Vendor*

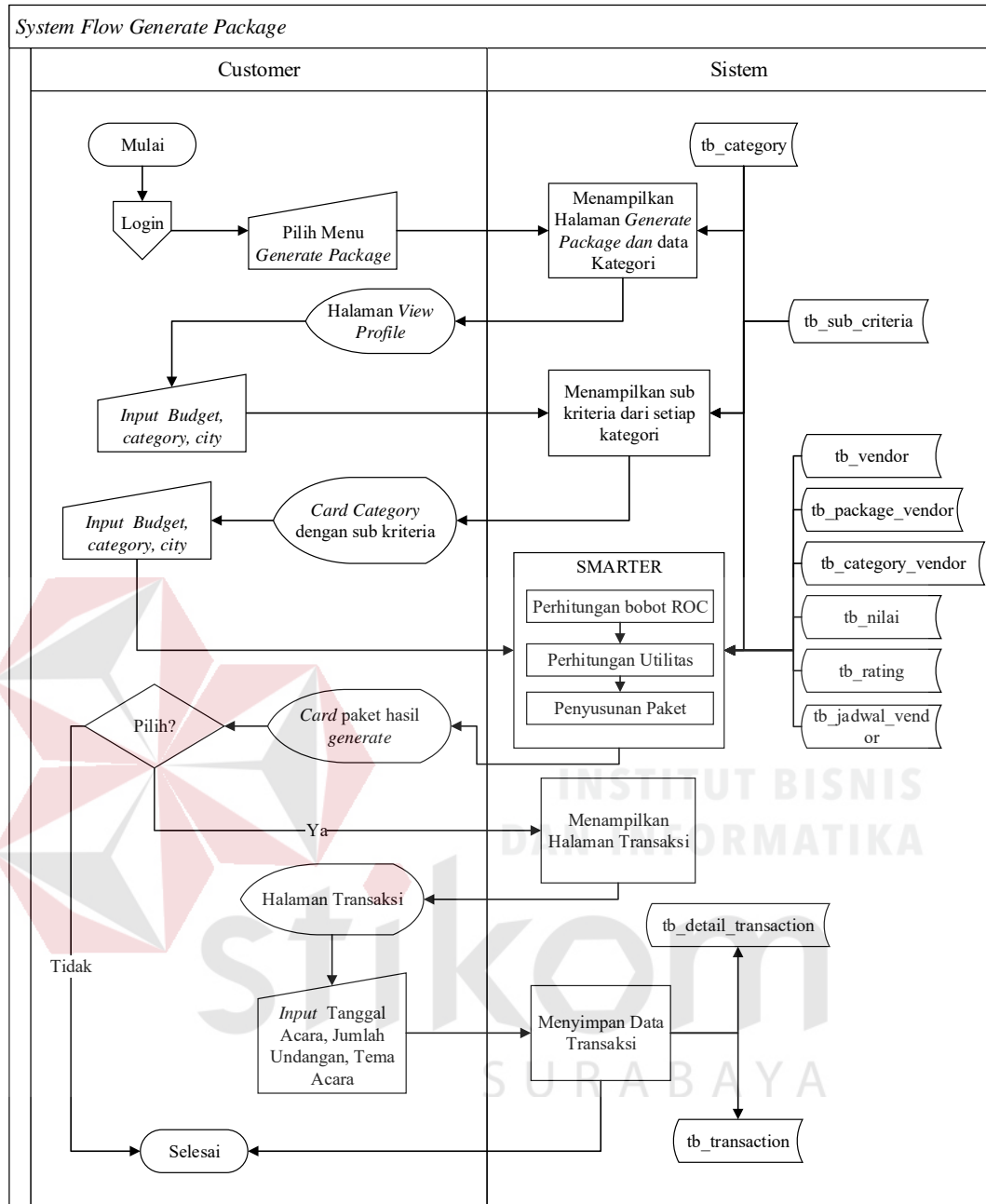
#### x. *System Flow Generate Paket dan Pemesanan Paket*

*Generate* paket dan pemesanan paket merupakan proses yang dilakukan oleh *customer* untuk mencari paket yang diinginkan berdasarkan kriteria yang dipilih. Setelah di *generate* maka akan muncul pilihan paket untuk di pesan oleh *customer*. *Generate* paket ini dilakukan dengan menerapkan metode SMARTER dengan mengambil alternatif (paket vendor), kriteria (kategori), dan sub kriteria (*rating*, *budget*, pengalaman). Kemudian *customer* memberikan urutan prioritas

pada kategori dan sub kriteria. Setelah itu sistem menghitung rumus pembobotan ROC berdasarkan data urutan prioritas yang telah ditentukan *customer*. Hasil dari perhitungan tersebut digunakan untuk menghitung utilitas dari setiap alternatif yang ada. Setelah menghitung utilitas setiap alternatif, maka sistem akan menampilkan 3 (tiga) kombinasi paket dengan nilai utilitas tertinggi untuk ditawarkan kepada *customer*. *System Flow* untuk *Generate* Paket dan Pemesanan Paket dapat dilihat pada Gambar 3.28.



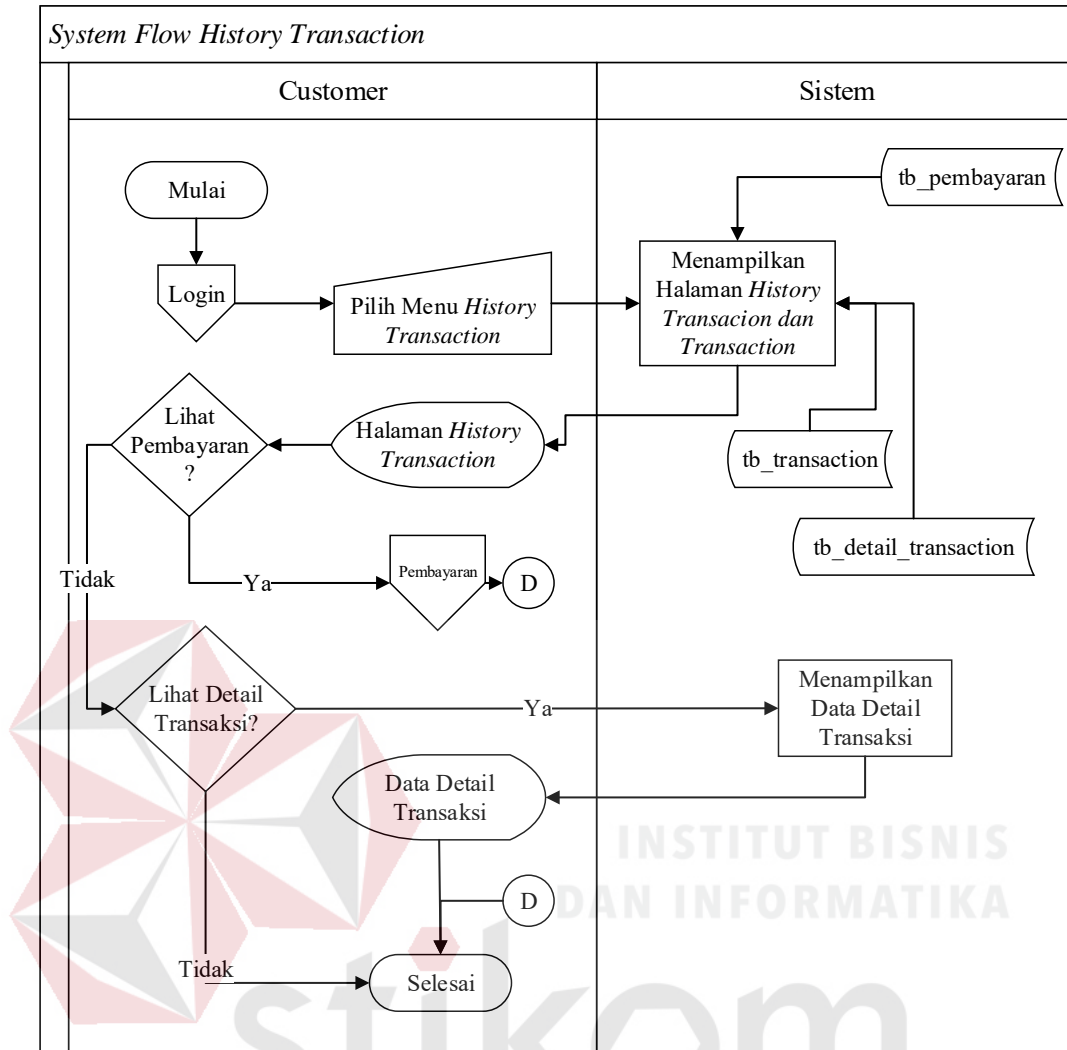




Gambar 3. 28 *System Flow Generate dan Pemesanan Paket*

#### y. *System Flow Histori Transaksi*

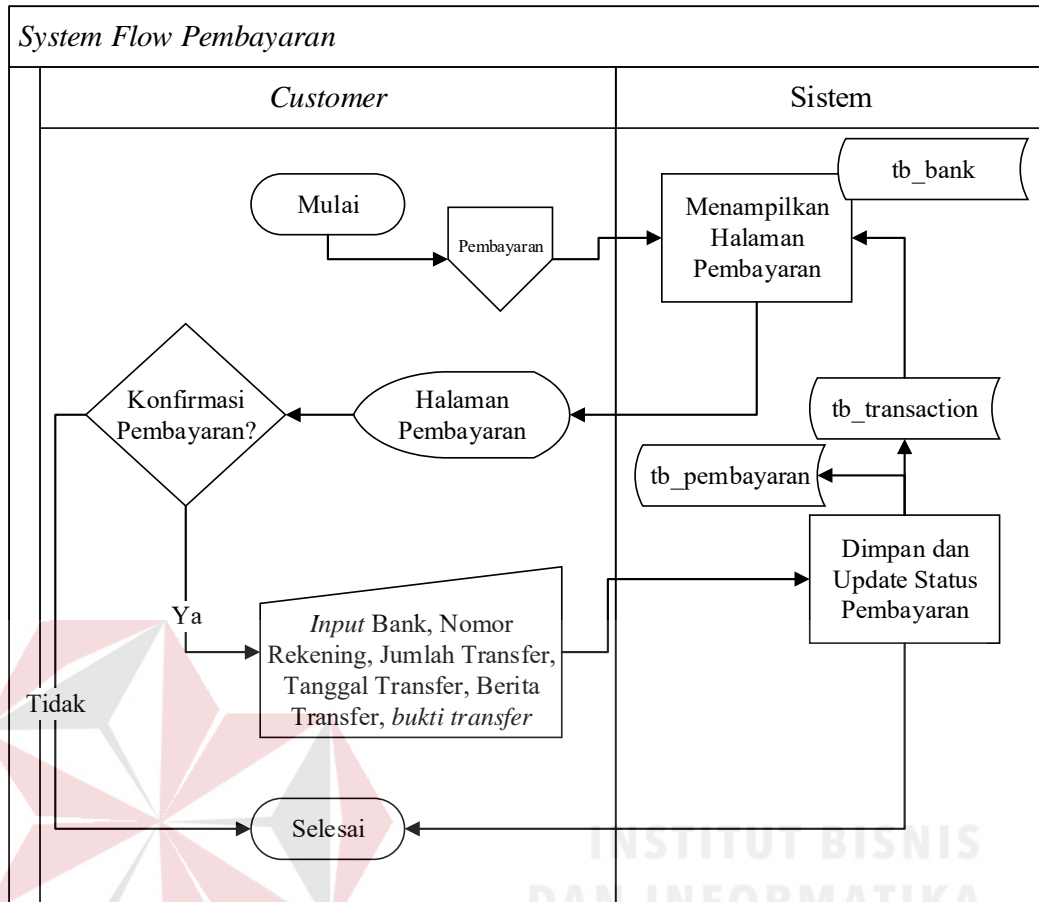
Histori Transaksi merupakan proses untuk menampilkan riwayat transaksi atau pemesanan yang dilakukan oleh *customer*. Berikut adalah *system flow* histori transaksi yang dapat dilihat pada Gambar 3.29.



Gambar 3. 29 *System Flow* Histori Transaksi

#### z. *System Flow* Konfirmasi Pembayaran *Customer*

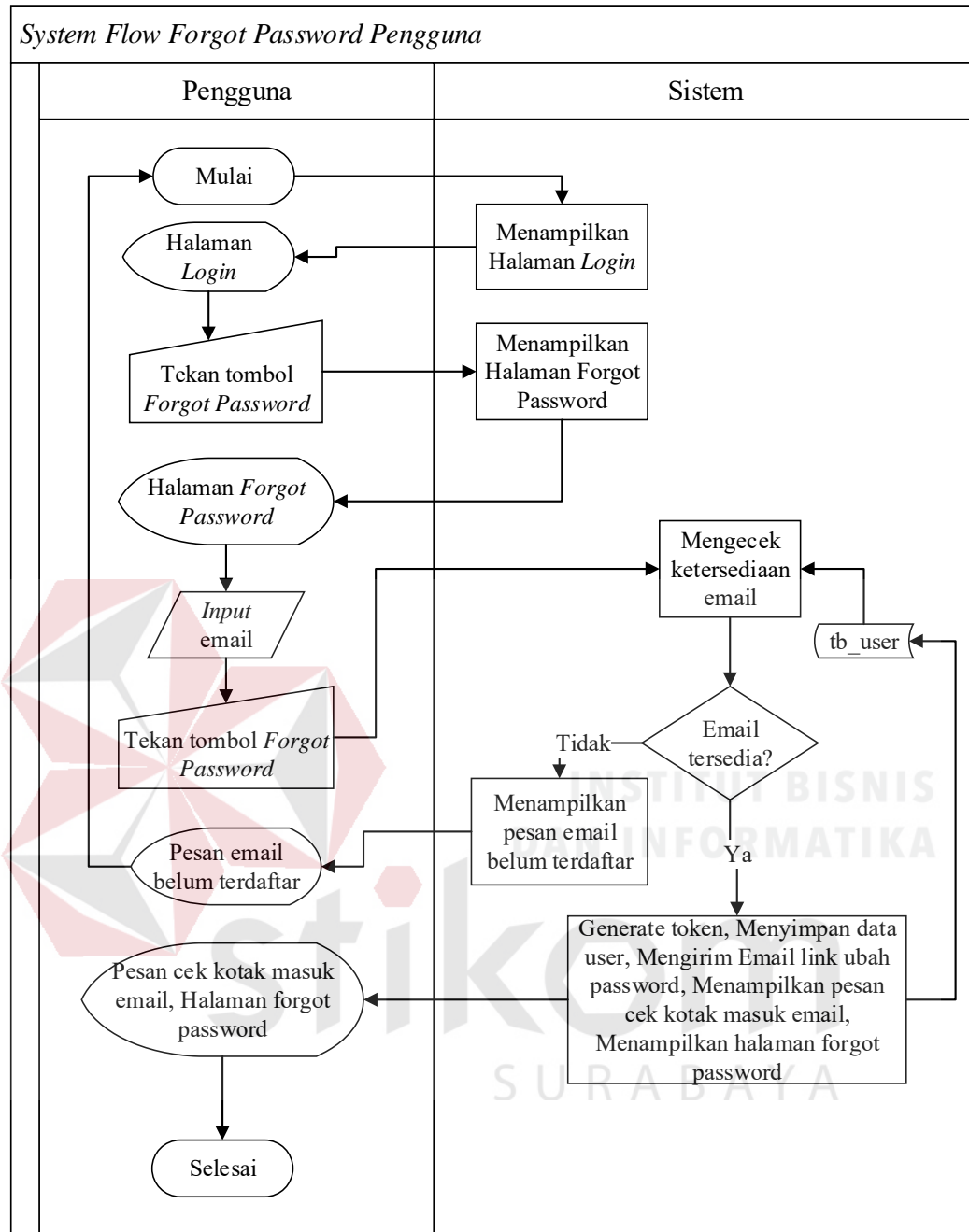
Konfirmasi pembayaran merupakan proses yang dilakukan oleh *customer* dengan menginputkan *bank*, nomor rekening, jumlah transfer, tanggal transfer, berita transfer dan meng-*upload* bukti transfer. Berikut adalah *system flow* konfirmasi pembayaran yang dilakukan *customer* dapat dilihat pada Gambar 3.30.



Gambar 3. 30 *System Flow* Konfirmasi Pembayaran *Customer*

#### aa. *System Flow* Forgot Password Pengguna

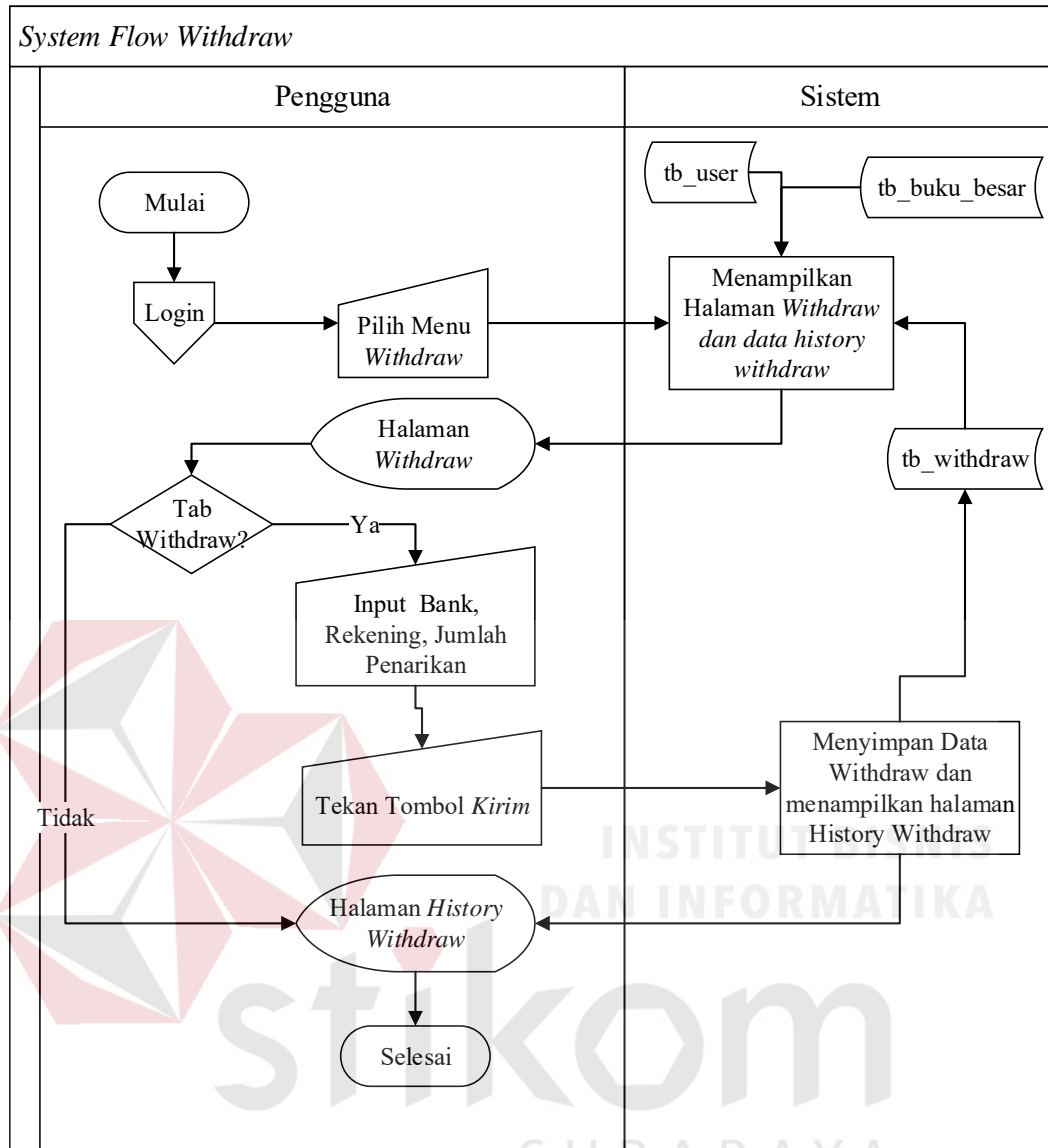
*System Flow* *Forgot Password* merupakan proses yang dilakukan oleh pengguna sistem jika pengguna lupa akan passwordnya. Pengguna dapat mengganti *password* tersebut dengan menekan tombol *forgot password* yang ada pada halaman *login*. Pengguna dapat mengganti *password* lama dengan menginputkan *email* lama dan *password* baru. Data yang telah diisi akan disimpan kedalam *database*. *System Flow* untuk *Forgot Password* dapat dilihat pada Gambar 3.31.



Gambar 3. 31 *System Flow Forgot Password Pengguna*

#### **bb. System Flow Withdraw**

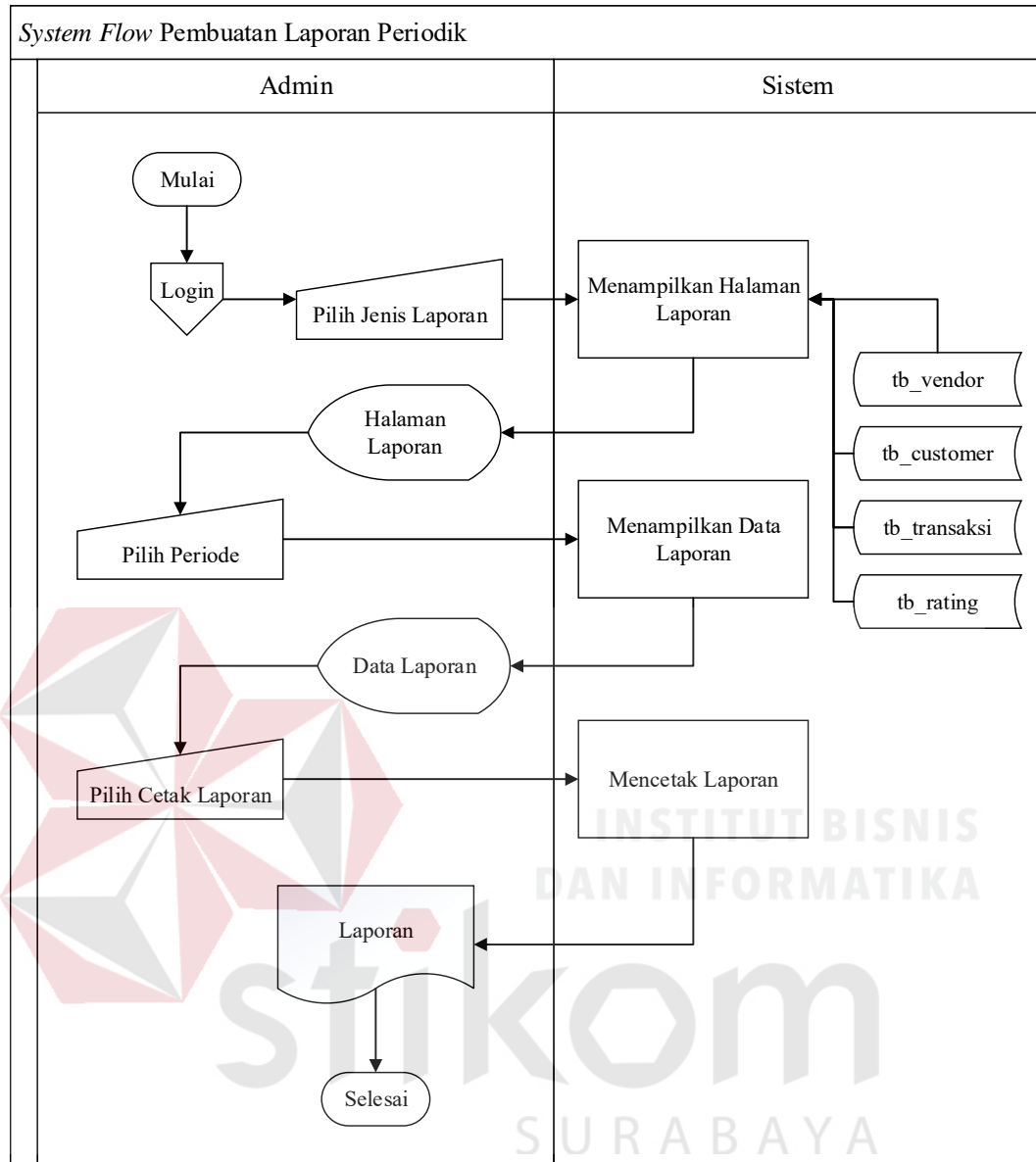
*System Flow Withdraw* merupakan proses penarikan dana yang dilakukan oleh *customer* dan vendor ketika memiliki saldo. *System Flow Withdraw* dapat dilihat pada Gambar 3.32.



Gambar 3. 32 *System Flow Withdraw*

**cc. *System Flow* Pembuatan Laporan Periodik (Member Vendor, Member Customer, Transaksi, Pendapatan, Peringkat Vendor)**

*System Flow* Pembuatan laporan merupakan proses yang dilakukan proses untuk membuat laporan terkait dengan laporan member vendor, member *customer*, transaksi, pendapatan dan peringkat vendor yang dilakukan oleh admin. *System Flow* Pembuatan Laporan dapat dilihat pada Gambar 3.33.

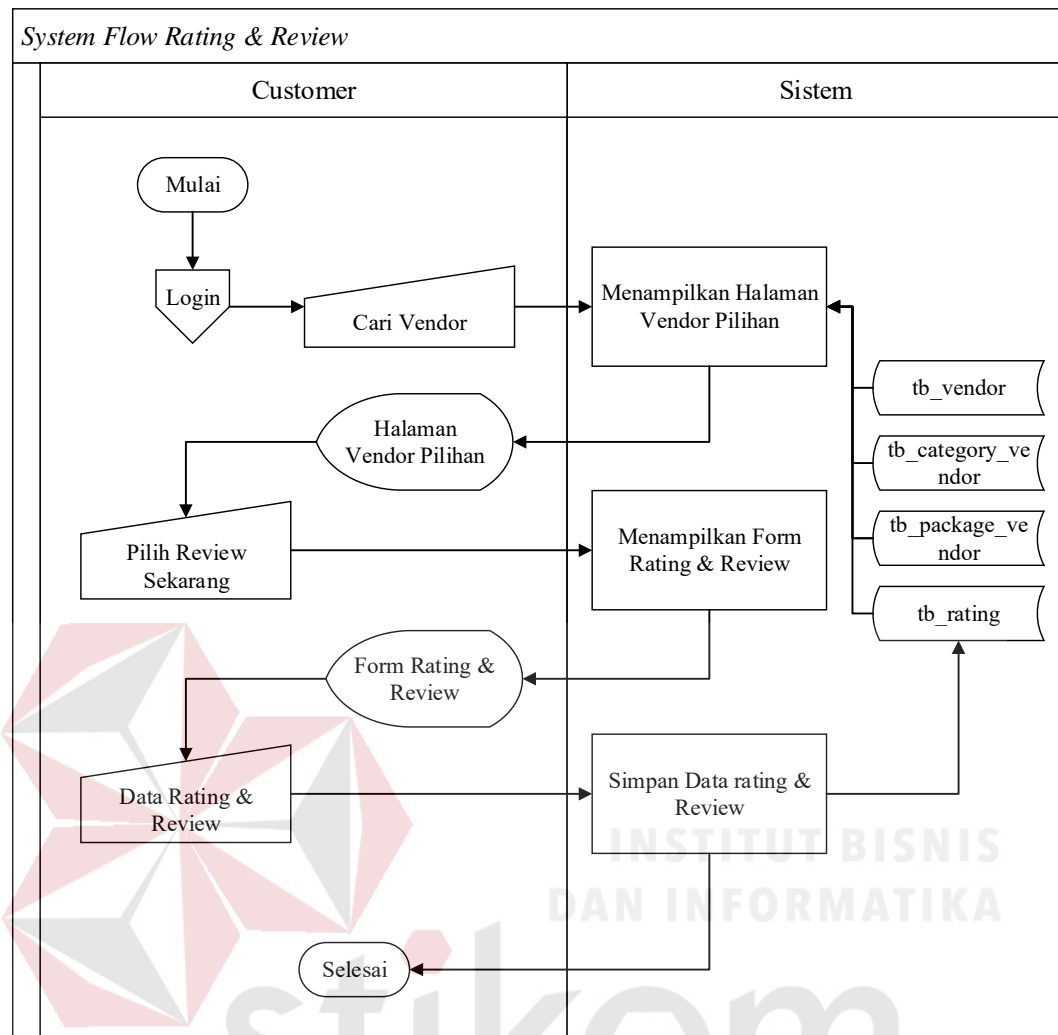


Gambar 3. 33 *System Flow* Pembuatan Laporan Periodik

#### **dd. *System Flow Rating & Review***

*System Flow Rating & Review* merupakan proses yang dilakukan untuk memberikan *rating & review* terhadap vendor yang dilakukan oleh *customer*.

*System Flow Flow Rating & Review* dapat dilihat pada Gambar 3.34.

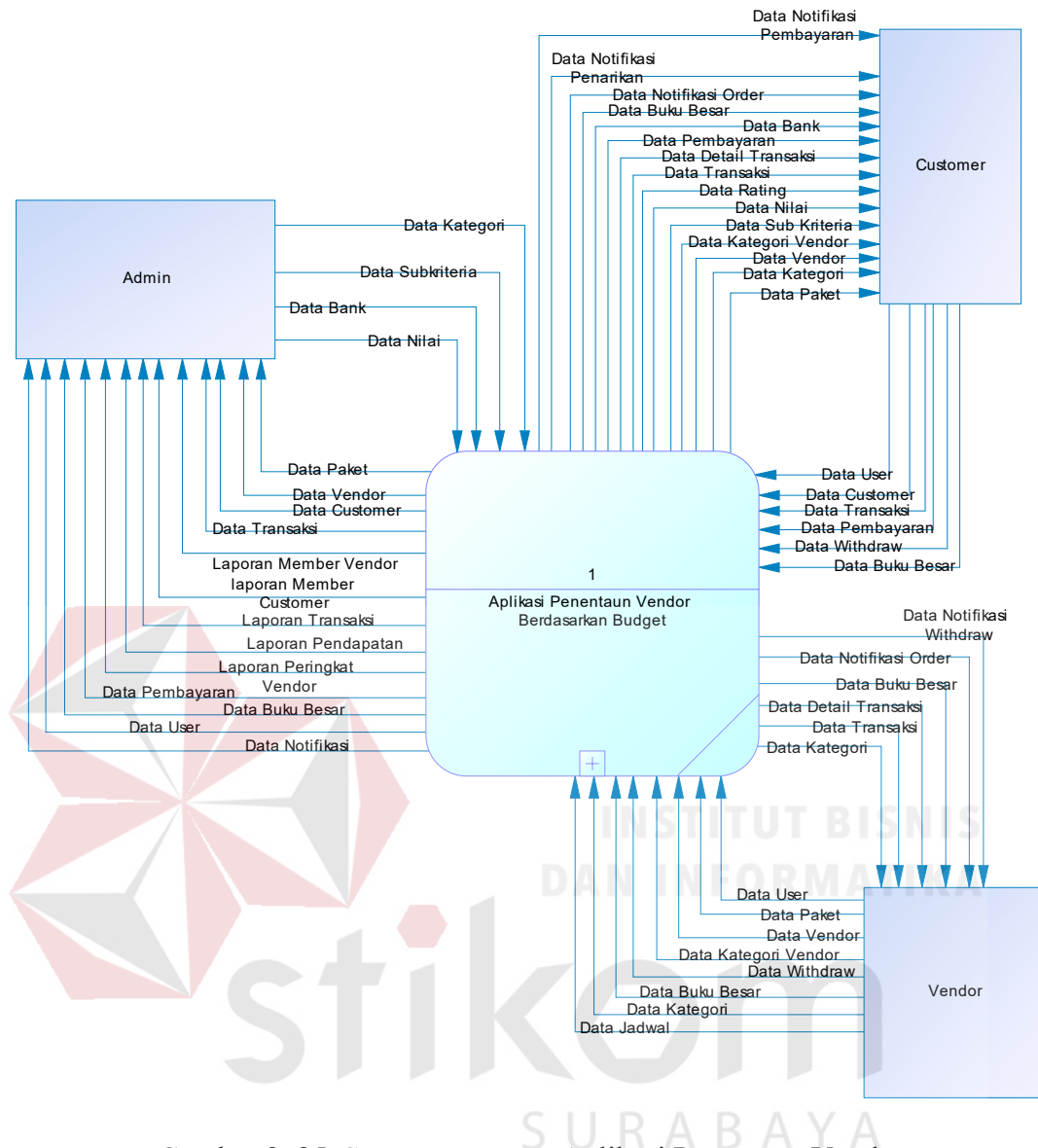


Gambar 3. 34 *System Flow Rating & Review*

## B. Data Flow Diagram

### i. Context Diagram

*Context Diagram* merupakan sebuah gambaran yang menampilkan secara menyeluruh mengenai aliran data pada Data Flow Diagram (DFD). *Context Diagram* atau diagram konteks juga merupakan level tertinggi dari DFD yang menggambarkan seluruh input ke sistem atau *output* dari sistem. Entitas yang berinteraksi dalam sistem yaitu Admin, Vendor, dan Customer. *Context Diagram* Aplikasi penentuan vendor pernikahan untuk usaha wedding Organizer dapat dilihat pada Gambar 3.35.



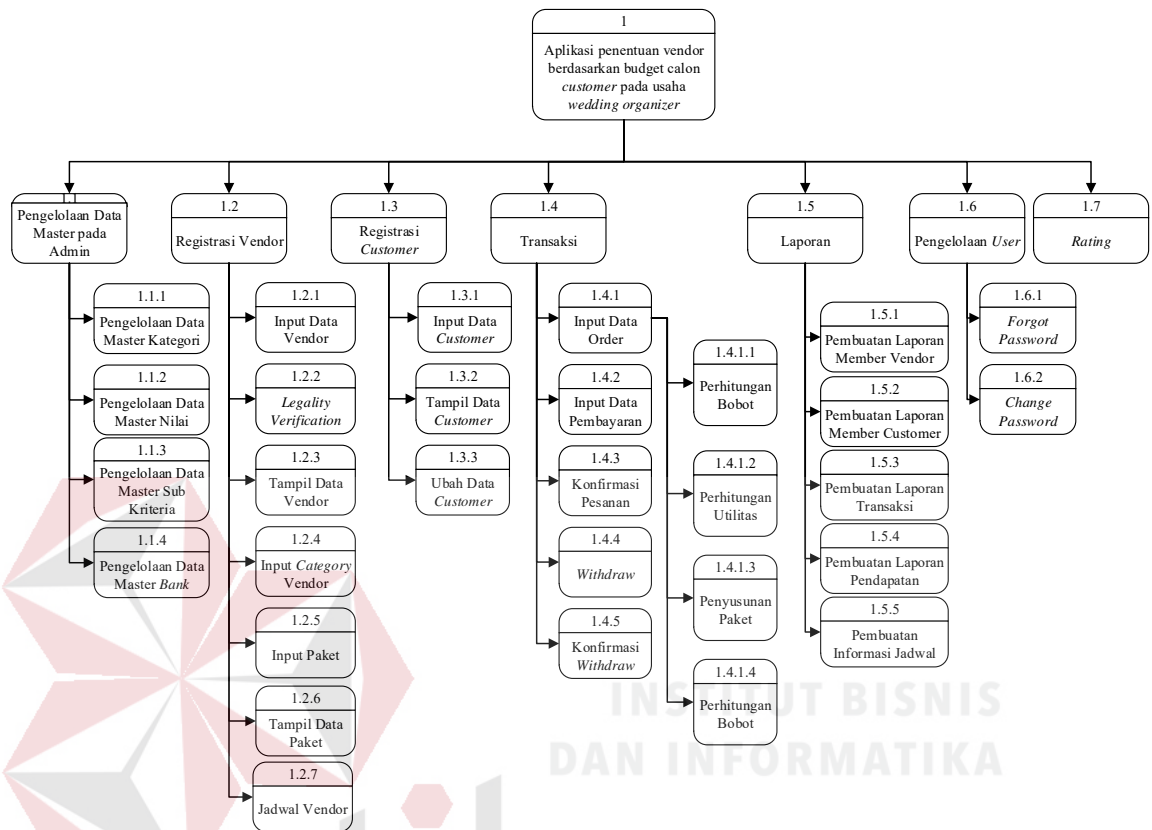
Gambar 3. 35 Context Diagram Aplikasi Penentuan Vendor

### i. Diagram Berjenjang

Diagram berjenjang merupakan sebuah bagan atau gambaran dari setiap proses-proses yang ada pada suatu aplikasi. Diagram berjenjang dari aplikasi penentuan *vendor* pernikahan pada usaha *wedding organizer* terdiri atas 4 proses besar yaitu pengelolaan data master, pendaftaran *vendor* baru, pendaftaran *customer* baru, transaksi, laporan, pengelolaan *user*, *rating*, dan notifikasi. Bagan dari diagram berjenjang ini akan digunakan sebagai pedoman dalam membuat Data



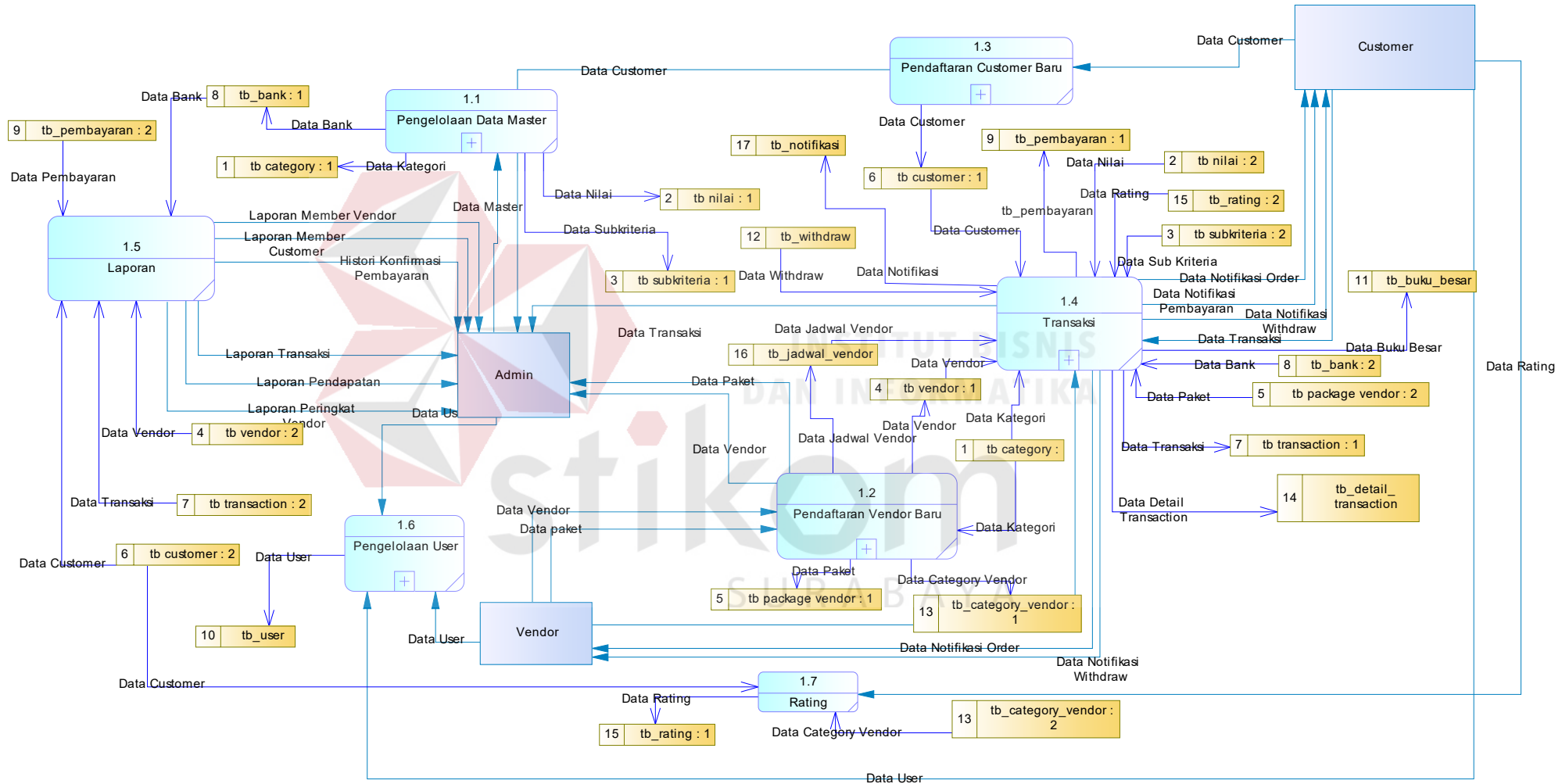
Flow Diagram. Diagram berjenjang aplikasi penentuan *vendor* pernikahan pada usaha *wedding Organizer* dapat dilihat di Gambar 3.36.



Gambar 3. 36 Diagram Berjenjang Aplikasi Penentuan Vendor Pernikahan

## ii. Data Flow Diagram level 0

Data flow diagram level 0 digambarkan secara global proses-proses yang ada pada aplikasi penentuan *vendor* pernikahan. DFD level 0 ini dimulai dari pengelolaan data master, pendaftaran *vendor* baru, pendaftaran *customer* baru, transaksi, laporan, pengelolaan *user*, *rating*, dan notifikasi. Data flow diagram level 0 dapat dilihat pada Gambar 3.37.



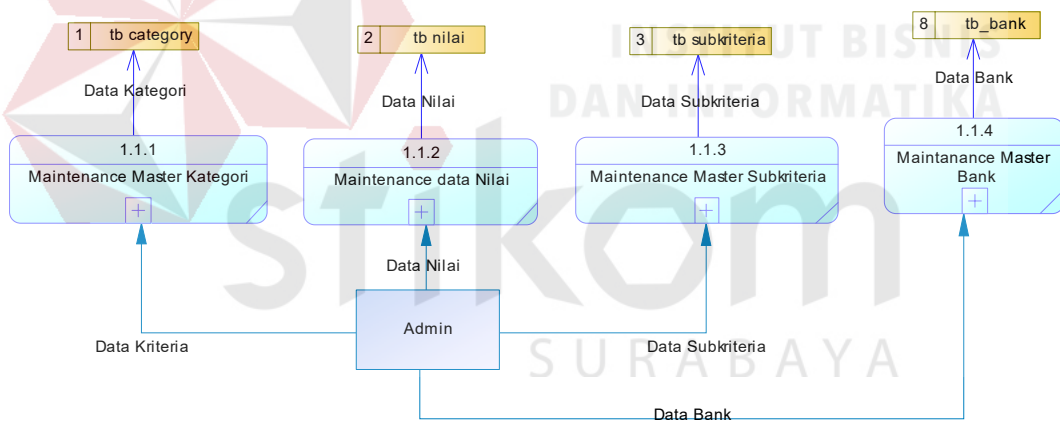
Gambar 3. 37 Data Flow Diagram Level 0

### iii. Data Flow Diagram Level 1

Data Flow Diagram Level 1 merupakan diagram yang menguraikan proses yang ada pada data flow diagram level 0. Pada DFD Level 1 digambarkan secara lebih detil proses-proses utama yakni pengelolaan data master, pendaftaran *vendor* baru, pendaftaran *customer* baru, transaksi, laporan, pengelolaan *user*, *rating*, dan notifikasi.

#### a. Pengelolaan Data Master

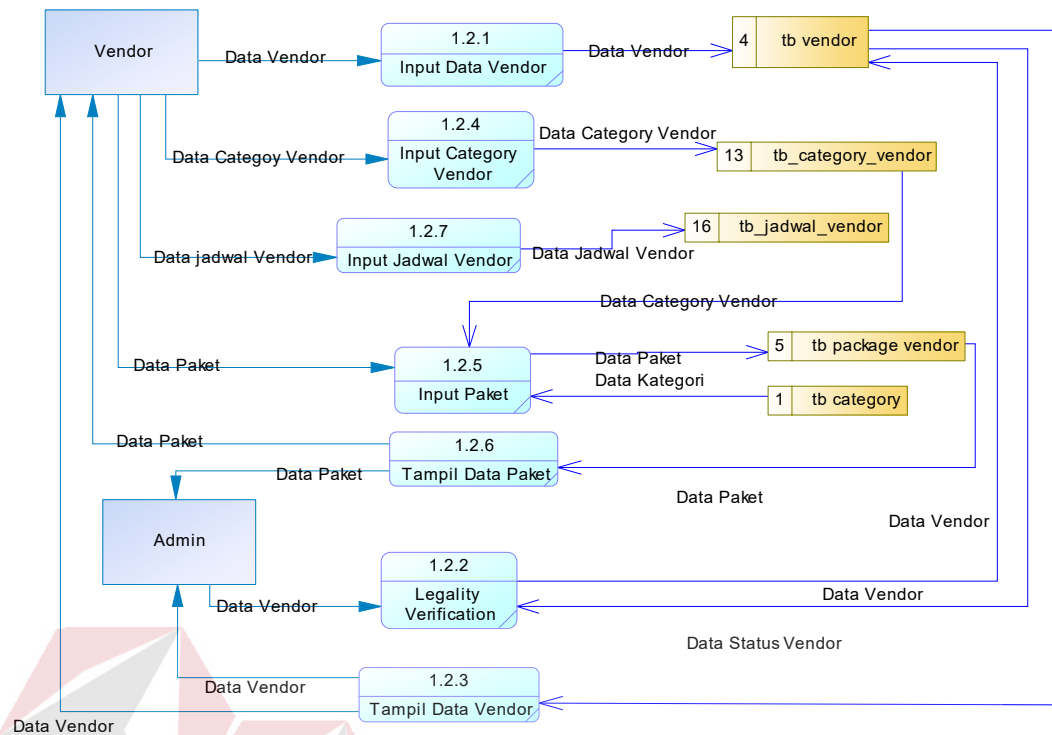
Pada level ini data yang mengalir untuk kegiatan pengelolaan data master kategori, nilai, sub kriteria dan *bank* yang akan dikelola dan disimpan pada masing-masing tabel master. Berikut DFD Level 1 Input Data Master dapat dilihat pada Gambar 3.38.



Gambar 3. 38 DFD Level 1 Input Data Master

#### b. Pendaftaran Vendor Baru

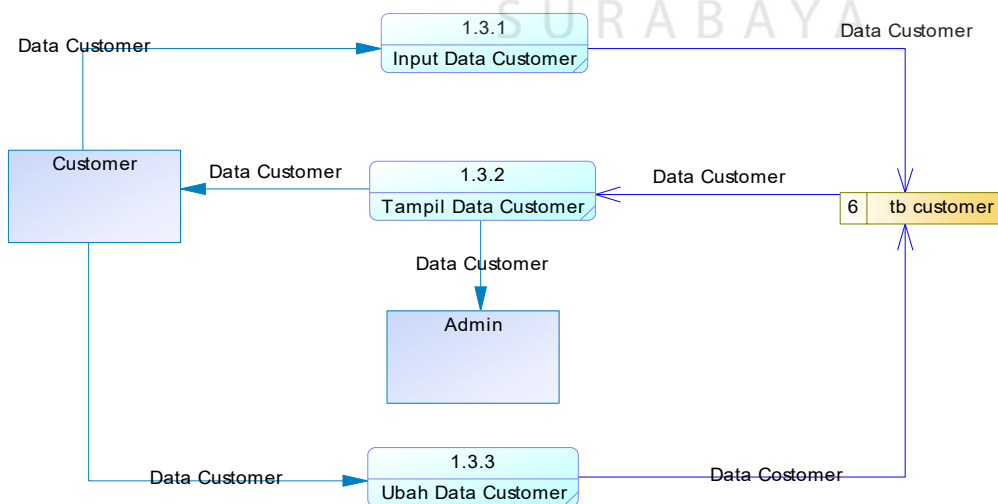
Pada level ini data yang mengalir untuk kegiatan pendaftaran *vendor* yaitu meliputi data-data *vendor*, *legality verification*, tampil data *vendor*, paket *vendor*, kategori *vendor* dan jadwal *vendor*. Berikut DFD Level 1 Pendaftaran *Vendor* Baru dapat dilihat pada Gambar 3.39.



Gambar 3. 39 DFD Level 1 Pendaftaran Vendor Baru

**c. Pendaftaran Customer Baru**

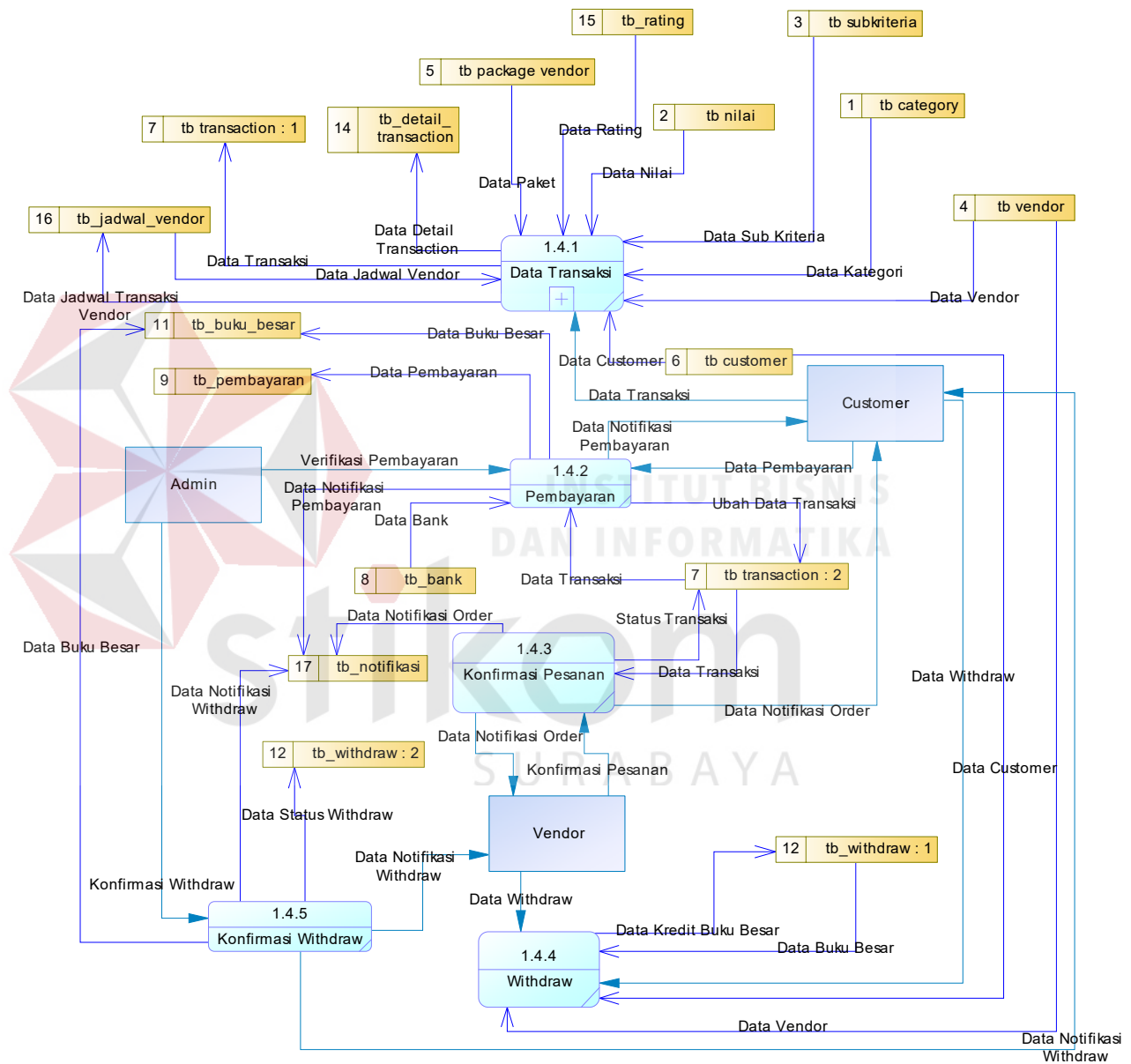
Pada level ini data yang mengalir untuk kegiatan pendaftaran *customer* yaitu data *customer* yang disimpan kedalam tabel *customer*. Berikut DFD Level 1 Pendaftaran *Customer* Baru dapat dilihat pada Gambar 3.40.



Gambar 3. 40 DFD Level 1 Pendaftaran Customer Baru

**d. Transaksi**

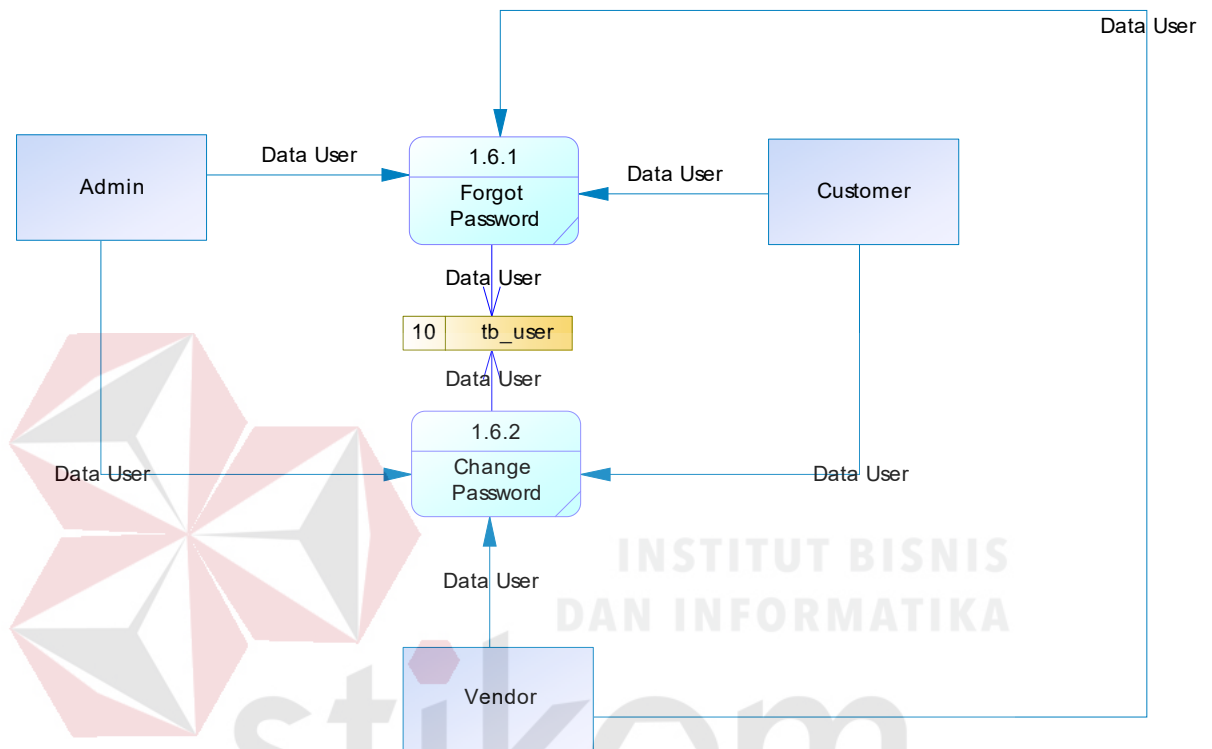
Pada level ini data yang mengalir untuk kegiatan transaksi yaitu meliputi data-data transaksi, pembayaran, konfirmasi pemesanan, *withdraw*, dan konfirmasi *withdraw*. Berikut DFD Level 1 Transaksi dapat dilihat pada Gambar 3.41.



Gambar 3. 41 DFD Level 1 Transaksi

### e. Pengelolaan User

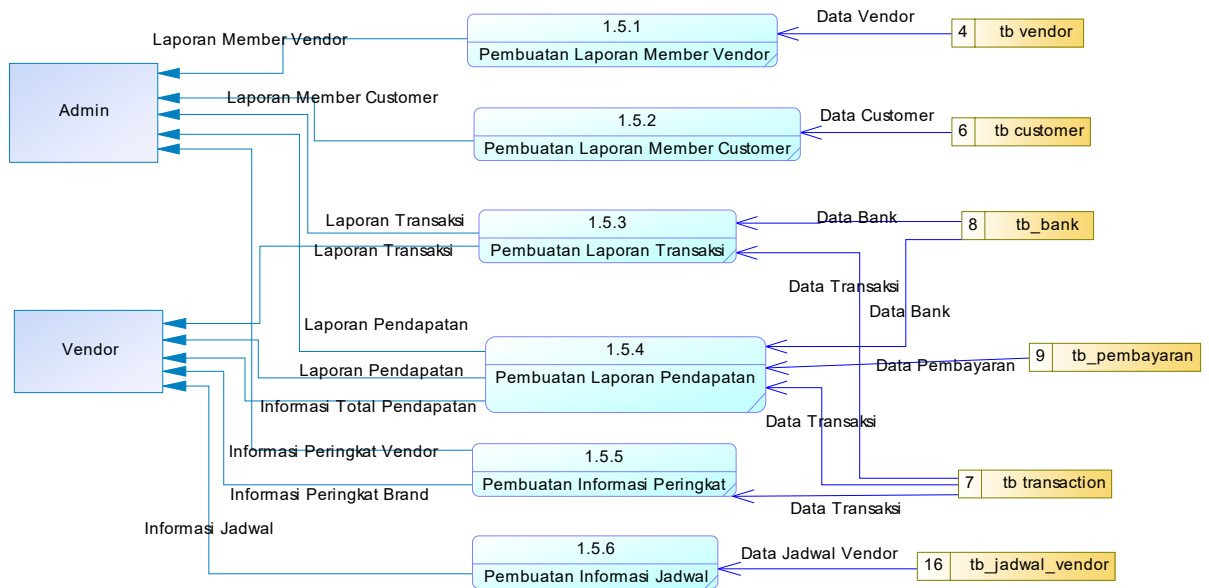
Pada level ini data yang mengalir untuk kegiatan pengelolaan user yaitu meliputi *forgot* dan *change password*.. Berikut DFD Level 1 Pengelolaan User dapat dilihat pada Gambar 3.42.



Gambar 3. 42 DFD Level 1 Pengelolaan User

### f. Laporan

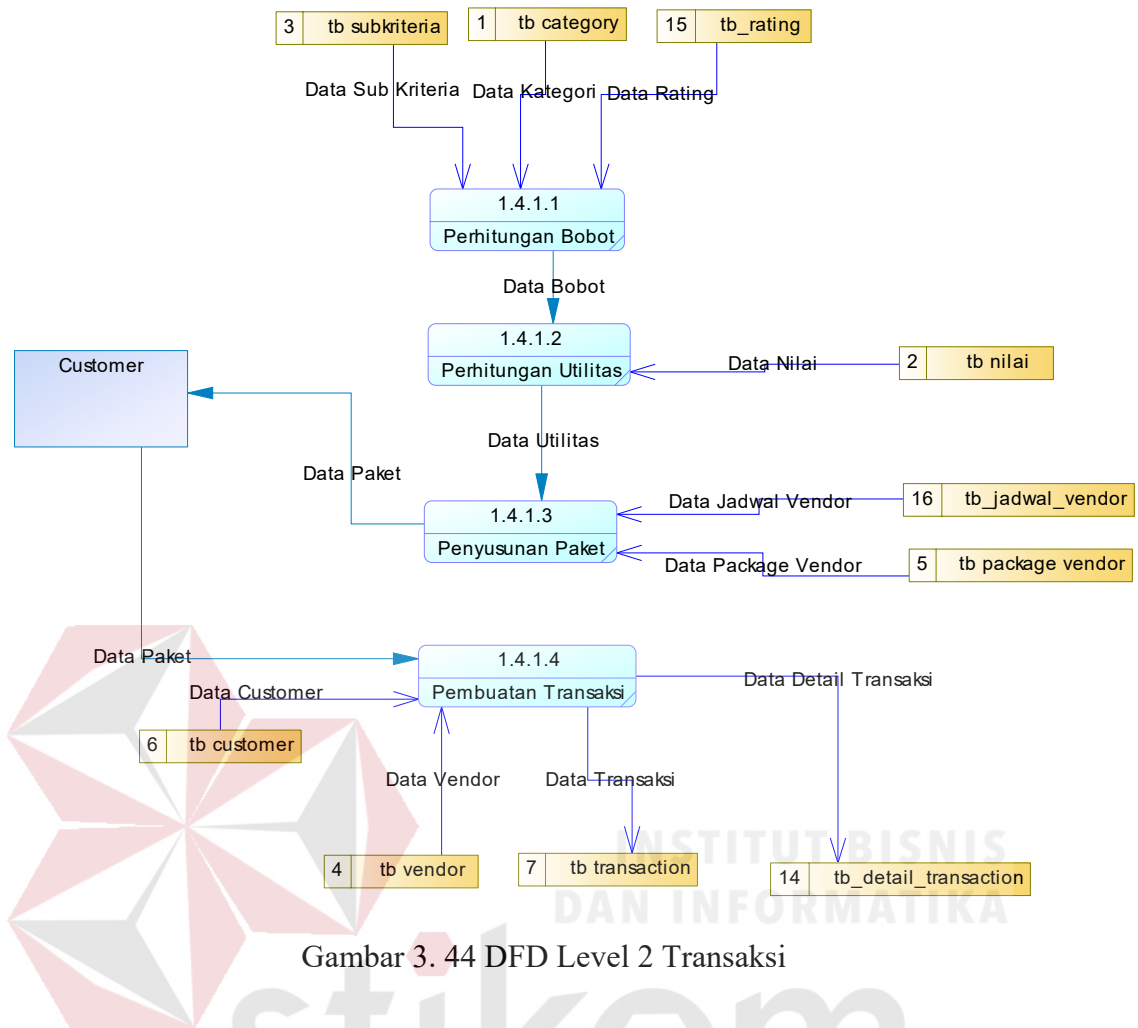
Pada level ini data yang mengalir untuk kegiatan pembuatan laporan yaitu meliputi pembuatan laporan member vendor, *customer*, laporan pendapatan, laporan pembuatan transaksi, pembuatan informasi peringkat dan jadwal. Berikut DFD level 1 pembuatan laporan dapat dilihat pada Gambar 3.43.



Gambar 3. 43 DFD Level 1 Pembuatan Laporan

iv. **Data Flow Diagram Level 2**

Data Flow Diagram Level 2 merupakan diagram yang menguraikan proses yang ada pada data flow diagram level 1. Pada DFD Level 2 digambarkan secara lebih detail proses-proses utama yakni transaksi yang dapat dilihat pada gambar dibawah.



Gambar 3. 44 DFD Level 2 Transaksi

### 3.2.3 Data Modelling

Pada data modelling menjelaskan model/konsep-konsep untuk menerangkan data, hubungan atau relasi antar data dan batasan-batasan data yang terintegrasi dalam database.

#### A. Entity Relationship Diagram (ERD)

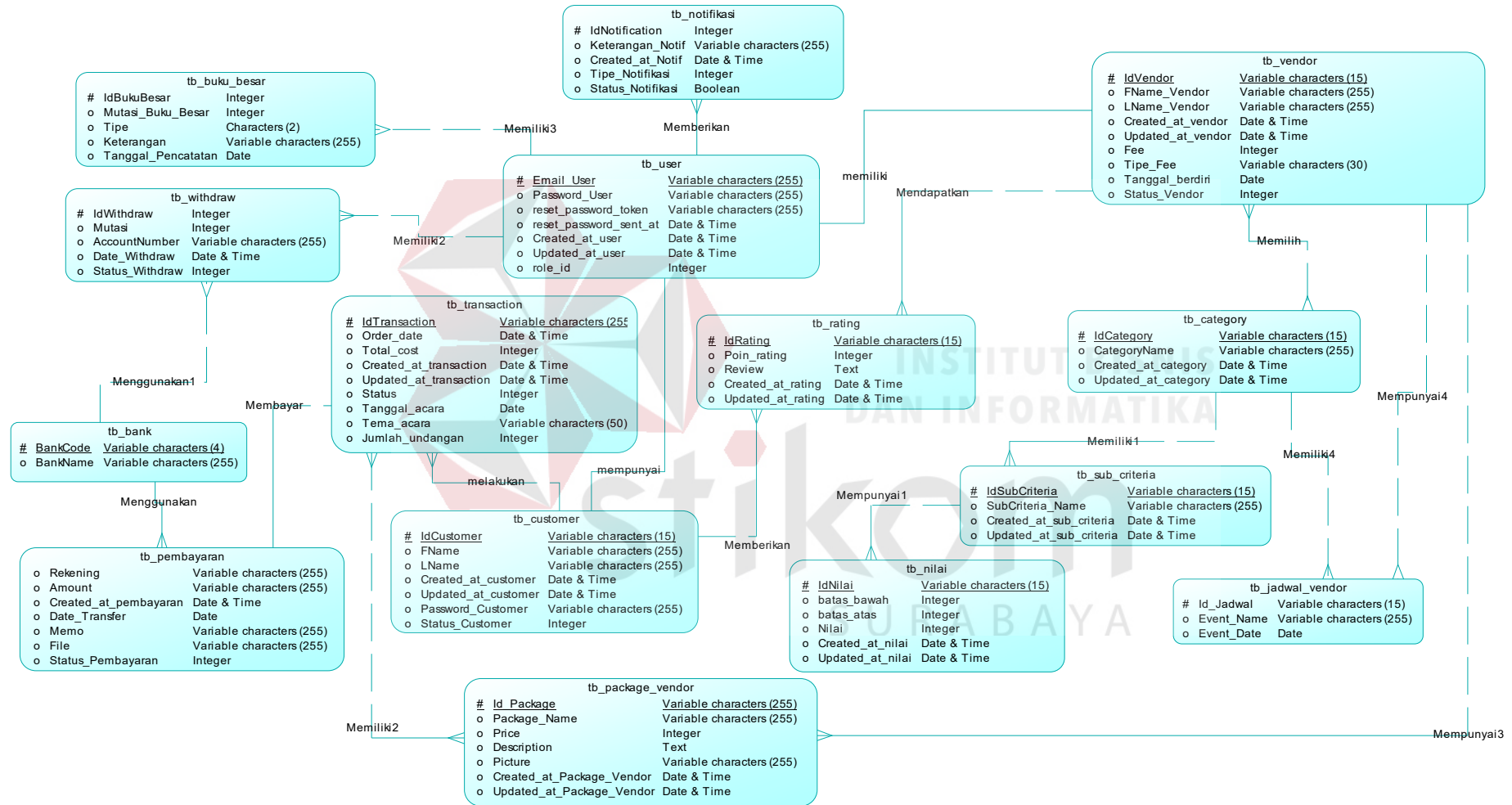
ERD menjelaskan hubungan antar data dalam *database* dari aplikasi penentuan vendor pernikahan untuk usaha *Wedding Organizer* menggunakan metode Smarter berbasis *web* ini menggunakan *System Development Life Cycle* (SDLC) *Waterfall Model*. ERD dapat dibagi menjadi 2 (dua), yaitu secara logika atau *Conceptual Data Model* (CDM) dan secara fisik *Physical Data Model* (PDM).



*i. Conceptual Data Model (CDM)*

CDM digunakan untuk menggambarkan secara detail struktur basis data dalam bentuk logik. CDM terdiri dari objek yang tidak diimplementasikan langsung kedalam basis data yang sesungguhnya. CDM pada aplikasi penentuan vendor pernikahan terdiri atas 8 (delapan) tabel master yaitu *tb\_user*, *tb\_vendor*, *tb\_category*, *tb\_customer*, *tb\_nilai*, *tb\_sub\_criteria*, *tb\_bank*, *tb\_package\_vendor* dan menghasilkan 7 (tujuh) tabel transaksi yaitu *tb\_transaction*, *tb\_pembayaran*, *tb\_withdraw*, *tb\_buku\_besar*, *tb\_notifikasi*, *tb\_jadwal\_vendor* dan *tb\_rating*. Berikut CDM aplikasi penentuan vendor pernikahan dapat dilihat pada gambar 3.45.

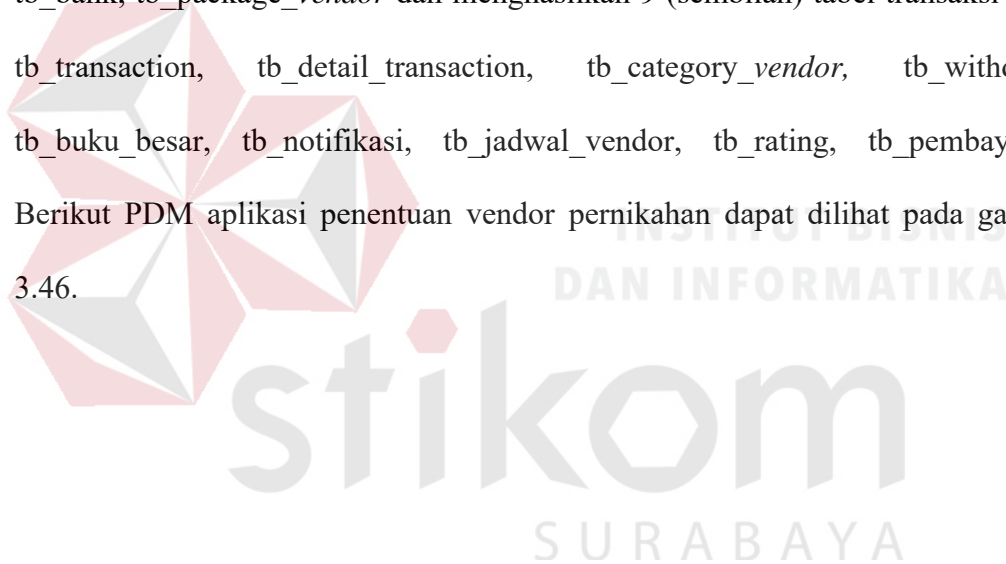


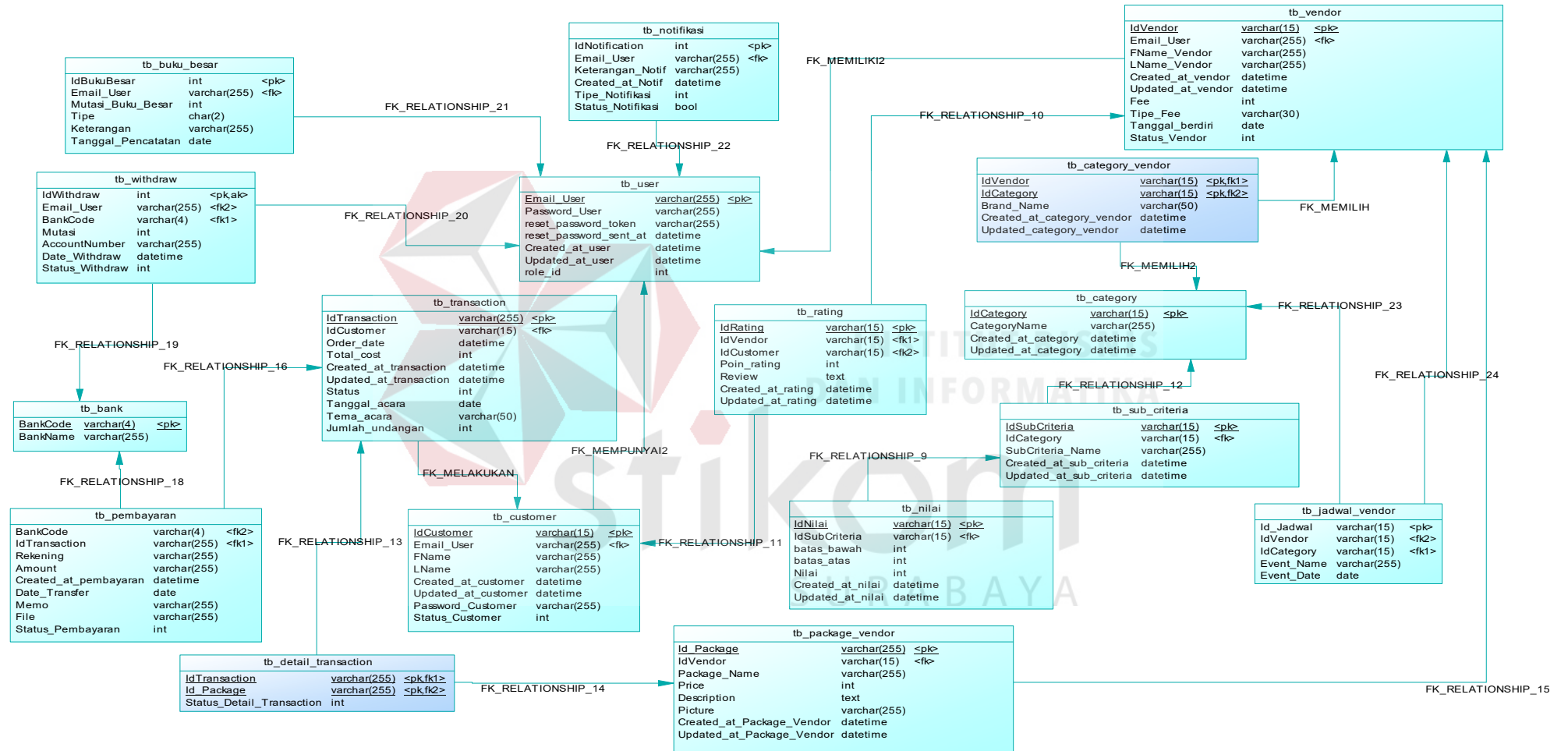


Gambar 3. 45 CDM Aplikasi Penentuan Vendor Pernikahan

## ii. *Physical Data Model (PDM)*

PDM merupakan hasil *generate* dari CDM yang menggambarkan secara detail basis data dalam bentuk fisik. Pada PDM juga sudah tergambar jelas relasi antar tabel yang ditunjukkan oleh adanya *primary key* dan *foreign key* pada setiap tabel. Hal tersebut menjelaskan bahwa PDM adalah rancangan struktur penyimpanan data yang benar pada basis data yang siap untuk digunakan. PDM pada aplikasi penentuan vendor pernikahan terdiri atas 8 (delapan) tabel master yaitu *tb\_user*, *tb\_vendor*, *tb\_category*, *tb\_customer*, *tb\_nilai*, *tb\_sub\_criteria*, *tb\_bank*, *tb\_package\_vendor* dan menghasilkan 9 (sembilan) tabel transaksi yaitu *tb\_transaction*, *tb\_detail\_transaction*, *tb\_category\_vendor*, *tb\_withdraw*, *tb\_buku\_besar*, *tb\_notifikasi*, *tb\_jadwal\_vendor*, *tb\_rating*, *tb\_pembayaran*. Berikut PDM aplikasi penentuan vendor pernikahan dapat dilihat pada gambar 3.46.





Gambar 3. 46 PDM Aplikasi Penentuan Vendor Pernikahan

### A. Struktur Basis Data

Struktur tabel pada aplikasi penentuan *vendor* pernikahan dirancang berdasarkan PDM. Berikut adalah struktur tabel yang dapat dilihat pada tabel dibawah:

#### 1 Tabel *User*

Nama Tabel : *tb\_user*

*Primary Key* : *Email\_User*

*Foreign Key* : -

Fungsi : Menyimpan data user setiap *customer*

Tabel 3. 35 Tabel *User*

No	Nama Kolom	Tipe Data	Size	Constraint
1	<i>Email_User</i>	<i>Varchar</i>	255	<i>Primary Key</i>
2	<i>Password_User</i>	<i>Varchar</i>	255	-
3	<i>reset_password_token</i>	<i>Varchar</i>	255	-
4	<i>reset_password_sent_at</i>	<i>Datetime</i>	-	-
5	<i>Created_at_user</i>	<i>Datetime</i>	-	-
6	<i>Updated_at_user</i>	<i>Datetime</i>	-	-
7	<i>role_id</i>	<i>Integer</i>	-	-

#### 2 Tabel *Vendor*

Nama Tabel : *tb\_vendor*

*Primary Key* : *IdVendor*

*Foreign Key* : *Email\_User*

Fungsi : Menyimpan data *vendor*

Tabel 3. 36 Tabel *Vendor*

No	Nama Kolom	Tipe Data	Size	Constraint
1	<i>IdVendor</i>	<i>Varchar</i>	15	<i>Primary Key</i>
2	<i>Email_User</i>	<i>Varchar</i>	255	<i>Foreign Key</i>

No	Nama Kolom	Tipe Data	Size	Constraint
3	FName_Vendor	Varchar	255	-
4	LName_Vendor	Varchar	255	-
5	Created_at_vendor	Datetime	-	-
6	Updated_at_vendor	Datetime	-	-
7	Fee	Integer	-	-
8	Tipe_Fee	Varchar	30	-
9	Tanggal_berdiri	Date	-	-
10	Status_Vendor	Integer	-	-

### 3 Tabel Kategori

Nama Tabel : tb\_category

Primary Key : IdCategory

Foreign Key : -

Fungsi : Menyimpan data kategori

Tabel 3. 37 Tabel *Category*

No	Nama Kolom	Tipe Data	Size	Constraint
1	IdCategory	Varchar	15	Primary Key
2	CategoryName	Varchar	255	-
3	Created_at_category	Datetime	-	-
4	Updated_at_category	Datetime	-	-

### 4 Tabel Kategori Vendor

Nama Tabel : tb\_category\_vendor

Primary Key : IdVendor, IdCategory

Foreign Key : IdVendor, IdCategory

Fungsi : Menyimpan data kategori vendor

Tabel 3. 38 Tabel *Category Vendor*

No	Nama Kolom	Tipe Data	Size	Constraint
1	IdVendor	Varchar	15	Primary Key, Foreign Key
2	IdCategory	Varchar	15	Primary Key, Foreign Key
3	Brand_Name	Varchar	50	-
4	Created_at_category_vendor	Datetime	-	-
5	Updated_at_category_vendor	Datetime	-	-

## 5 Tabel Sub Kriteria

Nama Tabel : tb\_sub\_criteria

Primary Key : IdSubCriteria

Foreign Key : IdCategory

Fungsi : Menyimpan data sub kriteria

Tabel 3. 39 Sub Kriteria

No	Nama Kolom	Tipe Data	Size	Constraint
1	IdSubCriteria	Varchar	15	Primary Key
2	IdCategory	Varchar	15	Foreign Key
3	SubCriteria_Name	Varchar	255	-
4	Created_at_sub_criteria	Datetime	-	-
5	Updated_at_sub_criteria	Datetime	-	-

## 6 Tabel Nilai

Nama Tabel : tb\_nilai

Primary Key : IdNilai

Foreign Key : IdSubCriteria

Fungsi : Menyimpan data nilai

Tabel 3. 40 Tabel Nilai

No	Nama Kolom	Tipe Data	Size	Constraint
1	IdNilai	Varchar	15	Primary Key
2	IdSubCriteria	Varchar	15	Foreign Key
3	batas_bawah	Integer	-	-
4	batas_atas	Integer	-	-
5	Nilai	Integer	-	-
6	Created_at_nilai	Datetime	-	-
7	Updated_at_nilai	Datetime	-	-

## 7 Tabel Paket Vendor

Nama Tabel : *tb\_package\_vendor*

Primary Key : *Id\_Package*

Foreign Key : *IdVendor*

Fungsi : Menyimpan data paket vendor

Tabel 3. 41 Tabel Paket Vendor

No	Nama Kolom	Tipe Data	Size	Constraint
1	Id_Package	Varchar	255	Primary Key
2	IdVendor	Varchar	15	Foreign Key
3	Package_Name	Varchar	255	-
4	Price	Integer	-	-
5	Description	Text	-	-
6	Picture	Varchar	255	-
7	Created_at_Package_Vendor	Datetime	-	-
8	Updated_at_Package_Vendor	Datetime	-	-

## 8 Tabel Customer

Nama Tabel : *tb\_customer*

Primary Key : *IdCustomer*

Foreign Key : *Email\_User*

Fungsi : Menyimpan data customer



Tabel 3. 42 Tabel *Customer*

No	Nama Kolom	Tipe Data	Size	Constraint
1	IdCustomer	Varchar	15	Primary Key
2	Email_User	Varchar	255	Foreign Key
3	FName	Varchar	255	-
4	LName	Varchar	255	-
5	Created_at_vendor	Datetime	-	-
6	Updated_at_vendor	Datetime	-	-
7	Password_Customer	Varchar	255	-
8	Status_Customer	Integer	-	-

## 9 Tabel Transaksi

Nama Tabel : tb\_transaction

Primary Key : IdTransaction

Foreign Key : IdCustomer

Fungsi : Menyimpan data transaksi

Tabel 3. 43 Tabel Transaksi

No	Nama Kolom	Tipe Data	Size	Constraint
1	IdTransaction	Varchar	255	Primary Key
2	IdCustomer	Varchar	15	Foreign Key
3	Total_cost	Integer	-	-
4	Created_at_vendor	Datetime	-	-
5	Updated_at_vendor	Datetime	-	-
6	Status	Integer	-	-
7	Tanggal_acara	Date	-	-
8	Tema_acara	Varchar	50	-
9	Jumlah_undangan	Integer	-	-

## 10 Tabel Detail Transaksi

Nama Tabel : tb\_detail\_transaction

Primary Key : IdTransaction

Foreign Key : Id\_Package

Fungsi : Menyimpan data detail transaksi

Tabel 3. 44 Tabel Detail Transaksi

No	Nama Kolom	Tipe Data	Size	Constraint
1	IdTransaction	Varchar	255	Primary Key, Foreign Key
2	Id_Package	Varchar	255	Primary Key, Foreign Key
3	Status_Detail_Transaction	Integer	-	-

### 11 Tabel Rating

Nama Tabel : tb\_rating

Primary Key : IdRating

Foreign Key : IdVendor, IdCustomer

Fungsi : Menyimpan data rating dari *customer*

Tabel 3. 45 Tabel Rating

No	Nama Kolom	Tipe Data	Size	Constraint
1	IdRating	Varchar	15	Primary Key
2	IdVendor	Varchar	15	Foreign Key
3	IdCustomer	Varchar	15	Foreign Key
4	Poin_rating	Integer	-	-
5	Review	Text	-	-
6	Created_at_rating	Datetime	-	-
7	Updated_at_rating	Datetime	-	-

### 12 Tabel Bank

Nama Tabel : tb\_bank

Primary Key : BankCode

Foreign Key : -

Fungsi : Menyimpan data bank

Tabel 3. 46 Tabel *Bank*

No	Nama Kolom	Tipe Data	Size	Constraint
1	BankCode	Varchar	4	Primary Key
2	BankName	Varchar	255	-

## 13 Tabel Pembayaran

Nama Tabel : tb\_pembayaran

Primary Key : BankCode, IdTransaction

Foreign Key : BankCode, IdTransaction

Fungsi : Menyimpan data pembayaran

Tabel 3. 47 Tabel Pembayaran

No	Nama Kolom	Tipe Data	Size	Constraint
1	BankCode	Varchar	4	Primary Key, Foreign Key
2	IdTransaction	Varchar	255	Primary Key, Foreign Key
3	Rekening	Varchar	255	-
4	Amount	Varchar	255	-
5	Created_at_pembayaran	Datetime	-	-
6	Date_Transfer	Date	-	-
7	Memo	Varchar	255	-
8	File	Varchar	255	-
9	Status_Pembayaran	Integer	-	-

14 Tabel *Withdraw*

Nama Tabel : tb\_withdraw

Primary Key : IdWithdraw

Foreign Key : Email\_User, BankCode,

Fungsi : Menyimpan data *withdraw*

Tabel 3. 48 Tabel *Withdraw*

No	Nama Kolom	Tipe Data	Size	Constraint
1	IdWithdraw	Integer	-	Primary Key
2	Email_User	Varchar	255	Foreign Key
3	BankCode	Varchar	4	Foreign Key
4	Mutasi	Integer	-	-
5	AccountNumber	Varchar	255	-
6	Date_Withdraw	Datetime	-	-
7	Status_Withdraw	Integer	-	-

## 15 Tabel Buku Besar

Nama Tabel : tb\_buku\_besar

Primary Key : IdBukuBesar

Foreign Key : Email\_User

Fungsi : Menyimpan data keluar masuknya saldo

Tabel 3. 49 Tabel Buku Besar

No	Nama Kolom	Tipe Data	Size	Constraint
1	IdBukuBesar	Integer	-	Primary Key
2	Email_User	Varchar	255	Foreign Key
3	Mutasi_Buku_Besar	Integer	-	-
4	Tipe	Char	2	-
5	Keterangan	Varchar	255	-
6	Tanggal_Pencatatan	Date	-	-

## 16 Tabel Jadwal Vendor

Nama Tabel : tb\_jadwal\_vendor

Primary Key : Id\_Jadwal

Foreign Key : IdVendor, IdCategory

Fungsi : Menyimpan data jadwal vendor

Tabel 3. 50 Tabel Jadwal Vendor

No	Nama Kolom	Tipe Data	Size	Constraint
1	IdJadwal	Varchar	15	Primary Key
2	IdVendor	Varchar	15	Foreign Key
3	IdCategory	Varchar	15	Foreign Key
4	Event_Name	Varchar	225	-
5	Event_Date	Date	-	-

## 17 Tabel Notifikasi

Nama Tabel : tb\_notifikasi

Primary Key : IdNotification

Foreign Key : Email\_User

Fungsi : Menyimpan data notifikasi

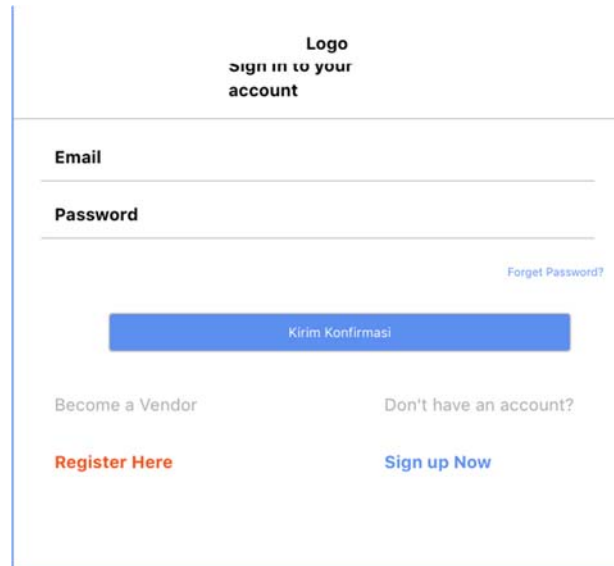
Tabel 3. 51 Tabel Notifikasi

No	Nama Kolom	Tipe Data	Size	Constraint
1	IdNotification	Integer	-	Primary Key
2	Email_User	Varchar	255	Foreign Key
3	Keterangan_Notif	Varchar	255	-
4	Created_at_Notif	Datetime	-	-
5	Tipe_Notifikasi	Integer	-	-
6	Status_Notifikasi	Boolean	-	-

### 3.2.4 Perancangan Antar Muka (Interface)

#### A. Desain Halaman Login Pengguna

Berikut adalah desain halaman *login* pengguna yang digunakan sebagai acuan dalam implementasi halaman *login* pengguna yang dapat dilihat pada Gambar 3.47.



**Logo**  
Sign in to your  
account

---

**Email**

---

**Password**

---

[Forget Password?](#)

**Kirim Konfirmasi**

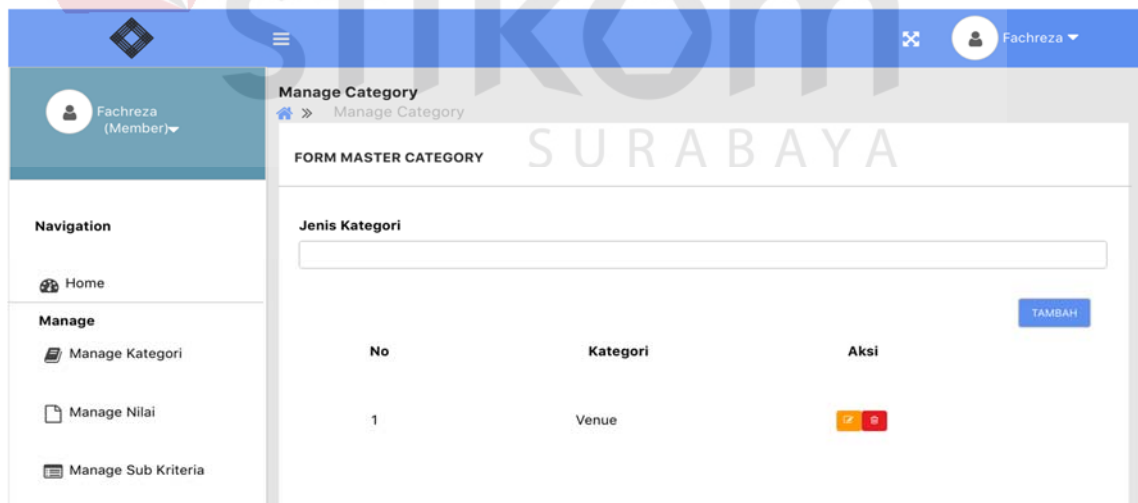
Become a Vendor      Don't have an account?

**Register Here**      **Sign up Now**

Gambar 3. 47 Desain Halaman Login Pengguna

## B. Desain Halaman Pengelolaan Master Kategori

Berikut adalah desain halaman master kategori yang digunakan sebagai acuan dalam implementasi halaman pengelolaan master kategori yang dapat dilihat pada Gambar 3.48.





**Manage Category**

Manage Category

**FORM MASTER CATEGORY**

Jenis Kategori

**TAMBAH**

No	Kategori	Aksi
1	Venue	 

Gambar 3. 48 Desain Halaman Pengelolaan Master Kategori

### C. Desain Halaman Master Nilai

Berikut adalah desain halaman master nilai yang digunakan sebagai acuan dalam implementasi halaman pengelolaan master nilai yang dapat dilihat pada Gambar 3.49.

Gambar 3. 49 Desain Halaman Pengelolaan Master Nilai

### D. Desain Pengelolaan Master Sub Kriteria



Berikut adalah desain halaman master sub kriteria yang digunakan sebagai acuan dalam implementasi halaman pengelolaan master sub kriteria yang dapat dilihat pada Gambar 3.50.

**Manage Sub Kriteria**  
» Manage Sub Kriteria

**FORM MASTER Sub Kriteria**

Nilai Batas Bawah  Nilai Batas Atas

TAMBAH

No	Sub Kriteria	Kriteria	Aksi
1	Pengalaman	Venue	 

Gambar 3. 50 Desain Halaman Pengelolaan Master Sub Kriteria

### E. Desain Pengelolaan *Bank*

Berikut adalah desain halaman pengelolaan master *bank* yang digunakan sebagai acuan dalam implementasi halaman pengelolaan master *bank* yang dapat dilihat pada Gambar 3.51.



**Manage Bank**  
» Manage Bank

**FORM MASTER Bank**

Kode Bank

Nama Bank

TAMBAH

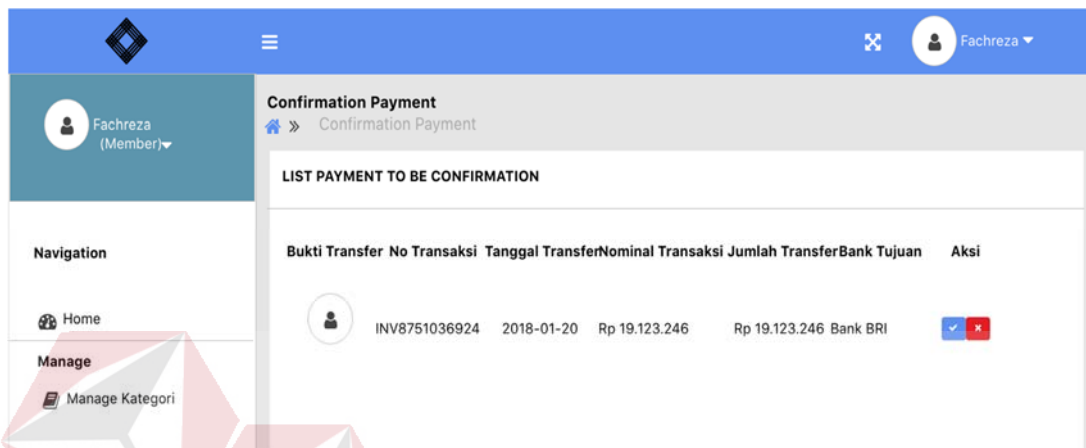
No	Kode bank	Nama Bank	Aksi
1	002	Bank BRI	 

Gambar 3. 51 Desain Halaman Pengelolaan Master Bank



## F. Desain Konfirmasi Pembayaran Pada Admin

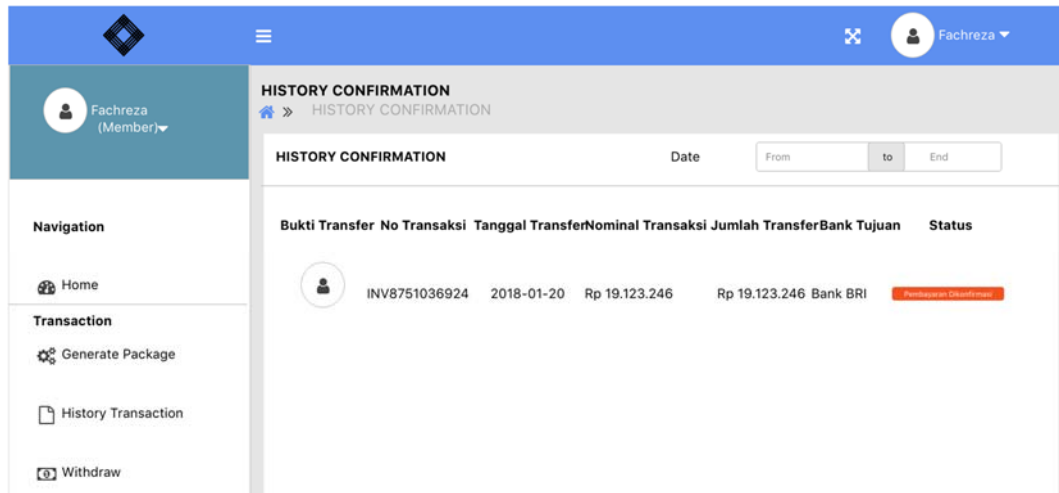
Berikut adalah desain halaman konfirmasi pembayaran pada admin yang digunakan sebagai acuan dalam implementasi halaman konfirmasi pembayaran pada admin yang dapat dilihat pada Gambar 3.52.



Gambar 3. 52 Desain Halaman Konfirmasi Pembayaran Pada Admin

## G. Desain Histori Konfirmasi Pembayaran Pada Admin

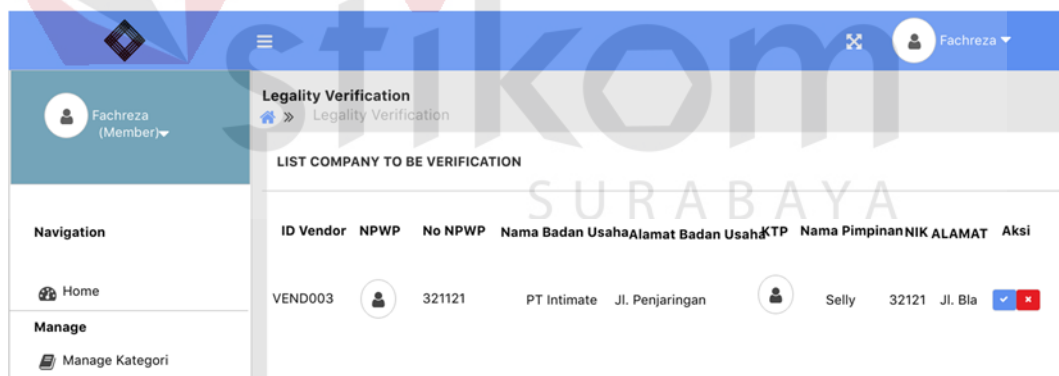
Berikut adalah desain halaman histori konfirmasi pembayaran yang digunakan sebagai acuan dalam implementasi halaman histori konfirmasi pembayaran pada admin yang dapat dilihat pada Gambar 3.53.



Gambar 3. 53 Desain Halaman Histori Konfirmasi Pembayaran

## H. Desain *Legality Verification*

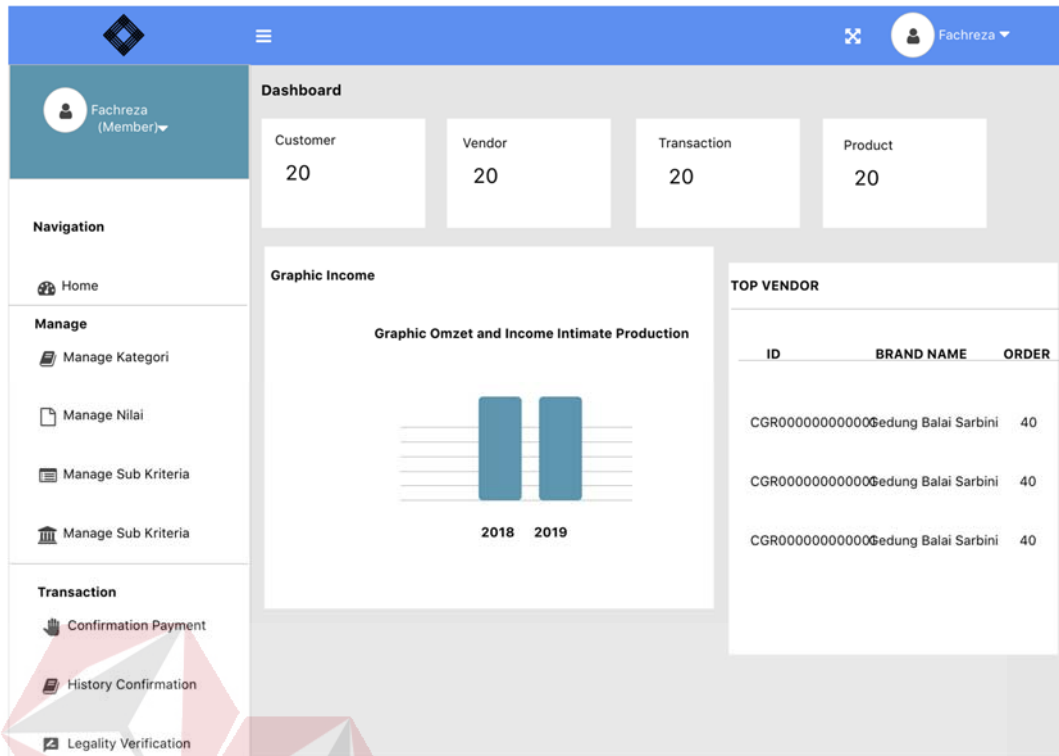
Berikut adalah desain halaman *legality verification* yang digunakan sebagai acuan dalam implementasi halaman *legality verification* yang dapat dilihat pada Gambar 3.54.



Gambar 3. 54 Desain Halaman *Legality Verification*

## I. Desain *Dashboard*

Berikut adalah desain halaman *dashboard* yang digunakan sebagai acuan dalam implementasi halaman *dashboard* yang dapat dilihat pada Gambar 3.55.



Gambar 3. 55 Desain Halaman *Dashboard* Admin

## J. Desain Registrasi Vendor

Berikut adalah desain halaman registrasi vendor yang digunakan sebagai acuan dalam implementasi halaman pengelolaan registrasi vendor yang dapat dilihat pada Gambar 3.56.

**Logo**

**Create an account for Your Business**

---

**Email Address**

**Password**

**Confirm Password**

[Sign Up](#)

Already have an account? [Sign In Here](#)

Gambar 3. 56 Desain Halaman Registrasi Vendor

### K. Desain *Legality Vendor*

Berikut adalah desain halaman *legality vendor* yang digunakan sebagai acuan dalam implementasi halaman *legality vendor* yang dapat dilihat pada Gambar 3.57.

The image shows a web form for vendor legality registration. The form is titled "Lengkapi Form Berikut" and is divided into two main sections: "FORM LEGALITAS PERUSAHAAN" and "KONFIRMASI PEMBAYARAN".

**FORM LEGALITAS PERUSAHAAN**

- Nama Badan Usaha
- Alamat Badan Usaha
- Tanggal Pendirian Badan Usaha
- NPWP
- Foto NPWP

**KONFIRMASI PEMBAYARAN**

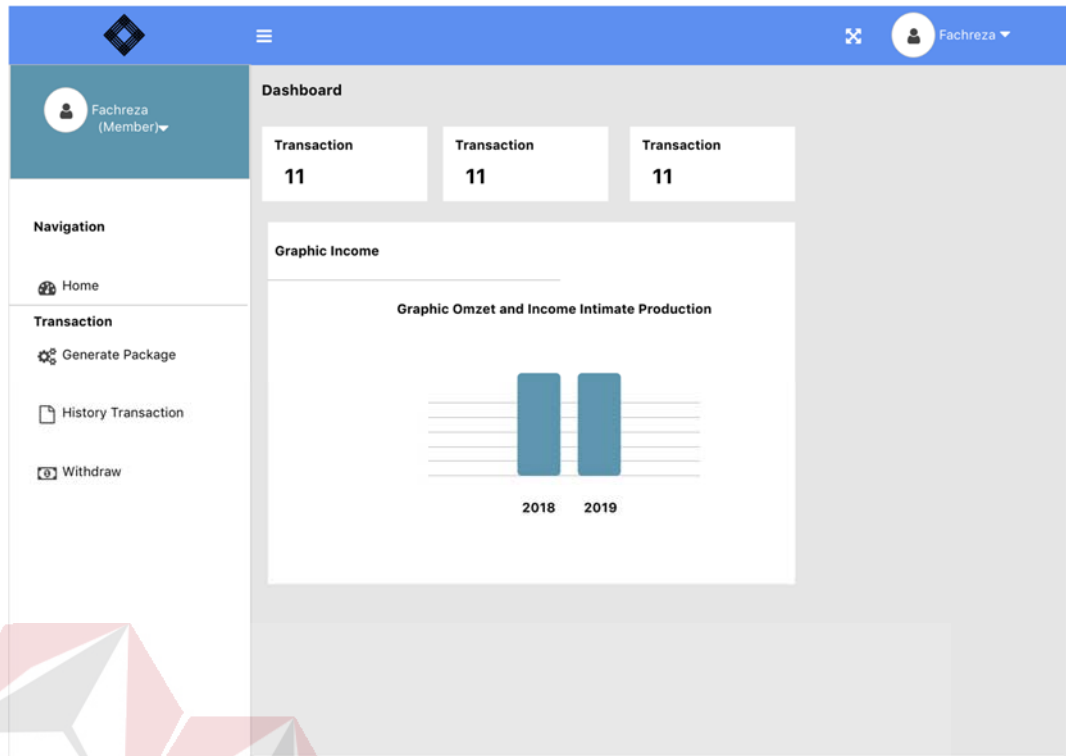
- Nama Depan Pimpinan
- Nama Belakang Pimpinan
- NIK
- Alamat
- Foto KTP

A "Submit" button is located at the bottom of the form. The page has a blue header with a logo, a menu icon, and a user profile icon labeled "Fachreza".

Gambar 3. 57 Desain Halaman *Legality Vendor*

### L. Desain *Dashboard Vendor*

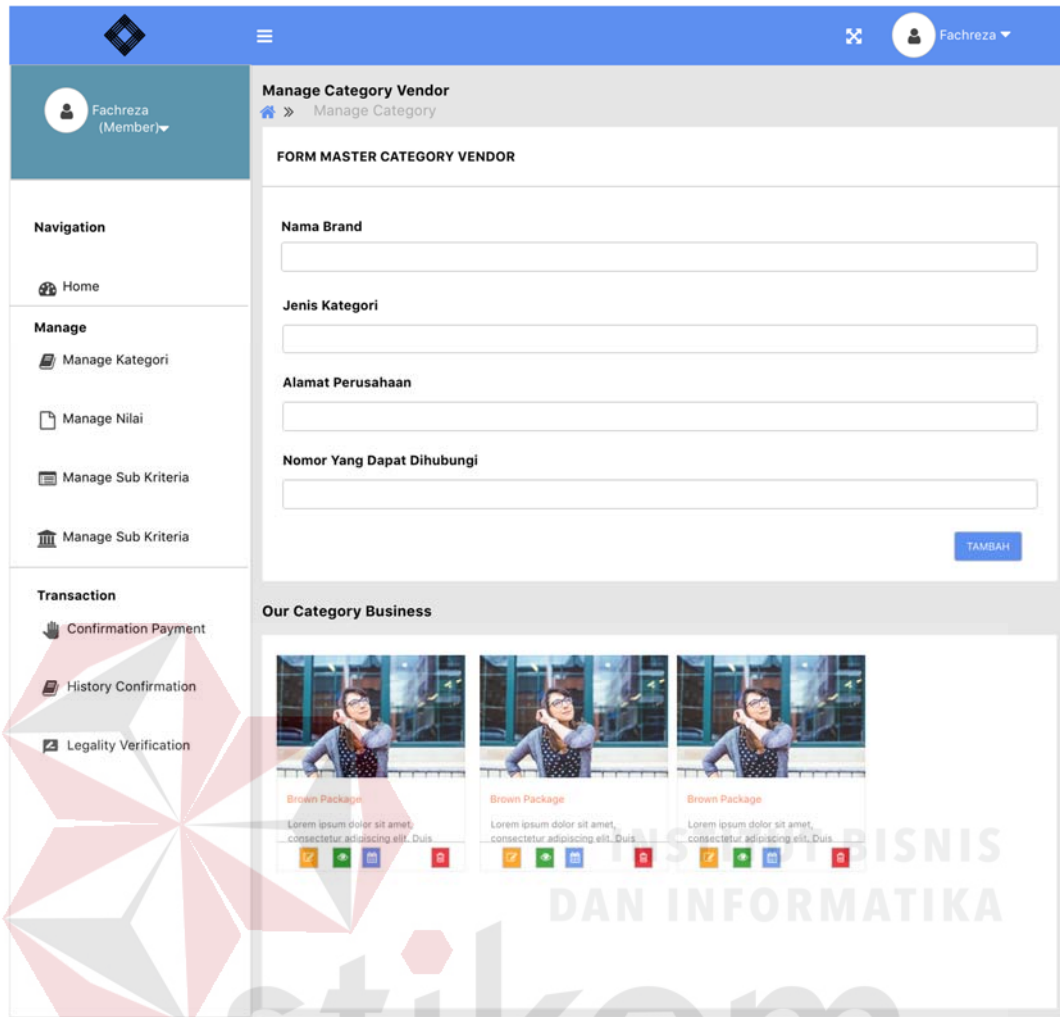
Berikut adalah desain *dashboard* pada vendor yang digunakan sebagai acuan dalam implementasi halaman *dashboard* yang dapat dilihat pada Gambar 3.58.



Gambar 3. 58 Desain Halaman *Dashboard* Vendor

### M. Desain Pengelolaan Kategori Vendor

Berikut adalah desain halaman kategori vendor yang digunakan sebagai acuan dalam implementasi halaman pengelolaan kategori vendor yang dapat dilihat pada Gambar 3.59.



Gambar 3. 59 Desain Halaman Pengelolaan Kategori Vendor

## N. Desain Pengelolaan Data Paket Vendor

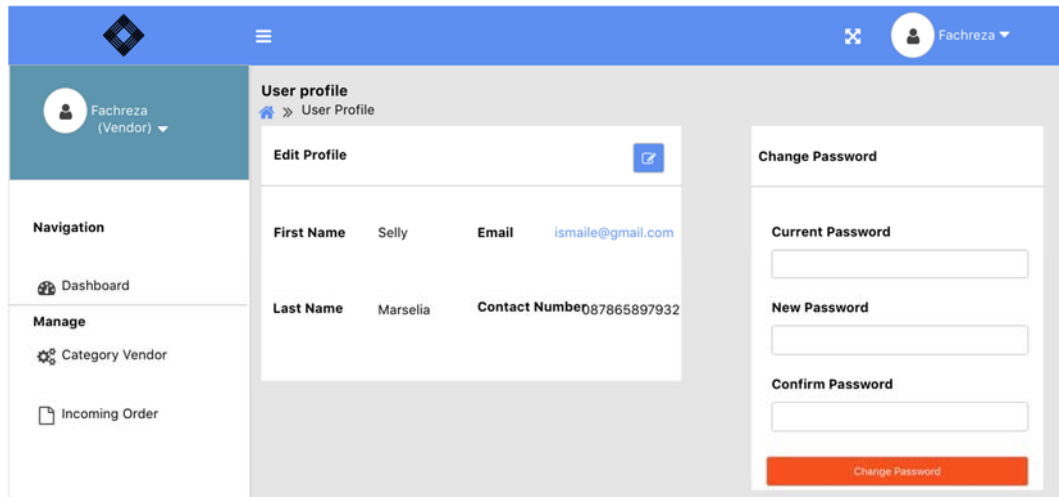
Berikut adalah desain halaman pengelolaan data paket vendor yang digunakan sebagai acuan dalam implementasi halaman pengelolaan data paket vendor yang dapat dilihat pada Gambar 3.60.

The image shows a web application interface for managing vendor data packages. It features a blue header with a logo and user profile 'Fachreza (Member)'. A left sidebar contains navigation options: Home, Manage (Manage Kategori, Manage Nilai, Manage Sub Kriteria), and Transaction (Confirmation Payment, History Confirmation, Legality Verification). The main area is titled 'Manage Data Package' and includes a breadcrumb 'Manage Category > Venue'. The 'FORM MASTER PACKAGE' section has input fields for 'Nama Paket', 'Harga Paket', and 'Deskripsi Paket', and an 'Unggah Gambar' section with a 'Choose File...' button and a 'Browse' button. A 'TAMBAH' button is at the bottom right. Below the form is a section 'Our Package' showing a preview of a package with a photo of a woman and placeholder text.

Gambar 3. 60 Desain Halaman Pengelolaan Data Paket Vendor

### O. Desain Pengelolaan *Profile Vendor*

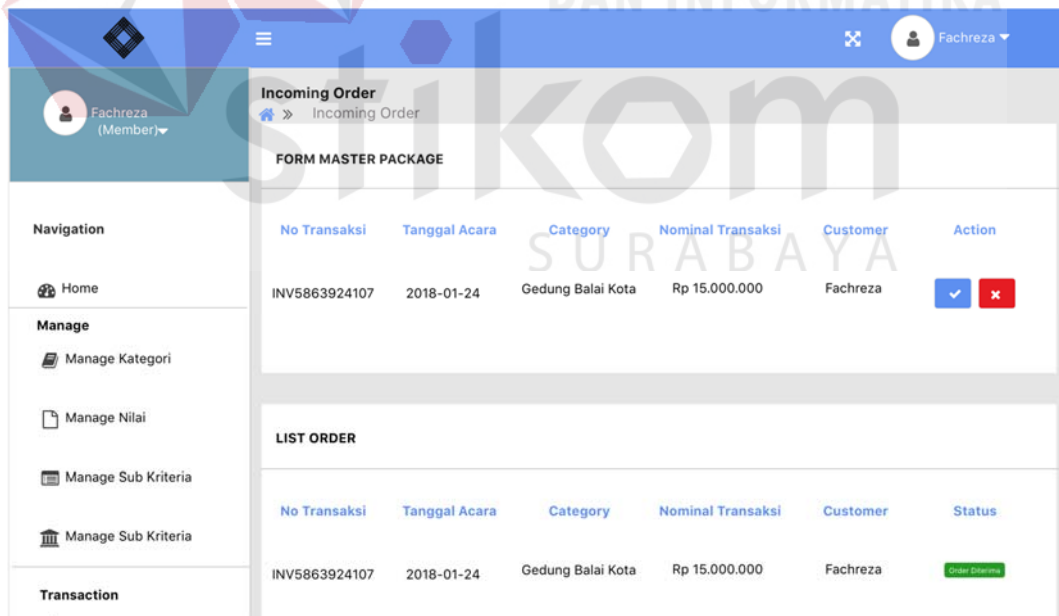
Berikut adalah desain halaman pengelolaan *profile vendor* yang digunakan sebagai acuan dalam implementasi halaman pengelolaan *profile vendor* yang dapat dilihat pada Gambar 3.61.



Gambar 3. 61 Desain Halaman Pengelolaan Profile Vendor

## P. Desain Konfirmasi *Order*

Berikut adalah desain halaman konfirmasi *order* pada vendor yang digunakan sebagai acuan untuk implementasi halaman konfirmasi *order* yang dapat dilihat pada Gambar 3.62.



Gambar 3. 62 Desain Halaman Konfirmasi *Order*



### Q. Desain Jadwal Vendor

Berikut adalah desain halaman jadwal vendor yang digunakan sebagai acuan untuk implementasi halaman jadwal vendor yang dapat dilihat pada Gambar 3.63.

**Jadwal Acara Nama Brand**

Tambahkan Jadwal Acara Yang Anda Memiliki

**Nama Acara**

**Tanggal Acara**

**Tambah Acara**

No	Nama Acara	Tanggal Acara	Aksi
1	INV8751036924 - Traditional	2018-01-20	<input type="checkbox"/> <input type="checkbox"/>

Gambar 3. 63 Desain Halaman Jadwal Vendor

### R. Desain Registrasi *Customer*

Berikut adalah desain halaman registrasi *customer* yang digunakan sebagai acuan dalam implementasi halaman registrasi *customer* yang dapat dilihat pada Gambar 3.64.

**Logo**

**Create an account**

**First Name** **Last Name**

**Email Address**

**Password**

**Confirm Password**

**Sign Up**

Already have an account? [Sign In Here](#)

Gambar 3. 64 Desain Halaman Registrasi *Customer*

### S. Desain Pengelolaan *Profile Customer*

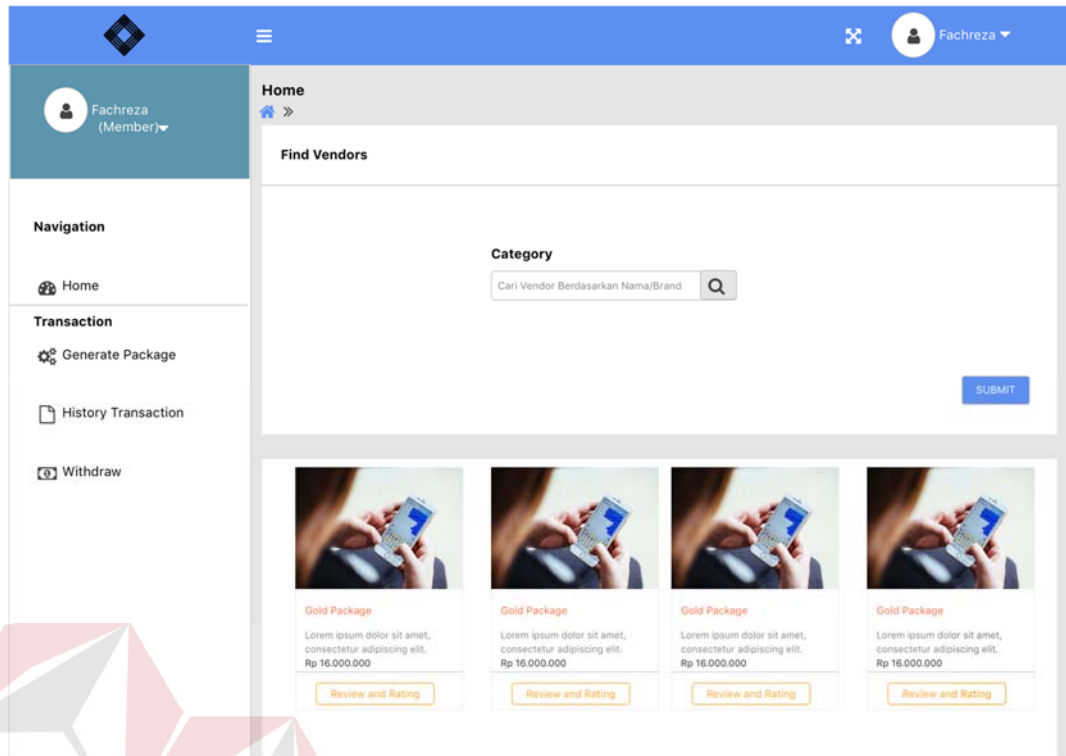
Berikut adalah desain halaman pengelolaan *profile customer* yang digunakan sebagai acuan dalam implementasi halaman pengelolaan *profile customer* yang dapat dilihat pada Gambar 3.65.

The image shows a user profile management interface. On the left, there is a sidebar with a user profile card for 'Fachreza (Member)', a 'Navigation' menu with 'Home', and a 'Transaction' menu with 'Generate Package', 'History Transaction', and 'Withdraw'. The main content area is divided into two sections: 'Edit Profile' and 'Change Password'. The 'Edit Profile' section has fields for 'First Name', 'Last Name', 'Email Address', and 'Contact No', with an 'Update Profile' button. The 'Change Password' section has fields for 'Current Password', 'New Password', and 'Confirm Password', with a 'Change Password' button. The top navigation bar is blue and contains a logo, a menu icon, a window icon, and the user's name 'Fachreza'.

Gambar 3. 65 Desain Halaman Pengelolaan *Profile Customer*

### T. Desain Pencarian Vendor

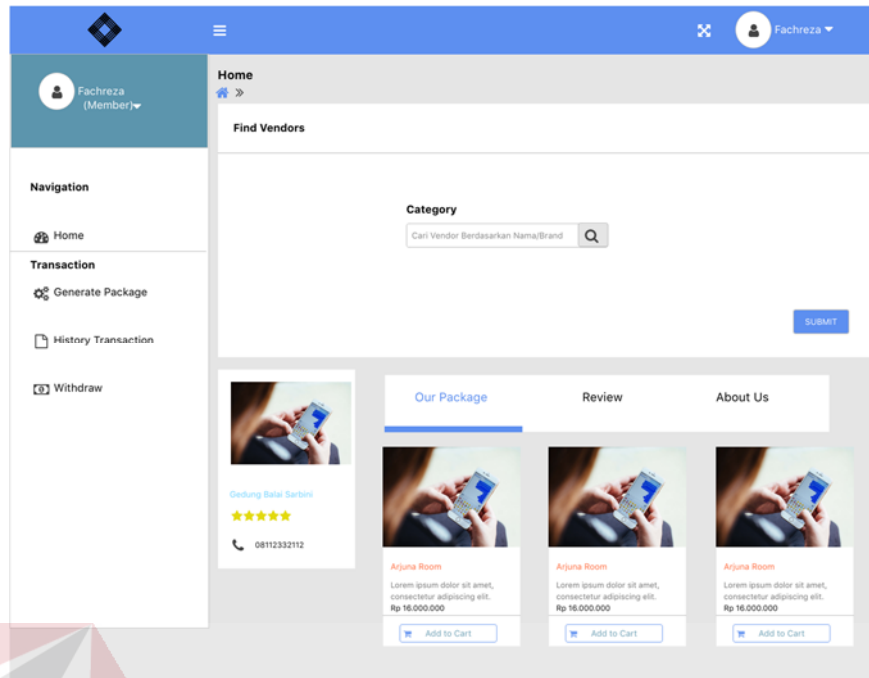
Berikut desain halaman pencarian vendor yang digunakan sebagai acuan untuk implementasi halaman pencarian vendor yang dapat dilihat pada Gambar 3.66.



Gambar 3. 66 Desain Halaman Pencarian Vendor

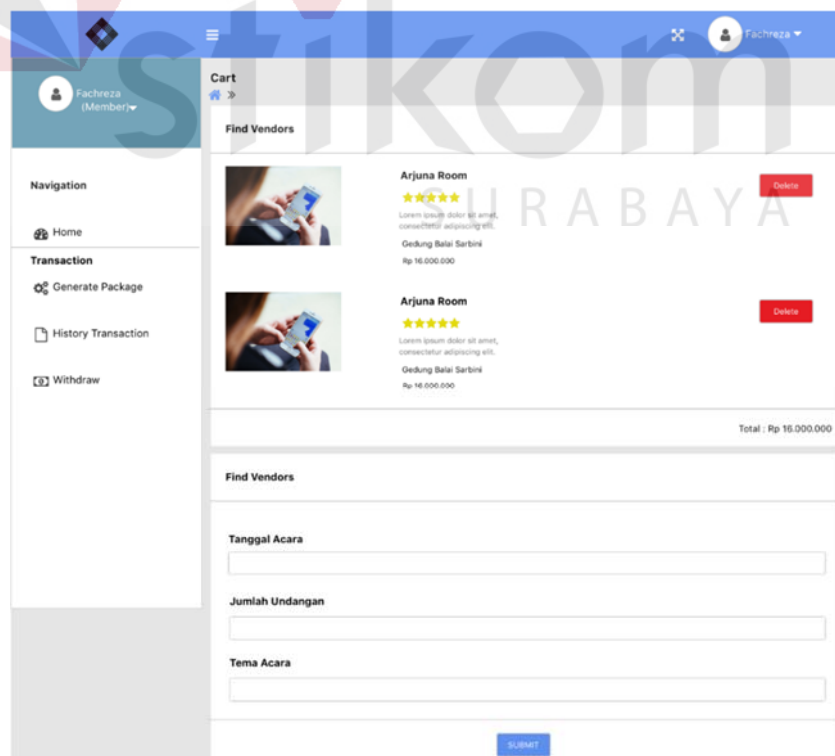
#### U. Desain *Customize Order*

Berikut adalah desain halaman *customize order* yang digunakan sebagai acuan untuk implementasi halaman *customize order* yang dapat dilihat pada Gambar 3.67.



Gambar 3. 67 Desain Halaman *Customize Order*

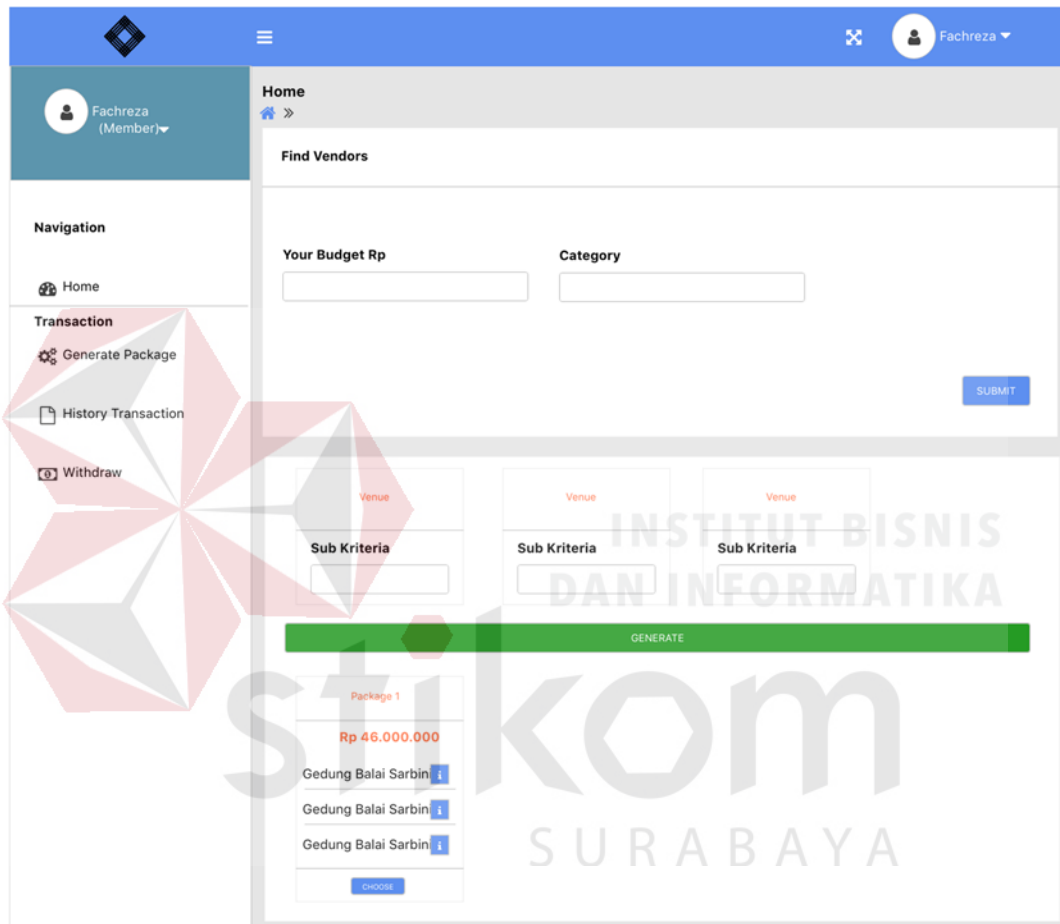
Sedangkan desain halaman *cart*/keranjang belanja setelah dilakukannya *order* secara *customize*, dapat dilihat pada Gambar 3.68.



Gambar 3. 68 Desain Halaman *Cart*

## V. Desain *Generate* Paket dan Pemesanan Paket

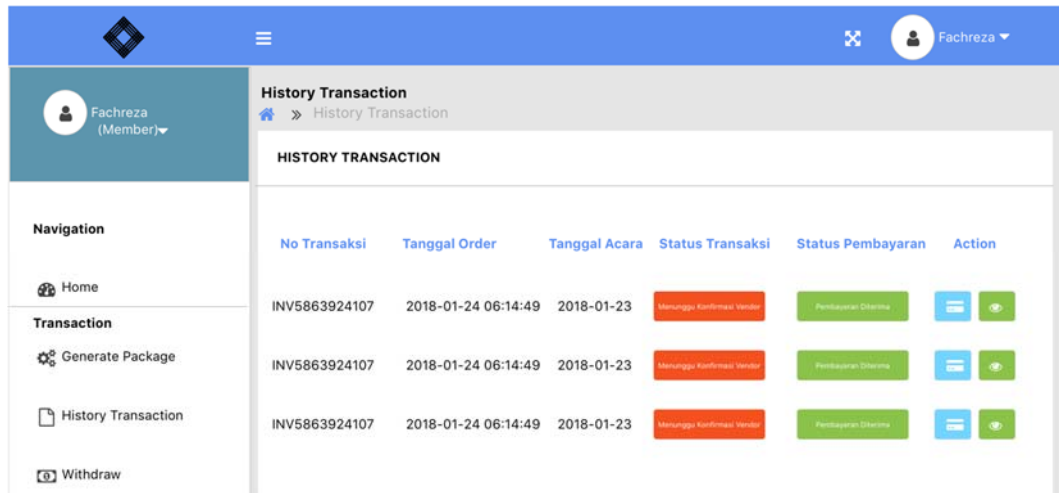
Berikut adalah desain halaman *generate* paket dan pemesanan paket yang digunakan sebagai acuan dalam implementasi halaman *generate* paket dan pemesanan paket yang dapat dilihat pada Gambar 3.69.



Gambar 3. 69 Desain Halaman *Generate* Paket dan Pemesanan Paket

## W. Desain Histori Transaksi

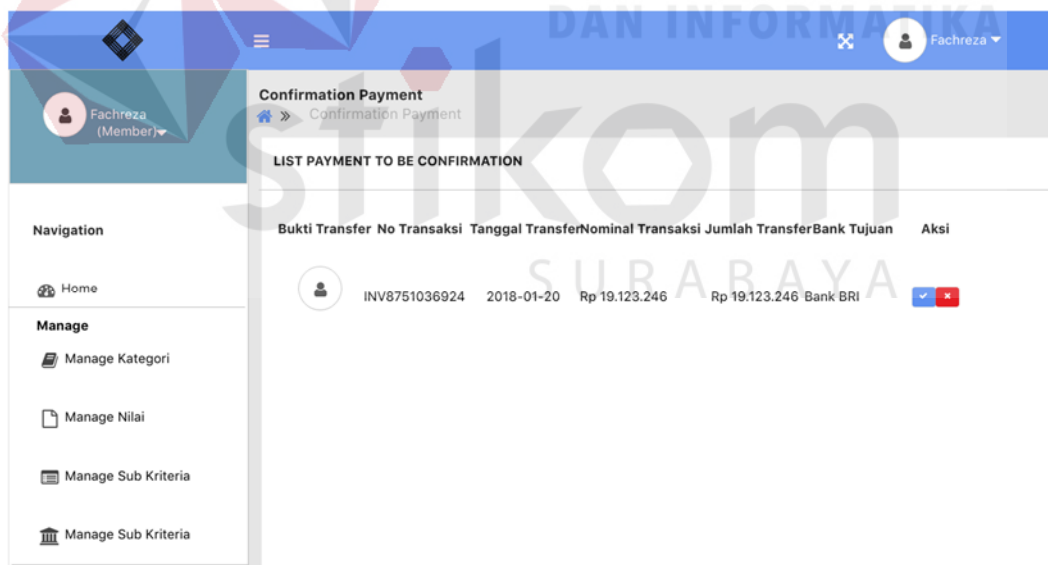
Berikut adalah desain halaman histori transaksi yang digunakan sebagai acuan dalam implementasi halaman histori transaksi yang dapat dilihat pada Gambar 3.70.



Gambar 3. 70 Desain Halaman Histori Transaksi

## X. Desain Konfirmasi Pembayaran Pada *Customer*

Berikut adalah desain halaman konfirmasi pembayaran pada *customer* yang digunakan sebagai acuan untuk implementasi halaman konfirmasi pembayaran yang dapat dilihat pada Gambar 3.71.



Gambar 3. 71 Desain Halaman Konfirmasi Pembayaran Pada *Customer*

### Y. Desain *Forgot Password*

Berikut adalah desain halaman *forgot password* yang digunakan sebagai acuan dalam implementasi halaman *forgot password* yang dapat dilihat pada Gambar 3.72.

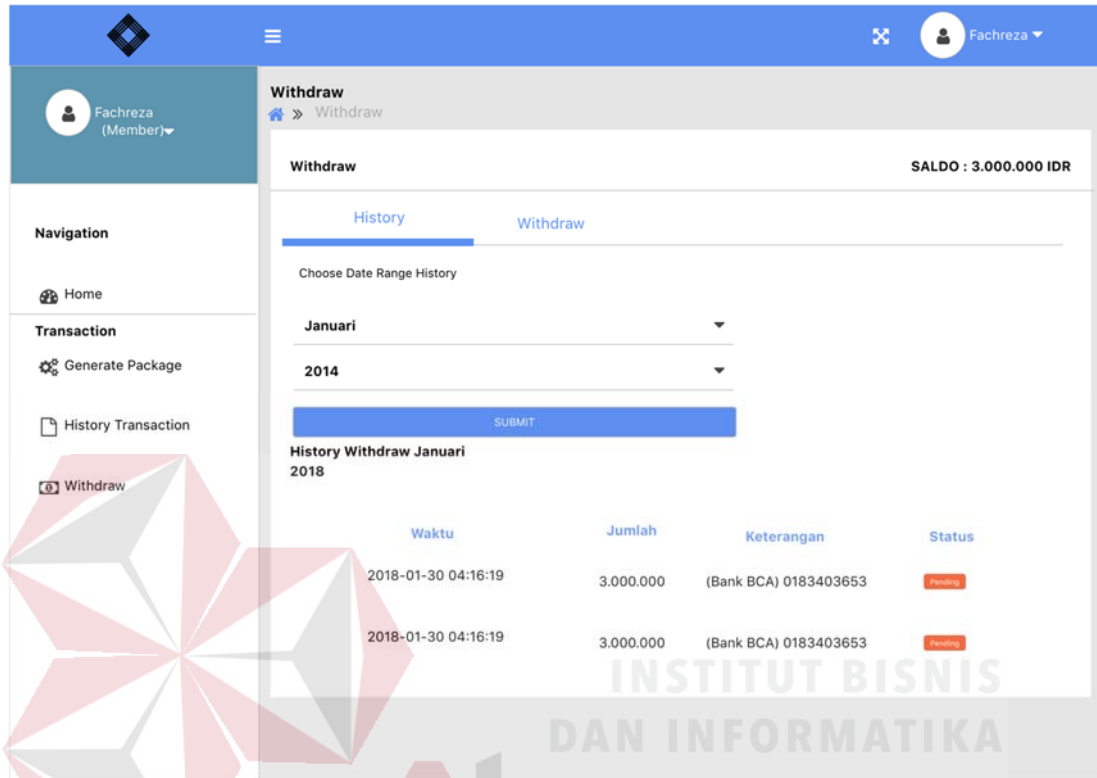
Gambar 3. 72 Desain Halaman *Forgot Password*

### Z. Desain *Withdraw*

Berikut adalah desain halaman *withdraw* yang digunakan sebagai acuan implementasi halaman *withdraw* yang dapat dilihat pada Gambar 3.73.

Gambar 3. 73 Desain Halaman *Withdraw*

Sedangkan desain halaman *history withdraw* dapat dilihat pada Gambar 3.74.



Gambar 3. 74 Desain Halaman *History Withdraw*

### 3.2.5 Perancangan Uji Coba Sistem dengan *Black Box*

#### A. Desain Uji Coba *Form Login*

Desain uji coba *login* bertujuan untuk menguji apakah fungsi melakukan *login* dapat berjalan sesuai yang diharapkan. Desain uji coba *login* dapat dilihat pada Tabel 3.52.

Tabel 3. 52 Desain Uji Coba *Form Login*

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
1.	Melakukan <i>Login</i>	<i>Email</i> dan <i>Password</i>	<i>Login</i> berhasil dan masuk ke halaman sesuai <i>role</i>	



Test Case ID	Tujuan	Input	Output yang diharapkan	Status
2.	Melakukan <i>Login</i> dengan <i>format</i> salah	<i>Email</i> dan <i>Password</i>	Muncul pesan " <i>Email</i> atau <i>Password</i> Salah"	

## B. Pengelolaan Pada *Role Admin*

### a. Desain Uji Coba Pengelolaan Master Kategori

Desain uji coba pengelolaan master kategori bertujuan untuk menguji apakah fungsi *Create, Read, Update, Delete* (CRUD) pada master kategori dapat berjalan sesuai yang diharapkan. Desain uji coba pengelolaan master kategori dapat dilihat pada Tabel 3.53.

Tabel 3. 53 Desain Uji Coba *Form* Master Kategori

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
3.	Menampilkan data kategori	Memilih menu " <i>Manage Kategori</i> "	Data Kategori	
4.	Menginputkan data kategori	Nama kategori	Data berhasil disimpan dan ditampilkan pada tabel <i>Manage Kategori</i>	
5.	Mengubah Data Kategori	Nama kategori	Data berhasil diubah dan ditampilkan pada tabel <i>Manage Kategori</i>	
6.	Menghapus Data Kategori	Memilih kategori yang akan di hapus	Muncul <i>pop up</i> konfirmasi hapus kategori dan jika di konfirmasi akan muncul notifikasi berhasil di hapus	

### b. Desain Uji Coba Pengelolaan Master Nilai

Desain uji coba pengelolaan master nilai bertujuan untuk menguji apakah fungsi *Create, Read, Update, Delete* (CRUD) pada master nilai dapat berjalan sesuai yang diharapkan. Desain uji coba pengelolaan master nilai dapat dilihat pada Tabel 3.54.

Tabel 3. 54 Desain Uji Coba *Form* Master Nilai

Test Case ID	Tujuan	<i>Input</i>	Output yang diharapkan	Status
7.	Menampilkan Data Nilai	Memilih menu “Manage Nilai”	Data Nilai	
8.	Menginputkan Data Nilai	Kriteria, sub kriteria, nilai batas bawah, nilai batas atas, nilai	Data berhasil disimpan dan ditampilkan pada tabel Manage Nilai	
9.	Mengubah Data Nilai	Kriteria, sub kriteria, nilai batas bawah, nilai batas atas, nilai	Data berhasil diubah dan ditampilkan pada tabel Manage Nilai	
10.	Menghapus Data Nilai	Memilih data Nilai yang akan di hapus	Muncul <i>pop up</i> konfirmasi hapus nilai dan jika di konfirmasi akan muncul notifikasi berhasil di hapus	

### c. Desain Uji Coba Pengelolaan Sub Kriteria

Desain uji coba pengelolaan sub kriteria bertujuan untuk menguji apakah fungsi *Create, Read, Update, Delete* (CRUD) pada sub kriteria dapat berjalan sesuai yang diharapkan. Desain uji coba pengelolaan sub kriteria dapat dilihat pada Tabel 3.55.

Tabel 3. 55 Desain Uji Coba *Form* Sub Kriteria

Test Case ID	Tujuan	<i>Input</i>	Output yang diharapkan	Status
11.	Menampilkan data sub kriteria	Memilih menu “Manage Sub Kriteria”	Data Sub Kriteria	
12.	Menginputkan data sub kriteria	Nama kriteria, nama sub kriteria	Data berhasil disimpan dan ditampilkan pada tabel Manage Sub Kriteria	
13.	Mengubah Data Sub Kriteria	Nama kriteria, nama sub kriteria	Data berhasil diubah dan ditampilkan pada tabel Manage Sub Kriteria	
14.	Menghapus Data Sub Kriteria	Memilih data Sub Kriteria yang akan di hapus	Muncul <i>pop up</i> konfirmasi hapus sub kriteria dan jika di konfirmasi akan muncul notifikasi berhasil di hapus	

#### d. Desain Uji Coba Pengelolaan Master *Bank*

Desain uji coba pengelolaan master *bank* bertujuan untuk menguji apakah fungsi *Create, Read, Update, Delete* (CRUD) pada master *bank* dapat berjalan sesuai yang diharapkan. Desain uji coba pengelolaan master *bank* dapat dilihat pada Tabel 3.56.

Tabel 3. 56 Desain Uji Coba *Form* Master *Bank*

Test Case ID	Tujuan	<i>Input</i>	Output yang diharapkan	Status
15.	Menampilkan data <i>bank</i>	Memilih menu “Manage <i>Bank</i> ”	Data <i>Bank</i>	
16.	Menginputkan data <i>bank</i>	Kode bank, Nama <i>Bank</i>	Data berhasil disimpan dan	

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
			ditampilkan pada tabel <i>Manage Bank</i>	
17.	Mengubah Data <i>Bank</i>	Kode <i>bank</i> , Nama <i>Bank</i>	Data berhasil diubah dan ditampilkan pada tabel <i>Manage Bank</i>	
18.	Menghapus Data <i>Bank</i>	Memilih data <i>Bank</i> yang akan di hapus	Muncul <i>pop up</i> konfirmasi hapus <i>Bank</i> dan jika di konfirmasi akan muncul notifikasi berhasil di hapus	

#### e. Desain Uji Coba Konfirmasi Pembayaran

Desain uji coba konfirmasi pembayaran bertujuan untuk menguji apakah fungsi *update* (terima/ditolak) status pada konfirmasi pembayaran dapat berjalan sesuai yang diharapkan. Desain uji coba konfirmasi pembayaran dapat dilihat pada Tabel 3.57.

Tabel 3. 57 Desain Uji Coba *Form* Konfirmasi Pembayaran

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
19.	Menampilkan list pembayaran yang akan dikonfirmasi	Memilih menu “Confirmation Payment”	Data <i>Confirmation Payment</i>	
20.	Menampilkan bukti transfer	Menekan ikon gambar bukti transfer	Menampilkan gambar bukti transfer secara utuh	
21.	Melakukan Konfirmasi Penerimaan Pembayaran	Menekan tombol konfirmasi	List bukti transfer yang telah dikonfirmasi menghilang dan jika di konfirmasi akan	

Test Case ID	Tujuan	<i>Input</i>	Output yang diharapkan	Status
			muncul notifikasi berhasil di konfirmasi	
22.	Melakukan Reject Penerimaan Pembayaran	Menekan tombol reject	List bukti transfer yang telah direject menghilang dan jika di konfirmasi akan muncul notifikasi berhasil di reject	

#### f. Desain Uji Coba *History Confirmation*

Desain uji coba *history confirmation* bertujuan untuk menguji apakah *order/transaksi* yang telah dikonfirmasi/ pernah dilakukan dapat dilihat kembali atau tidak (terekap atau tidak) sesuai yang diharapkan. Desain uji coba *history confirmation* dapat dilihat pada Tabel 3.58.

Tabel 3. 58 Desain Uji Coba *History Confirmation*

Test Case ID	Tujuan	<i>Input</i>	Output yang diharapkan	Status
23.	Menampilkan Data <i>History Confirmation</i>	Memilih menu “ <i>History Confirmation</i> ”	Data <i>History Confirmation</i>	

#### g. Desain Uji Coba *Legality Verification*

Desain uji coba *legality verification* bertujuan untuk menguji apakah data diri yang diinputkan sesuai dengan data yang dilampirkan sesuai yang diharapkan. Desain uji coba *legality verification* dapat dilihat pada Tabel 3.59.

Tabel 3. 59 Desain Uji Coba *Legality Verification*

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
24.	Menampilkan Data Vendor yang melakukan verifikasi	Memilih menu “Legality Verification”	Data <i>Legality Vendor</i>	
25.	Menampilkan Scan / gambar NPWP	Menekan ikon gambar NPWP	Menampilkan gambar NPWP secara utuh	
26.	Menampilkan Scan / gambar KTP	Menekan ikon gambar KTP	Menampilkan gambar KTP	
27.	Melakukan Konfirmasi <i>Legality Verification</i>	Menekan tombol konfirmasi	List <i>legality vendor</i> yang telah dikonfirmasi menghilang dan jika di konfirmasi akan muncul notifikasi berhasil di konfirmasi	
28.	Melakukan <i>Reject Legality Verification</i>	Menekan tombol <i>reject</i>	List <i>legality vendor</i> yang telah direject menghilang dan jika di konfirmasi akan muncul notifikasi berhasil di <i>reject</i>	

#### h. Desain Uji Coba *Dashboard Admin*

Desain uji coba *dashboard* pada admin bertujuan untuk menguji apakah fungsi data/grafik dapat ditampilkan dan memberi informasi yang *up to date* atau tidak sesuai yang diharapkan. Desain uji coba *dasboard* dapat dilihat pada Tabel 3.60.

Tabel 3. 60 Desain Uji Coba *Dashboard*

Test Case ID	Tujuan	<i>Input</i>	Output yang diharapkan	Status
29.	Menampilkan Data Dashboard	Memilih menu “Dashboard”	Data <i>Dashboard</i> yang ditampilkan dapat meng- <i>update</i> sesuai dengan adanya transaksi	

#### i. Desain Uji Coba Laporan *Customer* pada Admin

Desain uji coba laporan dilakukan terhadap laporan-laporan yang dihasilkan oleh sistem, yaitu laporan *customer*. Desain uji coba bertujuan apakah fungsi laporan yang dijalankan sesuai yang diharapkan atau memberikan rekapan *customer* yang dapat dicetak. Desain uji coba laporan *customer* dapat dilihat pada Tabel 3.61.

Tabel 3. 61 Desain Uji Coba Laporan *Customer* pada Admin

Test Case ID	Tujuan	<i>Input</i>	Output yang diharapkan	Status
30.	Menampilkan <i>pop up</i> periode laporan <i>customer</i>	Tekan tombol cetak pada <i>card box customer</i>	<i>Pop up</i> Laporan dengan periode	
31.	Menampilkan dokumen cetak laporan <i>customer</i>	Memilih periode laporan dan tekan tombol lihat laporan	Dokumen cetak berisi laporan sesuai periode yang dipilih	

#### j. Desain Uji Coba Laporan Vendor pada Admin

Desain uji coba laporan dilakukan terhadap laporan-laporan yang dihasilkan oleh sistem, yaitu laporan vendor. Desain uji coba bertujuan apakah fungsi laporan yang dijalankan sesuai yang diharapkan atau memberikan rekapan

vendor yang dapat dicetak. Desain uji coba laporan vendor dapat dilihat pada Tabel 3.62.

Tabel 3. 62 Desain Uji Coba Laporan Vendor pada Admin

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
32.	Menampilkan <i>pop up</i> periode laporan <i>vendor</i>	Tekan tombol cetak pada <i>card box vendor</i>	<i>Pop up</i> Laporan dengan periode	
33.	Menampilkan dokumen cetak laporan <i>vendor</i>	Memilih periode laporan dan tekan tombol lihat laporan	Dokumen cetak berisi laporan sesuai periode yang dipilih	

#### k. Desain Uji Coba Laporan Transaksi pada Admin

Desain uji coba laporan dilakukan terhadap laporan-laporan yang dihasilkan oleh sistem, yaitu laporan transaksi. Desain uji coba bertujuan apakah fungsi laporan yang dijalankan sesuai yang diharapkan atau memberikan rekapan transaksi yang dapat dicetak. Desain uji coba laporan transaksi dapat dilihat pada Tabel 3.63.

Tabel 3. 63 Desain Uji Coba Laporan Transaksi pada Admin

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
34.	Menampilkan <i>pop up</i> periode laporan transaksi	Tekan tombol cetak pada <i>card box customer</i>	Pop up Laporan dengan periode	
35.	Menampilkan dokumen cetak laporan transaksi	Memilih periode laporan dan tekan tombol lihat laporan	Dokumen cetak berisi laporan sesuai periode yang dipilih	



## I. Desain Uji Coba *Withdraw* / Penarikan

Desain uji coba *withdraw* bertujuan untuk menguji fungsi *withdraw*/penarikan dan CRU (*Create, Read, Update*) dapat berjalan sesuai yang diharapkan. Desain uji coba *withdraw* dapat dilihat pada Tabel 3.64.

Tabel 3. 64 Desain Uji Coba *Withdraw* Pada Admin

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
36.	Menampilkan list penarikan yang akan dikonfirmasi	Memilih menu “ <i>Confirmation Withdraw</i> ”	Data <i>Confirmation Withdraw</i>	
37.	Melakukan Konfirmasi Penarikan	Menekan tombol konfirmasi	List penarikan yang telah dikonfirmasi menghilang dan jika di konfirmasi akan muncul notifikasi berhasil di konfirmasi	
38.	Melakukan <i>Reject</i> Penarikan	Menekan tombol <i>reject</i>	List penarikan yang telah <i>direject</i> menghilang dan jika di konfirmasi akan muncul notifikasi berhasil di <i>reject</i>	

## C. Pengelolaan Pada *Role Vendor*

### a. Desain Uji Coba Registrasi Vendor

Desain uji coba registrasi vendor bertujuan untuk menguji apakah fungsi registrasi/daftar akun dapat berjalan sesuai yang diharapkan. Desain uji coba registrasi vendor dapat dilihat pada Tabel 3.65.

Tabel 3. 65 Desain Uji Coba *Form* Registrasi Vendor

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
39.	Registrasi Vendor	<i>First Name, Last Name, Email, Password, Confirm Password</i>	Registrasi Vendor Berhasil	

### b. Desain Uji *Legality* Vendor

Desain uji coba *legality* vendor bertujuan untuk menguji apakah fungsi *create* dan *upload* lampiran dapat berjalan sesuai yang diharapkan. Desain uji coba *legality* vendor dapat dilihat pada Tabel 3.66.

Tabel 3. 66 Desain Uji Coba *Form Legality* Vendor

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
40.	Registrasi Legalitas Vendor	<i>Form data pimpinan dan Form legalitas perusahaan</i>	Notifikasi “Mohon menunggu data anda akan diverifikasi oleh admin”	

### c. Desain Uji Coba *Dashboard* Vendor

Desain uji coba *dashboard* pada vendor bertujuan untuk menguji apakah fungsi data/grafik dapat ditampilkan dan memberi informasi yang *up to date* atau tidak sesuai yang diharapkan. Desain uji coba *dashboard* dapat dilihat pada Tabel 3.67.

Tabel 3. 67 Desain Uji Coba *Dashboard*

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
41.	Menampilkan Data Dashboard	Memilih menu “Dashboard”	Data <i>Dashboard</i> yang ditampilkan	

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
			dapat meng- <i>update</i> sesuai dengan adanya transaksi	

#### d. Desain Uji Coba Pengelolaan Data Master Kategori Vendor

Desain uji coba pengelolaan data master kategori vendor bertujuan untuk menguji apakah fungsi *Create, Read, Update, Delete* (CRUD) pada data master kategori vendor dapat berjalan sesuai yang diharapkan. Desain uji coba pengelolaan master kategori vendor dapat dilihat pada Tabel 3.68.

Tabel 3. 68 Desain Uji Coba *Form* Master Kategori Vendor

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
42.	Menampilkan Data Kategori <i>Vendor</i>	Memilih menu “Category <i>Vendor</i> ”	Data Kategori <i>Vendor</i>	
43.	Menginputkan Data Kategori <i>Vendor</i>	Nama Brand, Jenis Kategori, Alamat Perusahaan, Nomor Telpon	Data berhasil disimpan dan ditampilkan pada <i>card Our Category Business</i>	
44.	Menghapus Kategori <i>Vendor</i>	Menekan tombol hapus kategori bisnis	Muncul <i>pop up</i> untuk konfirmasi hapus, jika tekan ok data akan dihapus jika <i>cancel</i> data tidak terhapus	
45.	Menampilkan halaman <i>View Package</i>	Menekan tombol <i>view package</i>	Menampilkan halaman <i>view package</i>	

#### e. Desain Uji Coba Pengelolaan Data Master Paket Vendor

Desain uji coba pengelolaan data master paket vendor bertujuan untuk menguji apakah fungsi *Create, Read, Update, Delete* (CRUD) pada master paket

vendor dapat berjalan sesuai yang diharapkan. Desain uji coba pengelolaan master paket vendor dapat dilihat pada Tabel 3.69.

Tabel 3. 69 Desain Uji Coba *Form* Master Paket Vendor

Test Case ID	Tujuan	<i>Input</i>	Output yang diharapkan	Status
46.	Menginputkan Data Paket <i>Vendor</i>	Nama Paket, Harga Paket, Deskripsi Paket, Unggah Gambar	Data berhasil disimpan dan ditampilkan pada card <i>Our Package</i>	
47.	Mengubah Data Paket <i>Vendor</i>	Nama Paket, Harga Paket, Deskripsi Paket, Unggah Gambar	Data berhasil diubah dan ditampilkan pada <i>card Our Package</i>	
48.	Menghapus <i>Package</i>	Menekan tombol hapus <i>package</i>	Muncul pop up untuk konfirmasi hapus, jika tekan ok data akan dihapus jika cancel data tidak terhapus	

#### f. Desain Uji Coba Pengelolaan *Profile* Vendor

Desain uji coba pengelolaan *profile* vendor bertujuan untuk menguji apakah fungsi *Read, Update* (RU) pada pengelolaan *profile* vendor dapat berjalan sesuai yang diharapkan. Desain uji coba pengelolaan *profile* vendor dapat dilihat pada Tabel 3.70.

Tabel 3. 70 Desain Uji Coba *Form Profile* Vendor

Test Case ID	Tujuan	<i>Input</i>	Output yang diharapkan	Status
49.	Menampilkan <i>User Profile</i>	Memilih menu “ <i>View Profile</i> ”	Data <i>User Profile</i>	
50.	Mengubah <i>Profile</i>	<i>Input</i> data <i>profile</i>	Data <i>profile</i> telah diperbarui	

### g. Desain Uji Coba Konfirmasi *Order*

Desain uji coba konfirmasi *order* bertujuan untuk menguji apakah fungsi *update* (terima/ditolak) status pada konfirmasi *order* dapat berjalan sesuai yang diharapkan. Desain uji coba konfirmasi *order* dapat dilihat pada Tabel 3.71.

Tabel 3. 71 Desain Uji Coba *Form* Konfirmasi *Order*

Test Case ID	Tujuan	<i>Input</i>	Output yang diharapkan	Status
51.	Menampilkan Daftar <i>Order</i> Baru	Memilih menu “ <i>Incoming Order</i> ”	Data <i>Order</i> Baru	
52.	Melakukan Konfirmasi <i>Order</i>	Menekan tombol konfirmasi	List <i>order</i> yang telah dikonfirmasi menghilang dan jika dikonfirmasi akan muncul notifikasi berhasil di konfirmasi	
53.	Melakukan <i>Reject Order</i>	Menekan tombol <i>reject</i>	List <i>order</i> yang telah direject menghilang dan jika dikonfirmasi akan muncul notifikasi berhasil di <i>reject</i>	

### h. Desain Uji Coba Jadwal Vendor

Desain uji coba jadwal vendor bertujuan untuk menguji apakah fungsi CRU (*Create, Read, Update*) dapat berjalan sesuai yang diharapkan. Desain uji coba jadwal vendor dapat dilihat pada Tabel 3.72.

Tabel 3. 72 Desain Uji Coba Jadwal Vendor

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
54.	Menampilkan <i>pop up form</i> tambah jadwal vendor	Pilih tombol jadwal	Menampilkan <i>pop up form</i> jadwal	

#### D. Pengelolaan Pada *Role Customer*

##### a. Desain Uji Coba Registrasi *Customer*

Desain uji coba registrasi *customer* bertujuan untuk menguji apakah fungsi registrasi/daftar akun dapat berjalan sesuai yang diharapkan. Desain uji coba registrasi *customer* dapat dilihat pada Tabel 3.73.

Tabel 3. 73 Desain Uji Coba *Form* Registrasi *Customer*

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
55.	Registrasi <i>Customer</i>	<i>Email, Password, Confirm Password</i>	Registrasi <i>Customer</i> Berhasil	

##### b. Desain Uji Coba Pengelolaan *Profile Customer*

Desain uji coba pengelolaan *profile customer* bertujuan untuk menguji apakah fungsi *Read, Update* (RU) pada pengelolaan *profile customer* dapat berjalan sesuai yang diharapkan. Desain uji coba pengelolaan *profile customer* dapat dilihat pada Tabel 3.74.

Tabel 3. 74 Desain Uji Coba *Form* *Profile Customer*

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
56.	Menampilkan <i>User Profile</i>	Memilih menu "View Profile"	Data <i>User Profile</i>	
57.	Mengubah <i>Profile</i>	Input data profile	Data profile telah diperbarui	

### c. Desain Uji Coba Pencarian Vendor

Desain uji coba pencarian vendor bertujuan untuk menguji apakah fungsi *search* dapat berjalan sesuai yang diharapkan. Desain uji coba pencarian vendor dapat dilihat pada Tabel 3.75.

Tabel 3. 75 Desain Uji Coba Pencarian Vendor

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
58.	Menampilkan Daftar <i>Brand</i>	Input data nama <i>brand</i>	Data <i>brand</i> sesuai <i>input</i>	

### d. Desain Uji Coba *Customize Order*

Desain uji coba *customize order* bertujuan untuk menguji apakah fungsi *Create, Read, Update* (CRU) dan pemesanan dapat berjalan sesuai yang diharapkan. Desain uji coba *customize order* dapat dilihat pada Tabel 3.76.

Tabel 3. 76 Desain Uji Coba *Customize Order*

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
59.	Menampilkan Profile <i>Brand</i> ( <i>Our package</i> )	Tekan tombol lihat <i>profile</i>	Menampilkan profile <i>brand</i> dan paket yang dimiliki	
60.	Menampilkan Profile <i>Brand</i> ( <i>Review</i> )	Tekan tab <i>review</i> lihat <i>profile</i>	Menampilkan profile <i>brand</i> dan semua ulasan	
61.	Menampilkan form pemberian <i>rating</i> dan <i>review</i>	Tekan tombol <i>review</i> sekarang	Menampilkan <i>pop up form rating and review</i>	
62.	Menampilkan <i>cart</i> dan <i>form</i> transaksi	Tekan tombol <i>add to cart</i> pada paket yang dipilih dan tekan tombol keranjang pada bagian atas	Menampilkan daftar paket yang akan dibeli pada <i>cart</i> dan menampilkan form transaksi	

### e. Desain Uji Coba *Generate* Paket dan Pemesanan Paket

Desain uji coba *generate* paket dan pemesanan paket bertujuan untuk menguji apakah fungsi *Create*, *Read* (RU), *generate* paket (menghasilkan pilihan paket untuk ditawarkan kepada *customer*), dan pemesanan dapat berjalan sesuai yang diharapkan. Desain uji coba *generate* paket dan pemesanan paket dapat dilihat pada Tabel 3.77.

Tabel 3. 77 Desain Uji Coba *Form Generate* Paket dan Pemesanan Paket

Test Case ID	Tujuan	<i>Input</i>	Output yang diharapkan	Status
63.	Menampilkan <i>Generate</i> Paket	Memilih menu “ <i>Generate Package</i> ”	Menampilkan <i>Form Generate Package</i>	
64.	Menentukan Kategori yang dibutuhkan dan menentukan prioritas kategori dan sub kriteria	<i>Budget</i> , <i>Category</i> , <i>tanggal acara</i> , <i>kota</i>	list pilihan kategori dan sub kriteria	
65.	Membuat pilihan paket pernikahan	Menekan tombol <i>generate</i>	Daftar paket dengan berbagai <i>category vendor</i> , pilihan paket dari <i>vendor</i>	
66.	Menampilkan <i>pop up</i> dari detail paket	Tekan tombol (i)	Menampilkan <i>pop up</i> detail paket	
67.	Menampilkan <i>form</i> pemesanan	Menekan tombol <i>choose</i>	Menampilkan <i>form</i> pemesanan	

### f. Desain Uji Coba Histori Transaksi (Pemesanan)

Desain uji coba histori transaksi bertujuan untuk menguji apakah *order*/transaksi yang telah dikonfirmasi/pernah dilakukan dapat dilihat kembali atau tidak (terekap atau tidak) sesuai yang diharapkan. Desain uji coba histori transaksi dapat dilihat pada Tabel 3.78.



Tabel 3. 78 Desain Uji Coba *Form* Histori Transaksi

Test Case ID	Tujuan	<i>Input</i>	Output yang diharapkan	Status
68.	Menampilkan Histori Transaksi	Memilih menu “ <i>History Transaction</i> ”	Data <i>history transaction</i>	
69.	Melihat Detail Transaksi	Menekan tombol Lihat Detail	Menampilkan pop up daftar <i>category vendor</i>	

### g. Desain Uji Coba Konfirmasi Pembayaran

Desain uji coba konfirmasi pembayaran bertujuan untuk menguji apakah fungsi *update* (terima/ditolak) status pada konfirmasi pembayaran dapat berjalan sesuai yang diharapkan. Desain uji coba konfirmasi pembayaran dapat dilihat pada Tabel 3.79.

Tabel 3. 79 Desain Uji Coba *Form* Konfirmasi Pembayaran Pada *Customer*

Test Case ID	Tujuan	<i>Input</i>	Output yang diharapkan	Status
70.	Menampilkan konfirmasi pembayaran	Bank, Nomor Rekening, Jumlah Transfer, Tanggal Transfer, Unggah Gambar	Menampilkan pesan “Konfirmasi pembayaran berhasil”	

### E. Desain Uji Coba Notifikasi

Desain uji coba notifikasi bertujuan untuk menguji apakah fungsi *Read*, *Update* (RU) dapat berjalan sesuai yang diharapkan. Desain uji coba notifikasi dapat dilihat pada Tabel 3.80.

Tabel 3. 80 Desain Uji Coba Notifikasi

Test Case ID	Tujuan	<i>Input</i>	Output yang diharapkan	Status
71.	Menampilkan notifikasi secara	Tekan tombol notifikasi	Notifikasi terbaru pada	

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
	ringkas pada tombol notifikasi		tombol notifikasi	
72.	Menampilkan notifikasi secara keseluruhan	Tekan tombol <i>see all notications</i>	Seluruh list notifikasi	

#### F. Desain Uji Coba *Forgot Password*

Desain uji coba *forgot password* bertujuan untuk menguji apakah fungsi *update password* dapat berjalan sesuai yang diharapkan. Desain uji coba *forgot password* dapat dilihat pada Tabel 3.81.

Tabel 3. 81 Desain Uji Coba *Form Forgot Password*

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
73.	Membuat <i>link reset password</i>	<i>Email</i>	<i>Link reset password</i> berhasil dikirim ke email anda	
74.	Membuat <i>Password Baru</i>	<i>New Passwprd, Confirm Password</i>	Menampilkan pesan “ <i>Password berhasil dirubah</i> ”	

#### G. Desain Uji Coba *Withdraw*

Desain uji coba *withdraw* bertujuan untuk menguji fungsi *withdraw*/penarikan dan CRU (*Create, Read, Update*) dapat berjalan sesuai yang diharapkan. Desain uji coba *withdraw* dapat dilihat pada Tabel 3.82.

Tabel 3. 82 Desain Uji Coba *Withdraw*

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
75.	Menampilkan <i>history withdraw</i>	Tekan tombol <i>withdraw</i>	Menampilkan list <i>history withdraw</i>	

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
76.	Menampilkan <i>form withdraw</i>	Tekan tab <i>withdraw</i>	Menampilkan <i>form withdraw</i>	



## BAB IV

### IMPLEMENTASI DAN EVALUASI

#### 4.1 Implementasi Sistem

Tahap implementasi sistem merupakan tahapan untuk membuat perangkat lunak yang disesuaikan dengan rancangan atau desain sistem yang telah dirancang dan dikembangkan sebelumnya. Implementasi sistem digunakan untuk menampilkan dan menjelaskan fitur-fitur yang ada pada aplikasi penentuan *wedding* untuk usaha *Wedding Organizer* menggunakan metode SMARTER berbasis *web*. Tahapan implementasi perlu memperhatikan spesifikasi kebutuhan sistem yang akan digunakan untuk dapat mengakses dan menggunakan sistem. Kebutuhan sistem yang perlu diperhatikan ialah kebutuhan perangkat keras dan kebutuhan perangkat lunak.

##### 4.1.1 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras digunakan untuk membentuk atau menjalankan fungsi suatu sistem komputer terstruktur. Kebutuhan *hardware* yang dibutuhkan pada penelitian ini adalah sebagai berikut:

1. *Client Side*

- *Processor* Intel Core Duo
- Ram 2 GB
- Screen

2. *Server Side*

- *Processor* Intel Core i5

- RAM 8 GB dan SSD 500 GB
- Monitor dengan resolusi 1024 x 768
- *Mouse & Keyboard*

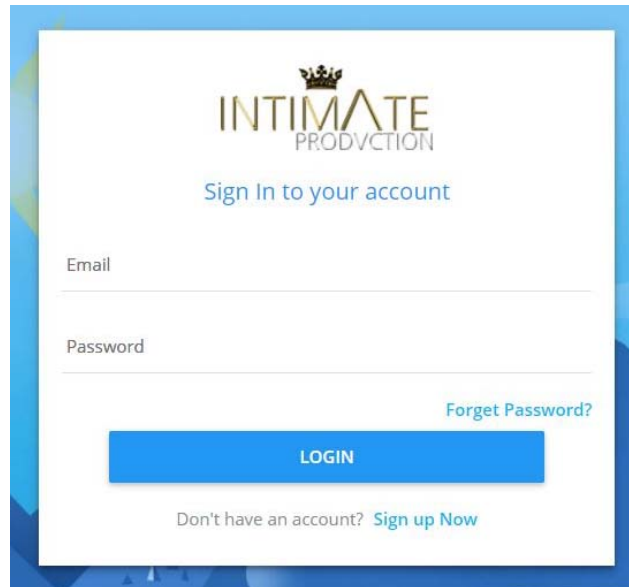
#### 4.1.2 Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak yang digunakan untuk mengembangkan dan membangun aplikasi penentuan *vendor* Pernikahan untuk usaha *Wedding Organizer* menggunakan metode SMARTER berbasis *web* adalah sebagai berikut:

1. *Microsoft Visio 2013* untuk membuat rancangan *document flow* dan *system flow*.
2. *Power Designer 15.2* untuk membuat *context diagram*, *data flow diagram*, *conceptual data model* dan *physical data model*.
3. XAMPP untuk *web server*.
4. MySQL sebagai *Database Management System (DBMS)*.
5. *Codeigniter* versi 3.0.0 sebagai *framework* aplikasi.
6. *Mockingbot* untuk membuat perancangan *interface* aplikasi.

#### 4.1.3 Implementasi Halaman *Login* Pengguna

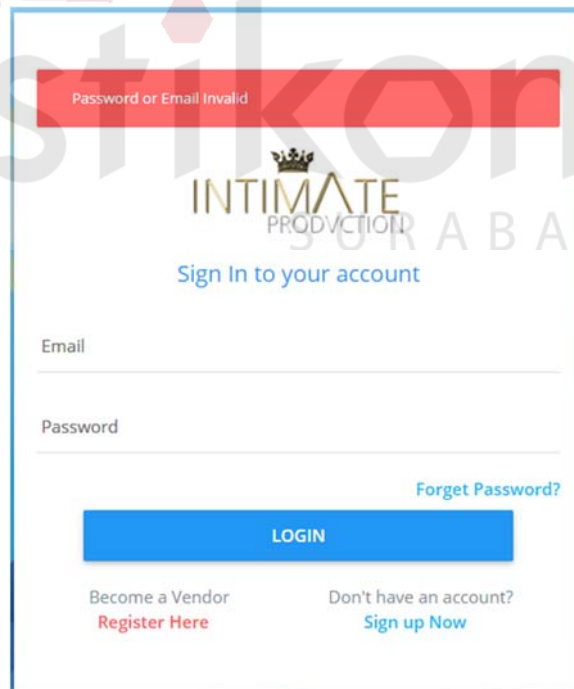
Halaman login merupakan halaman pertama bagi pengguna untuk masuk kedalam aplikasi penentuan *vendor*. Akses aplikasi oleh pengguna dibagi berdasarkan peran dari masing-masing pengguna yang ada. Untuk masuk kedalam aplikasi, pengguna hanya memasukkan *email* dan *password* yang telah terdaftar atau ada di *database*. Berikut adalah gambar dari implementasi halaman login yang ditunjukkan pada Gambar 4.1.



The screenshot shows the login page for Intimate Production. At the top center is the logo "INTIMATE PRODUCTION" with a crown icon above the word "INTIMATE". Below the logo is the text "Sign In to your account". There are two input fields: "Email" and "Password". To the right of the password field is a link "Forget Password?". Below the input fields is a blue "LOGIN" button. At the bottom, there is a link "Don't have an account? Sign up Now".

Gambar 4. 1 Halaman Login Pengguna

Jika email atau password yang di inputkan salah, maka aplikasi akan menampilkan pesan error seperti Gambar 4. 2



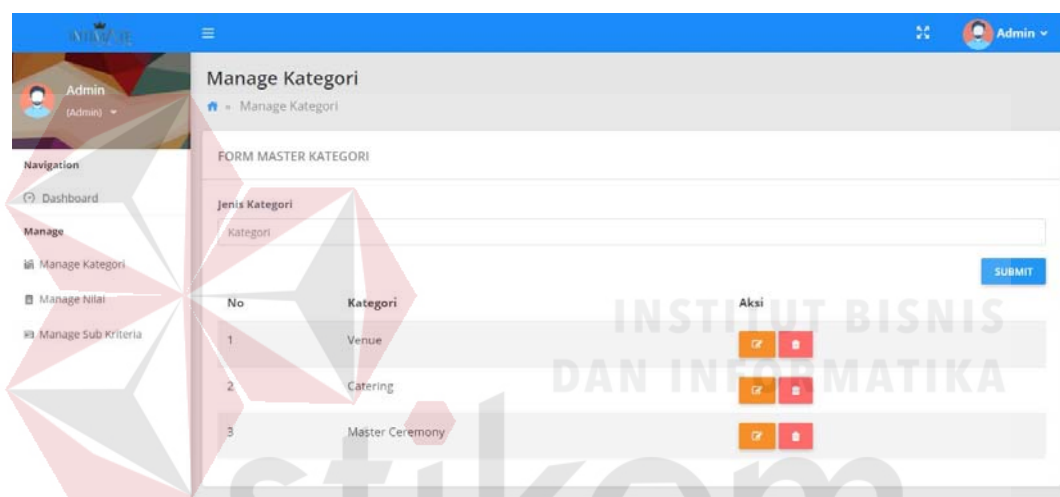
The screenshot shows the same login page as in Gambar 4.1, but with an error message. A red banner at the top of the form area contains the text "Password or Email Invalid". Below the banner, the logo and "Sign In to your account" text are visible. The "Email" and "Password" input fields are present. To the right of the password field is a link "Forget Password?". Below the input fields is a blue "LOGIN" button. At the bottom, there are two links: "Become a Vendor Register Here" and "Don't have an account? Sign up Now".

Gambar 4. 2 Pesan *Login Error*

#### 4.1.4 Implementasi Halaman Pada Admin

##### A. Pengelolaan Master Kategori

Implementasi halaman pengelolaan master kategori merupakan halaman yang digunakan untuk pengelolaan data master kategori yang dilakukan oleh admin yang nantinya digunakan oleh *vendor* saat mendaftar dengan memilih kategori yang sesuai dengan *vendor* tersebut. Berikut adalah gambar dari implementasi halaman pengelolaan master kategori yang ditunjukkan pada Gambar 4.3.



Gambar 4. 3 Implementasi Halaman Pengelolaan Master Kategori

Ketika data berhasil di tambahkan kedalam database, aplikasi akan memberikan pesan seperti pada Gambar 4.4. Untuk melakukan perubahan data kategori, pengguna dapat menekan tombol berwarna kuning, aplikasi akan menampilkan data yang akan diubah kedalam form. Setelah data dirubah dan disimpan, sistem akan menampilkan pesan seperti Gambar 4. 5



Gambar 4. 4 Notifikasi *Add New Category Success*

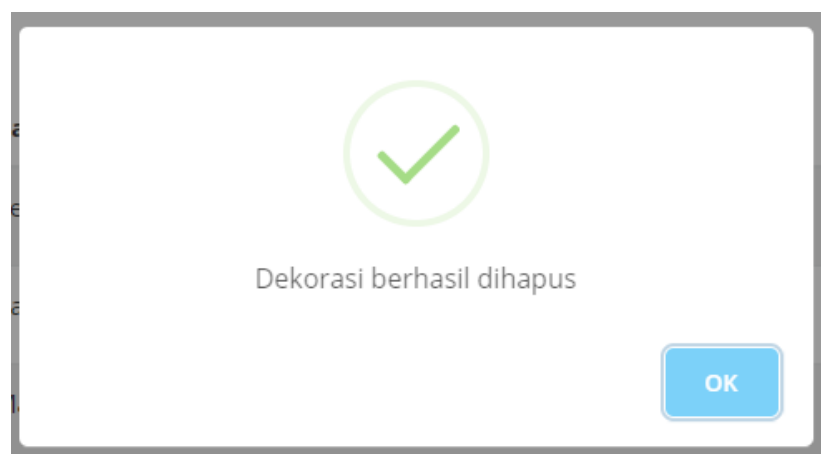


Gambar 4. 5 Notifikasi *Update Category Success*

Sedangkan ketika pengguna menghapus data, dapat menekan tombol berwarna merah, aplikasi akan memunculkan konfirmasi untuk menghapus seperti Gambar 4. 6. Ketika data berhasil dihapus aplikasi akan menampilkan notifikasi seperti Gambar 4. 7



Gambar 4. 6 Konfirmasi Hapus Data Kategori



Gambar 4. 7 Notifikasi data kategori berhasil dihapus



## B. Pengelolaan Master Nilai

Implementasi halaman pengelolaan master nilai merupakan halaman yang digunakan untuk pengelolaan data master nilai untuk memberikan nilai kriteria dan sub kriteria berdasarkan range *budget*. Pengelolaan master nilai dilakukan oleh admin sistem dengan memilih kriteria, sub kriteria, input nilai batas bawah, nilai batas atas dan nilai. Berikut adalah gambar dari implementasi halaman pengelolaan master nilai yang ditunjukkan pada Gambar 4.8.

No	Kriteria	Sub Kriteria	Nilai Batas Bawah	Nilai Batas Atas	Nilai	Aksi
1	Venue	Harga	0	500000	80	[Edit] [Delete]
2	Venue	Rating	1	2	30	[Edit] [Delete]

Gambar 4. 8 Implementasi Halaman Pengelolaan Master Nilai

Ketika data berhasil di tambahkan kedalam database, aplikasi akan memberikan pesan seperti pada Gambar 4. 9. Untuk melakukan perubahan data nilai, pengguna dapat menekan tombol berwarna kuning, aplikasi akan menampilkan data yang akan diubah kedalam form. Setelah data dirubah dan disimpan, sistem akan menampilkan pesan seperti Gambar 4. 10.



Gambar 4. 9 Notifikasi *Add Nilai Success*

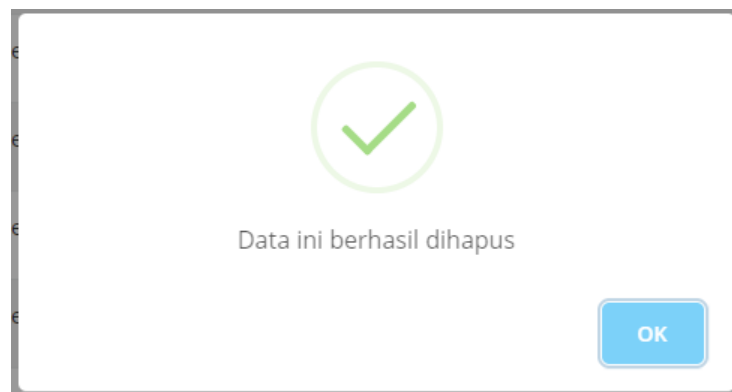


Gambar 4. 10 Notifikasi *Update Master Nilai Success*

Sedangkan ketika pengguna menghapus data, dapat menekan tombol berwarna merah, aplikasi akan memunculkan konfirmasi untuk menghapus seperti Gambar 4. 11. Ketika data berhasil dihapus aplikasi akan menampilkan notifikasi seperti Gambar 4. 12.



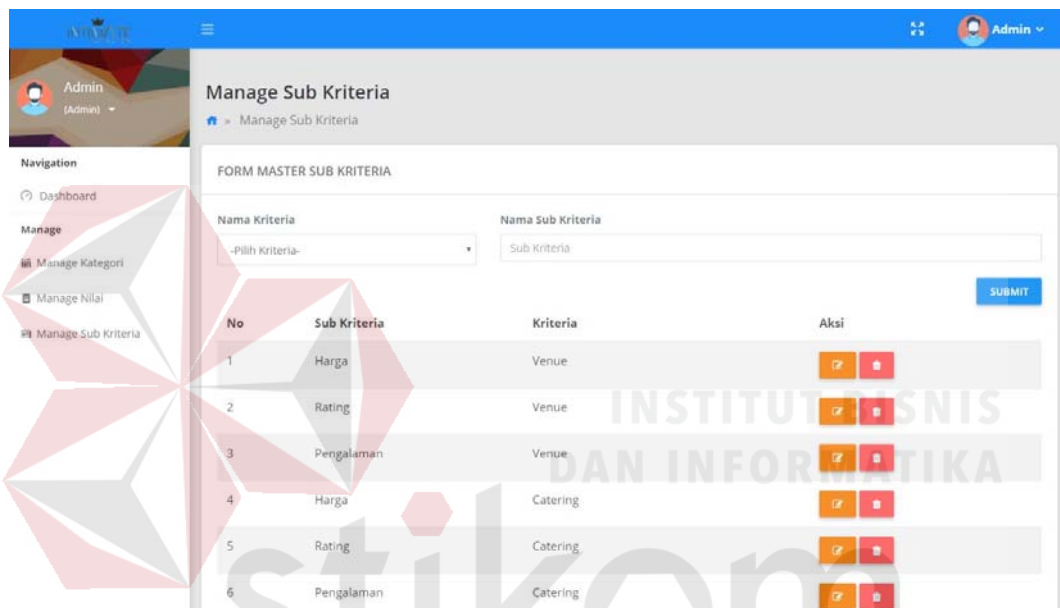
Gambar 4. 11 Konfirmasi hapus data nilai



Gambar 4. 12 Notifikasi berhasil hapus data nilai

### C. Pengelolaan Sub Kriteria

Implementasi halaman pengelolaan master sub kriteria merupakan halaman yang digunakan admin untuk pengelolaan data master sub kriteria dari setiap kriteria yang dilakukan oleh admin sistem. Berikut adalah gambar dari implementasi halaman pengelolaan master sub kriteria yang ditunjukkan pada Gambar 4.13

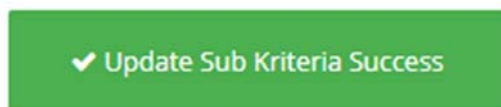


Gambar 4. 13 Implementasi Halaman Pengelolaan Master Sub Kriteria

Ketika data berhasil di tambahkan kedalam database, aplikasi akan memberikan pesan seperti pada Gambar 4. 14. Untuk melakukan perubahan data sub kriteria, pengguna dapat menekan tombol berwarna kuning, aplikasi akan menampilkan data yang akan diubah kedalam form. Setelah data dirubah dan disimpan, sistem akan menampilkan pesan seperti Gambar 4. 15

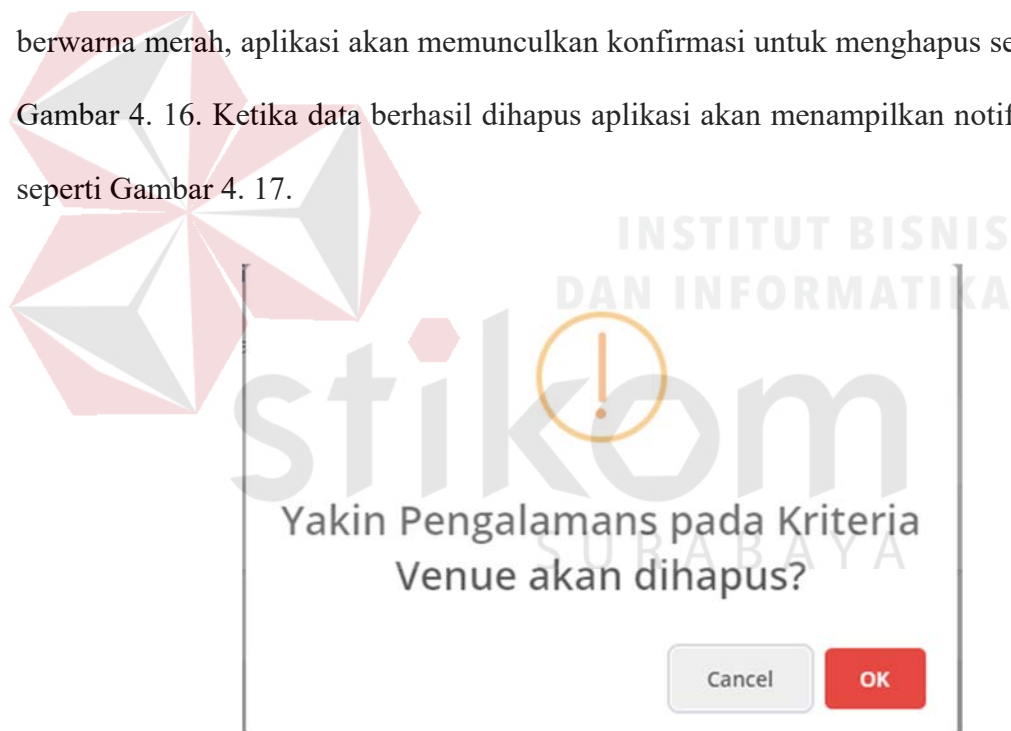


Gambar 4. 14 Notifikasi *Add New Sub Kriteria Success*

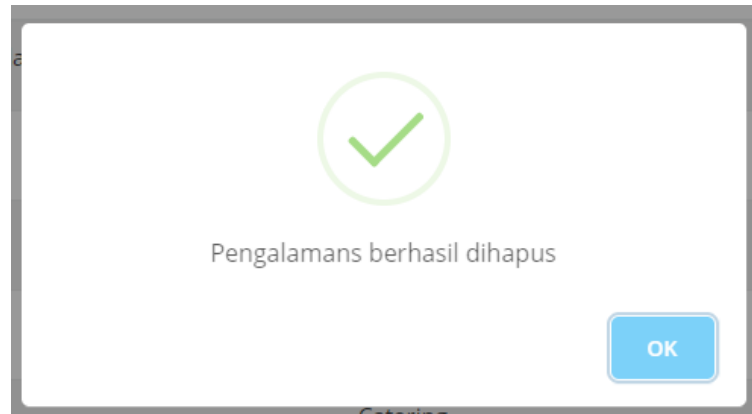


Gambar 4. 15 Notifikasi *Update Sub Kriteria Success*

Sedangkan ketika pengguna menghapus data, dapat menekan tombol berwarna merah, aplikasi akan memunculkan konfirmasi untuk menghapus seperti Gambar 4. 16. Ketika data berhasil dihapus aplikasi akan menampilkan notifikasi seperti Gambar 4. 17.



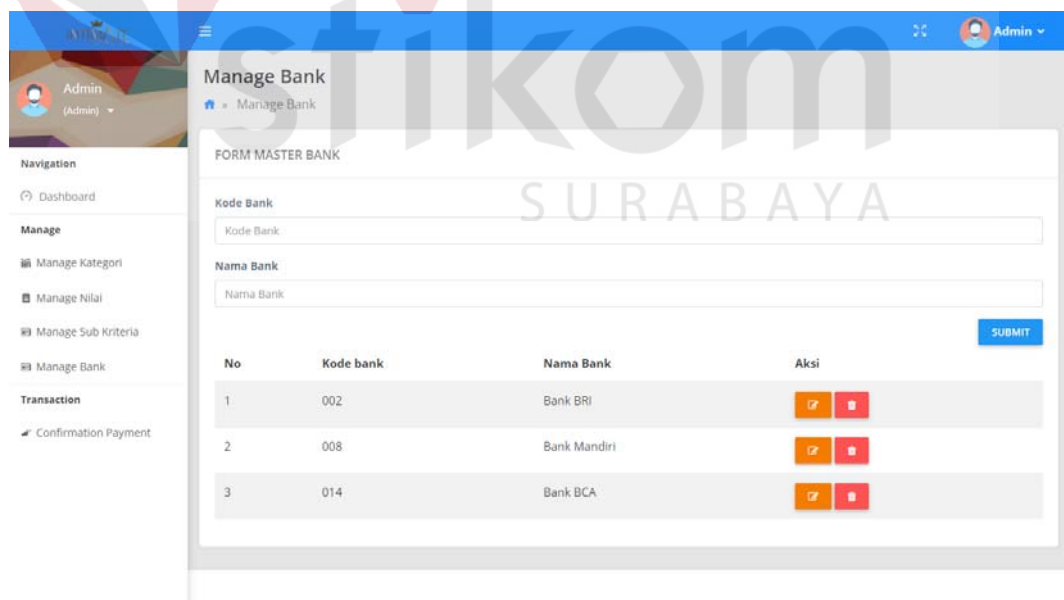
Gambar 4. 16 Konfirmasi Hapus Data Sub Kriteria



Gambar 4. 17 Notifikasi hapus data sukses

#### D. Pengelolaan Bank

Implementasi halaman *manage bank* merupakan halaman yang digunakan admin untuk mengelola (menambah, mengubah dan menghapus data) master *bank* berupa kode *bank* dan nama *bank*. Berikut adalah gambar dari implementasi halaman pengelolaan master *bank* yang ditunjukkan pada Gambar 4.18



Gambar 4. 18 Implementasi Halaman Pengelolaan Master *Bank*

Ketika data berhasil di tambahkan kedalam database, aplikasi akan memberikan pesan seperti pada Gambar 4.19. Untuk melakukan perubahan data sub kriteria, pengguna dapat menekan tombol berwarna kuning, aplikasi akan menampilkan data yang akan diubah kedalam *form*. Setelah data dirubah dan disimpan, sistem akan menampilkan pesan seperti Gambar 4.20

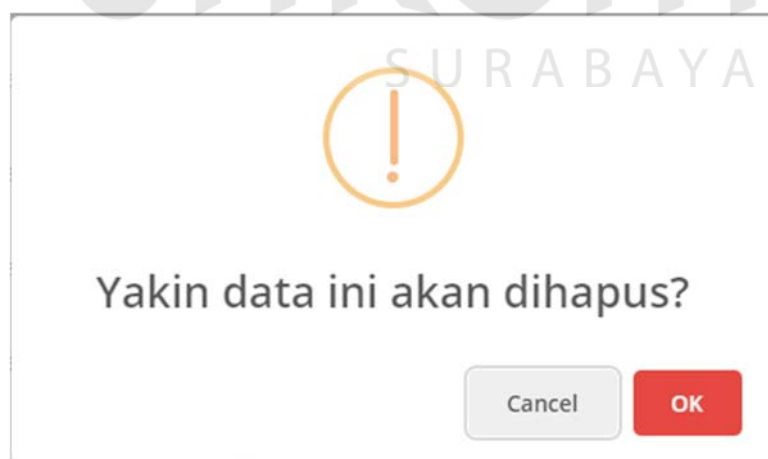


Gambar 4. 19 Notifikasi *Add New Bank Success*

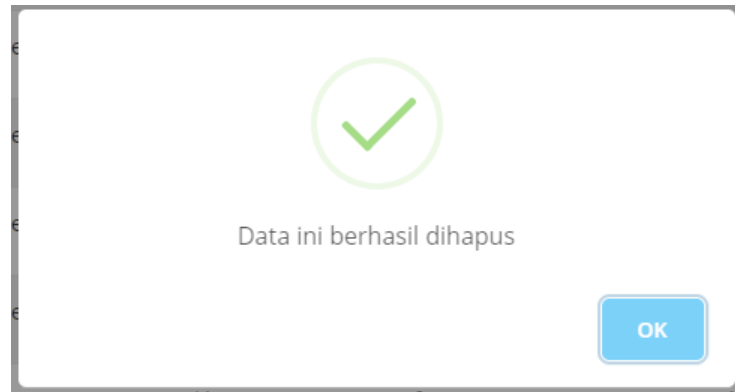


Gambar 4. 20 Notifikasi *Update Master Bank Success*

Sedangkan ketika pengguna menghapus data, dapat menekan tombol berwarna merah, aplikasi akan memunculkan konfirmasi untuk menghapus seperti Gambar 4. 21. Ketika data berhasil dihapus aplikasi akan menampilkan notifikasi seperti Gambar 4. 22



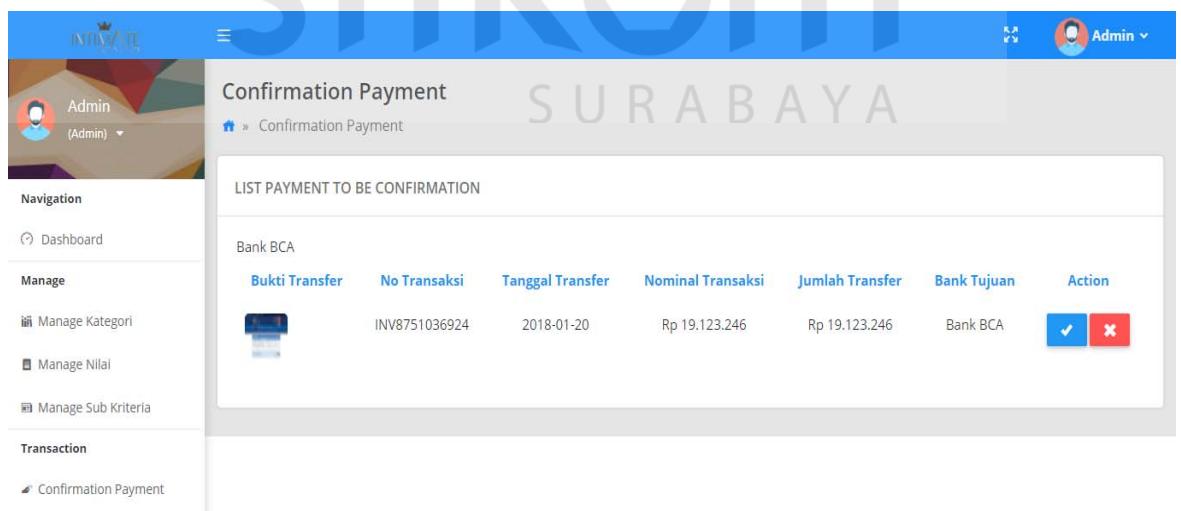
Gambar 4. 21 Konfirmasi hapus data *bank*



Gambar 4. 22 Notifikasi berhasil hapus data *bank*

### E. Konfirmasi Pembayaran

Implementasi halaman konfirmasi pembayaran merupakan halaman yang digunakan admin untuk mengkonfirmasi dan mengecek pembayaran yang dilakukan oleh *customer*. Berikut adalah gambar dari implementasi halaman konfirmasi pembayaran yang ditunjukkan pada Gambar 4.23. Sedangkan ketika foto / gambar pada bukti transfer di klik maka, aplikasi akan menampilkan foto / gambar lebih jelas dan besar seperti pada Gambar 4.24



Gambar 4. 23 Implementasi Halaman Konfirmasi Pembayaran Pada Admin

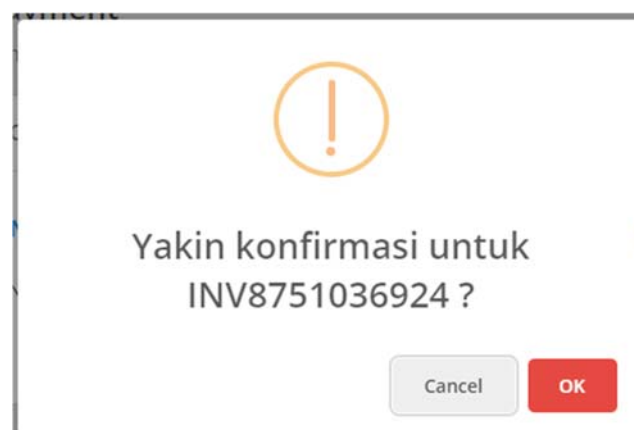
KARTU RENCANA STUDI							SEMESTER GASAL 2017	
Hari	Waktu	Matakuliah (Kelas)	Brilian Ruang	SKS	Nilai	Nilai Minimal	Kehadiran	Keterangan
Senin	07:30 - 10:00	Manajemen Layanan Teknologi Informasi (P1)	Site	A305	3	Detail	C	100%
Selasa	13:30 - 16:00	Manajemen Proyek Teknologi Informasi (P1)	-	A404	3	Detail	B	92,86%
		Tugas Akhir (P1)	-		6	Detail	C	%

Note:  
 \*Nilai untuk method detail nilai per matakuliah  
 \*Nilai kehadiran untuk method detail kehadiran per matakuliah  
 \*Nilai matakuliah untuk method detail matakuliah

[Column View](#)
[Save PDF](#)
[Print](#)

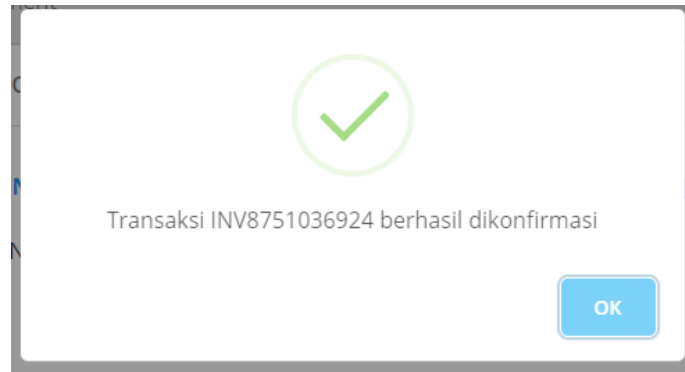
Gambar 4. 24 Implementasi Menampilkan Bukti Transfer

Ketika pengguna telah melakukan pencocokan bukti transfer dengan rekening perusahaan, pengguna dapat melakukan konfirmasi dengan menekan tombol konfirmasi berwarna biru, setelah itu aplikasi akan memastikan sebelum melakukan konfirmasi seperti Gambar 4. 25 dan ketika dikonfirmasi akan muncul notifikasi seperti Gambar 4. 26. Sementara untuk melakukan reject, pengguna dapat menekan tombol reject berwarna merah, dan aplikasi akan memastikan sebelum melakukan reject seperti Gambar 4.27 dan ketika direject akan muncul notifikasi seperti Gambar 4. 28.

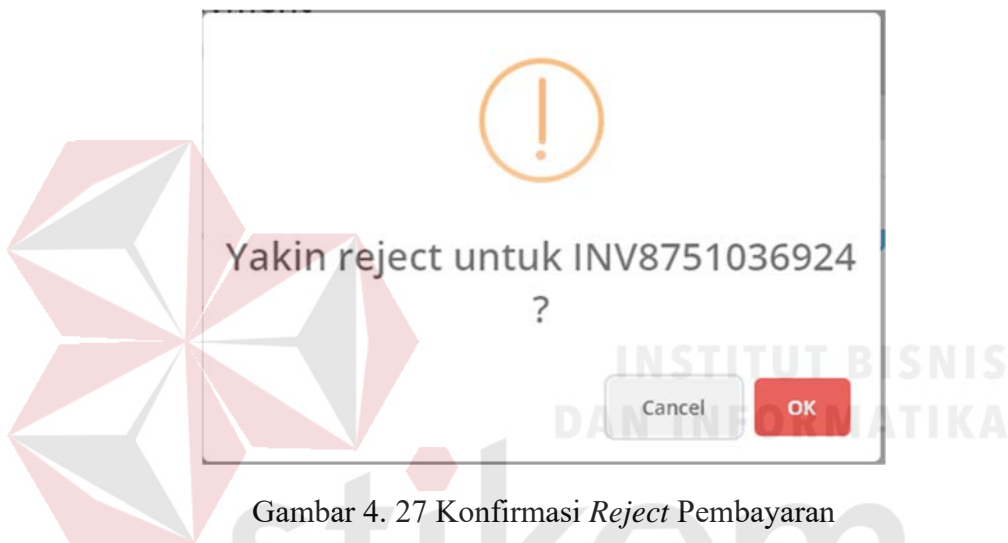


Gambar 4. 25 Konfirmasi Penerimaan Pembayaran





Gambar 4. 26 Notifikasi Kofirmasi Penerimaan Pembayaran



Gambar 4. 27 Konfirmasi *Reject* Pembayaran

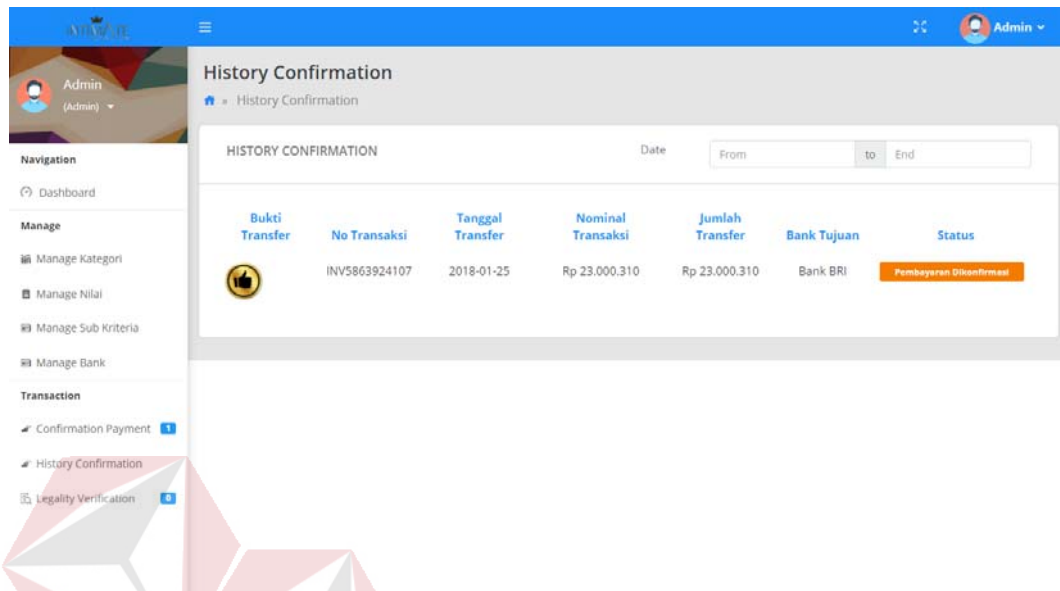


Gambar 4. 28 Notifikasi *Reject* Pembayaran

## F. Histori Konfirmasi Pembayaran

Implementasi halaman histori konfirmasi merupakan halaman yang digunakan admin untuk melihat riwayat konfirmasi pembayaran dan statusnya.

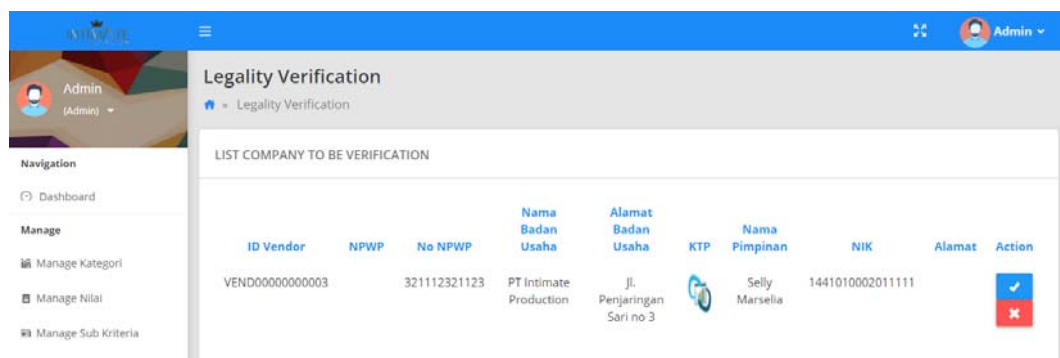
Berikut adalah gambar dari implementasi halaman histori konfirmasi pembayaran yang ditunjukkan pada Gambar 4.29.



Gambar 4. 29 Implementasi Halaman Histori Pembayaran Pada Admin

### G. Legality Verification

Implementasi halaman *legality verification* merupakan halaman yang digunakan admin untuk mengecek data legal *vendor* yang mendaftar berupa nomor NPWP, nama badan usaha, alamat, KTP, nama pimpinan, NIK pimpinan, dan alamat pimpinan. Berikut adalah gambar dari implementasi halaman *legality verification* pembayaran yang ditunjukkan pada Gambar 4.30.



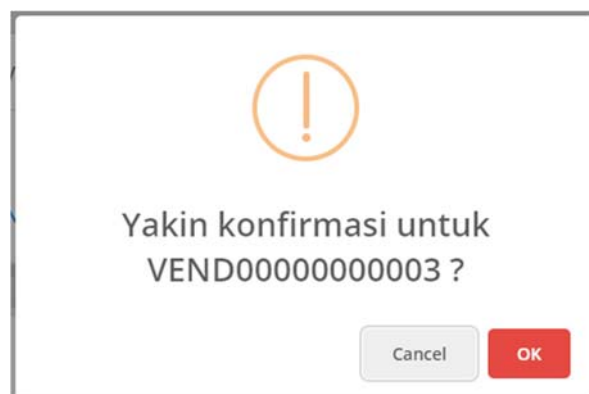
Gambar 4. 30 Implementasi Halaman *Legality Verification*



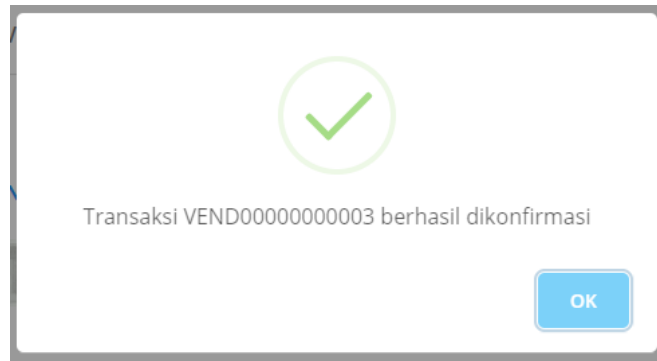
Gambar 4. 31 Implementasi View NPWP *Legality Verification*



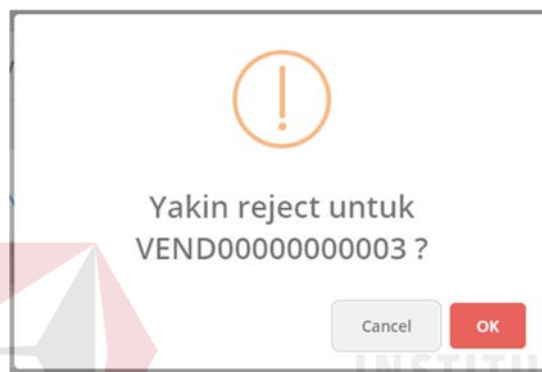
Gambar 4. 32 Implementasi View KTP *Legality Verification*



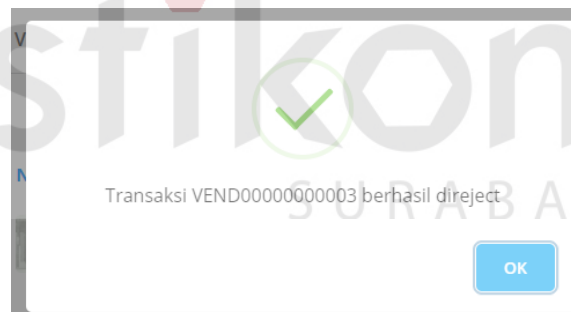
Gambar 4. 33 Konfirmasi *Legality Verification*



Gambar 4. 34 Notifikasi Konfirmasi *Legality Verification*



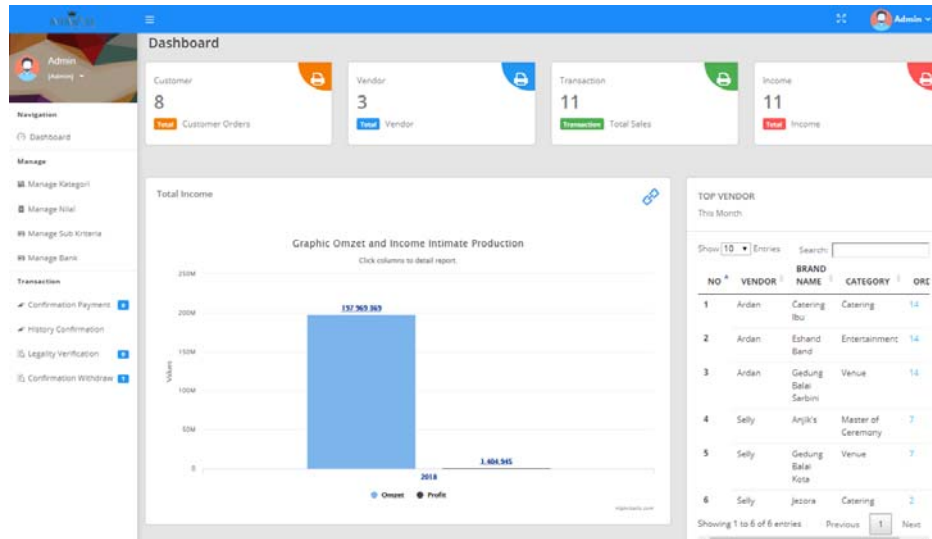
Gambar 4. 35 Konfirmasi *Reject Legality Verification*



Gambar 4. 36 Notifikasi *Legality Verification*

## H. *Dashboard Admin*

Berikut merupakan implementasi halaman *dashboard* yang terdiri dari laporan member vendor, member *customer*, transaksi, pendapatan, peringkat vendor dan grafik pendapatan setiap bulan pada admin yang dapat dilihat pada Gambar 4.37.



Gambar 4. 37 Implementasi Halaman *Dashboard* Pada Admin

## I. Laporan *Customer*

Berikut merupakan implementasi laporan member *customer* yang terdiri dari nama, email dan tanggal daftar yang dapat dicetak berdasarkan bulan dalam tahun tertentu yang dapat dilihat pada Gambar 4.38.

Report Customer

Tahun: 2018  
Pilih Periode Laporan

Bulan: Januari  
Pilih Periode Laporan

**LIHAT LAPORAN**

No	Nama Depan	Email	Tanggal Daftar
1	Fachreza	ilhmrezafx@gmail.com	2017-11-29 08:17:41
2	cosment	cosmentcosment@gmail.com	2017-12-29 04:11:38
3	as	asd@e.d	2018-01-09 06:36:27
4	Fikri	ilhmreza002@gmail.com	2018-02-07 01:04:41
5	John	john001@gmail.com	2018-02-08 03:26:12
6	Fikri	fikri@gmail.com	2018-02-13 08:42:58
7	abdul	abdul@gmail.com	2018-02-13 08:44:31
8	Aditya	Adit.anugrahp@gmail.com	2018-02-15 08:05:23

**CLOSE** **CETAK LAPORAN**

Gambar 4. 38 Implementasi Halaman Laporan *Customer*



**Intimate Production**  
 Wedding Organizer  
 Sidoarjo Indah Permai Blok B-1  
 087851305598

### Report Customer

Periode: 2017

Dicetak Pada : 12-02-2018

No.	Nama Depan	Email	Tanggal Daftar
1	Fachreza	Ilhmreza01@gmail.com	2017-11-29 08:17:41
2	Cosment	Cosmentcosment@gmail.com	2017-12-29 04:11:38
3	As	Asd@E.D	2018-01-09 06:36:27
4	Fikri	Ilhmreza002@gmail.com	2018-02-07 01:04:41
5	John	John001@gmail.com	2018-02-08 03:26:12

Gambar 4. 39 Implementasi Halaman Cetak Laporan *Customer*

## J. Laporan Vendor

Berikut merupakan implementasi laporan member vendor yang terdiri dari nama, email dan tanggal daftar yang dapat dicetak berdasarkan bulan dalam tahun tertentu yang dapat dilihat pada Gambar 4.40.

Report Vendor ×

Tahun

Pilih Periode Laporan

Bulan

Pilih Periode Laporan

**LIHAT LAPORAN**

No	Nama Depan	Email	Tanggal Daftar
1	Ardan	ilhmreza001@gmail.com	2017-11-29 10:05:03
2	Selly	ismaile@gmail.com	2017-12-02 09:51:17

Gambar 4. 40 Implementasi Halaman Laporan Vendor



**Intimate Production**  
Wedding Organizer  
Sidoarjo Indah Permai Blok B-1  
087851305598

**Report Vendor**  
Periode: 2017

Dicetak Pada : 12-02-2018

No.	Nama Depan	Email	Tanggal Daftar
1	Ardan	ilh mreza001@gmail.com	2017-11-29 10:05:03
2	Selly	ismaile@gmail.com	2017-12-02 09:51:17

Gambar 4. 41 Implementasi Halaman Cetak Laporan Vendor

### K. Laporan Transaksi

Berikut merupakan implementasi laporan transaksi yang terdiri dari *invoice*, total biaya, dan tanggal transaksi yang dapat dicetak berdasarkan bulan dalam tahun tertentu yang dapat dilihat pada Gambar 4.42.

Report Transaction
×

**Tahun**

2018
▼

Pilih Periode Laporan

**Bulan**

Februari
▼

Pilih Periode Laporan

LIHAT LAPORAN

No	Nama Depan	Email	Tanggal Daftar
1	Ardan	ilh mreza001@gmail.com	2017-11-29 10:05:03
2	Selly	ismaile@gmail.com	2017-12-02 09:51:17

CLOSE

CETAK LAPORAN

Gambar 4. 42 Implementasi Halaman Laporan Transaksi



**Intimate Production**  
**Wedding Organizer**  
 Sidoarjo Indah Permai Blok B-1  
 087851305598

### Report Transaction

Periode: 2017

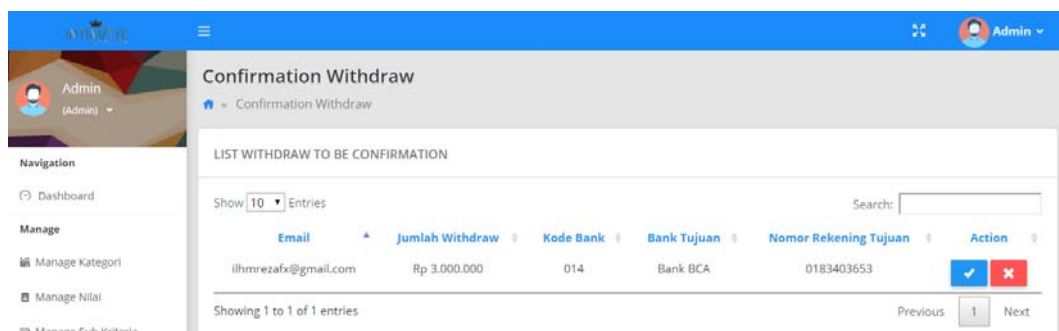
Dicetak Pada : 12-02-2018

No.	Invoice	Total Biaya	Tanggal Transaksi
1	INV5863924107	Rp 23.000.000	2018-01-24 06:08:32
2	INV6948305271	Rp 26.123.123	2018-02-08 05:34:46
3	INV8740365912	Rp 46.000.000	2018-01-30 09:24:33

Gambar 4. 43 Implementasi Halaman Cetak Laporan Transaksi

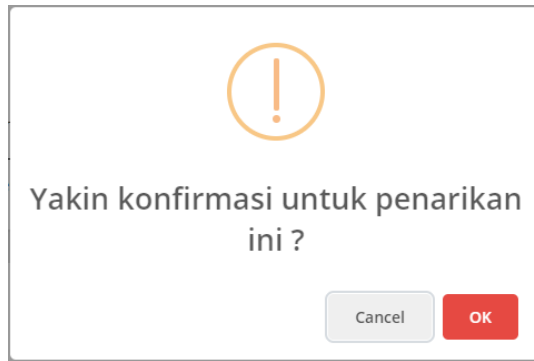
### L. Konfirmasi Penarikan

Berikut merupakan implementasi konfirmasi penarikan saldo yang dilakukan oleh *customer* untuk dikonfirmasi oleh admin. Data yang disimpan pada konfirmasi ini terdiri atas *email*, jumlah *withdraw*, kode *bank*, *bank* tujuan, nomor rekening dan aksi untuk melakukan konfirmasi disetujui atau ditolak. Implementasi konfirmasi penarikan dapat dilihat pada Gambar 4.44.

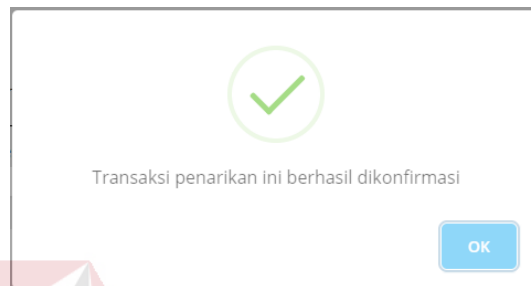


Gambar 4. 44 Implementasi Konfirmasi Penarikan Pada Admin

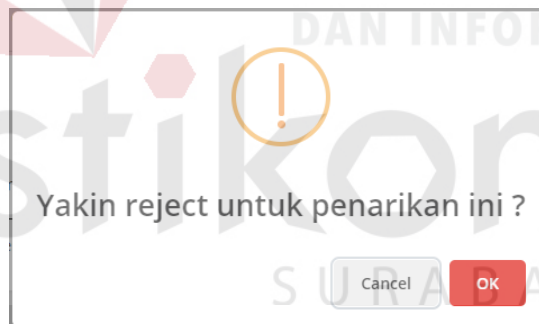




Gambar 4. 45 Konfirmasi Penarikan



Gambar 4. 46 Notifikasi Penarikan Berhasil Dikonfirmasi

Gambar 4. 47 Konfirmasi *Reject* PenarikanGambar 4. 48 Notifikasi Penarikan Berhasil di *Reject*

#### 4.1.5 Implementasi Halaman Pada *Vendor*

##### A. Registrasi *Vendor*

Implementasi halaman registrasi *vendor* merupakan syarat yang harus dilakukan oleh *vendor* untuk masuk kedalam aplikasi dengan melakukan pendaftaran terlebih dahulu dengan memasukkan *email* dan *password vendor* agar dapat masuk kedalam aplikasi. Berikut adalah gambar dari implementasi halaman registrasi *vendor* yang ditunjukkan pada Gambar 4.49.



The image shows a web page for 'INTIMATE PRODUCTION' with a registration form. At the top, there is a green banner that says 'Register success'. Below the banner is the company logo and the text 'Create an account for Your Business'. The form has three input fields: 'Email Address', 'Password', and 'Confirm Password'. A blue button labeled 'SIGN UP' is positioned below the form. At the bottom of the form, there is a link that says 'Already have an account? Sign In Here'. The page is overlaid with a large, semi-transparent watermark that reads 'stikom SURABAYA' and 'INSTITUT BISNIS DAN INFORMATIKA'.

Gambar 4. 49 Implementasi Halaman Registrasi Vendor

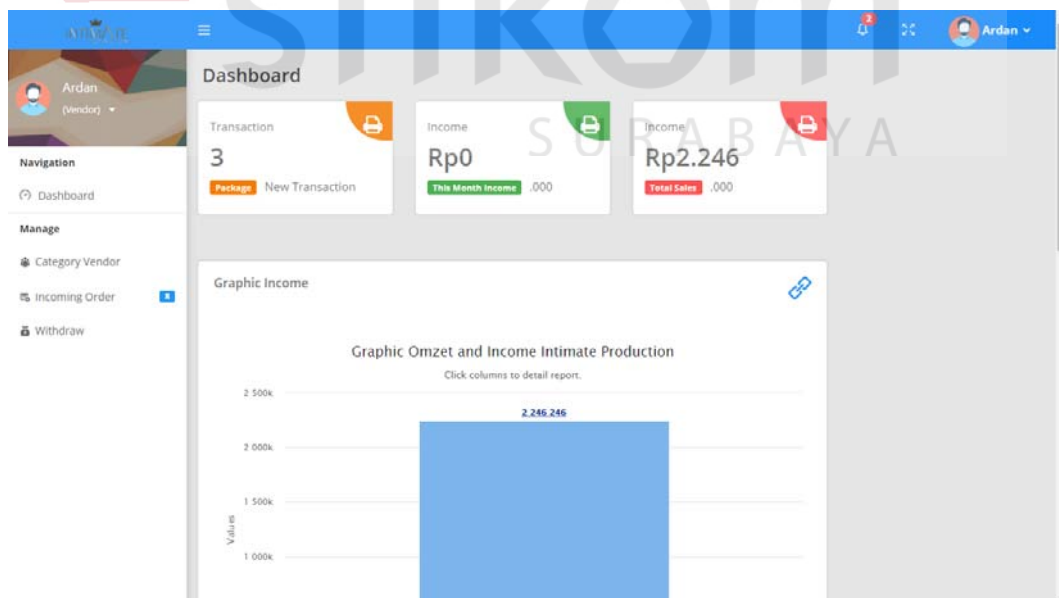
##### B. *Legality Vendor*

Implementasi *Legality Vendor* adalah tahapan setelah melakukan registrasi, yang digunakan untuk melengkapi data legalitas dari perusahaan, sehingga hanya perusahaan legal yang dapat bergabung demi kenyamanan *customer*. Vendor dapat mengisi form seperti pada Gambar 4.50.

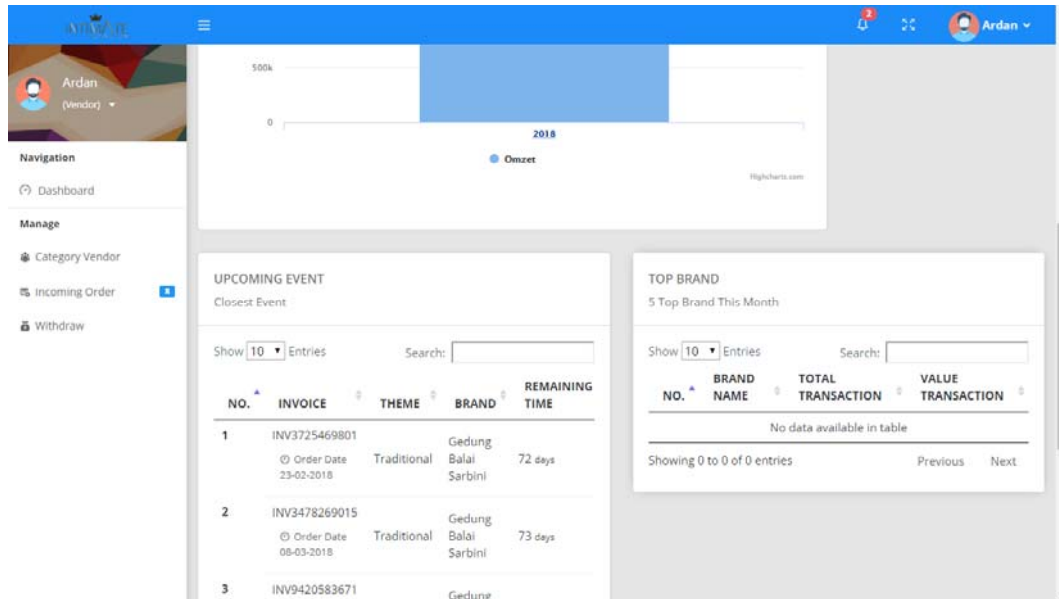
Gambar 4. 50 Implementasi Halaman *Legality Vendor*

### C. *Dashboard Vendor*

Berikut merupakan implementasi halaman *dashboard* pada vendor yang terdiri dari transaksi, *invoice*, pendapatan, grafik pendapatan, *upcoming event*, dan *top brand* yang dapat dilihat pada Gambar 4.51 dan Gambar 4.52.



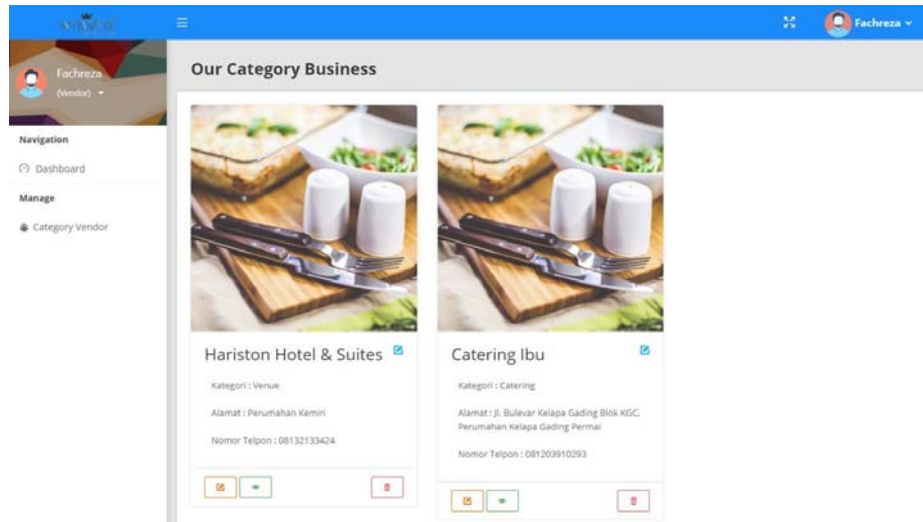
Gambar 4. 51 Implementasi Halaman *Dashboard* Pada Vendor 1

Gambar 4. 52 Implementasi Halaman *Dashboard 2*

#### D. Pengelolaan Kategori Vendor

Gambar 4.53 dibawah menunjukkan gambaran dari implementasi halaman pengelolaan kategori yang dilakukan oleh vendor dengan menginputkan nama *brand*, memilih jenis kategori, alamat perusahaan dan nomor yang dapat dihubungi. Sedangkan Gambar 4.54 menunjukkan implementasi halaman *list* kategori *vendor* yang telah mendaftar/bekerjasama.

Gambar 4. 53 Implementasi Halaman Pengelolaan Kategori oleh Vendor

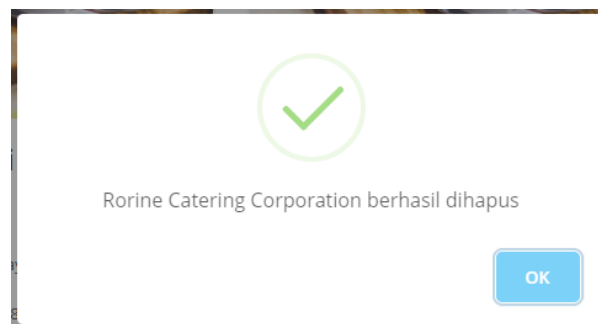


Gambar 4. 54 Implementasi Halaman *List* Kategori Vendor



Gambar 4. 55 Notifikasi Add New Category Vendor

Gambar 4. 56 Konfirmasi Hapus *Category* Vendor



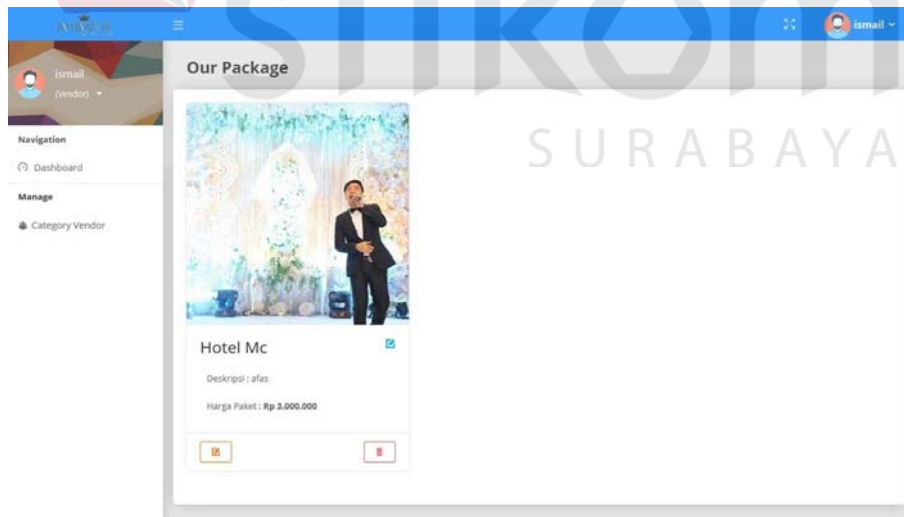
Gambar 4. 57 Notifikasi Hapus *Category* Vendor Success

## E. Pengelolaan Data Paket Vendor

Gambar 4.58 adalah implementasi pengelolaan data paket vendor yang merupakan halaman yang digunakan oleh vendor untuk membuat paket pernikahan dengan menginputkan nama paket, harga paket, deskripsi paket, dan unggah gambar. Sedangkan Gambar 4.59 merupakan implementasi halaman *list* paket pernikahan dari setiap vendor.



Gambar 4. 58 Implementasi Halaman Pengelolaan Data Paket Vendor



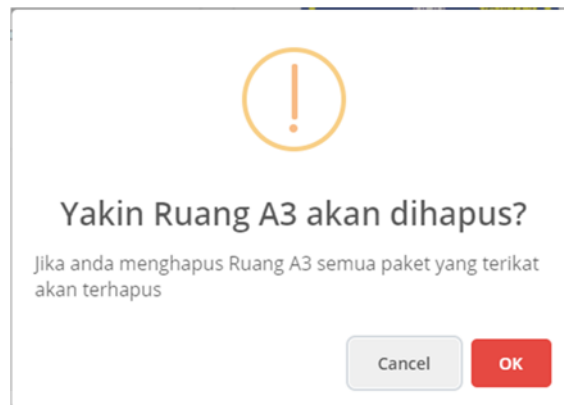
Gambar 4. 59 Implementasi Halaman *List* Paket Vendor



Gambar 4. 60 Notifikasi *Add New Package Success*



Gambar 4. 61 Notifikasi *Update Package Success*



Gambar 4. 62 Konfirmasi Hapus *Package*



Gambar 4. 63 Notifikasi *Package* Berhasil Dihapus

## F. Pengelolaan *Profile Vendor*

Implementasi halaman pengelolaan profil *vendor* merupakan halaman yang digunakan *vendor* untuk menambah dan mengubah data *vendor*. Berikut adalah gambar dari implementasi halaman pengelolaan profil *vendor* pembayaran yang ditunjukkan pada Gambar 4.64.

Gambar 4. 64 Implementasi Halaman Pengelolaan *Profile Vendor*

### G. Konfirmasi *Order*

Implementasi halaman konfirmasi *order* merupakan halaman untuk mengonfirmasi pesanan/*order* dari *customer* terhadap *vendor* yang dipilih. Berikut adalah gambar dari implementasi halaman konfirmasi *order vendor* pembayaran yang ditunjukkan pada Gambar 4.65.

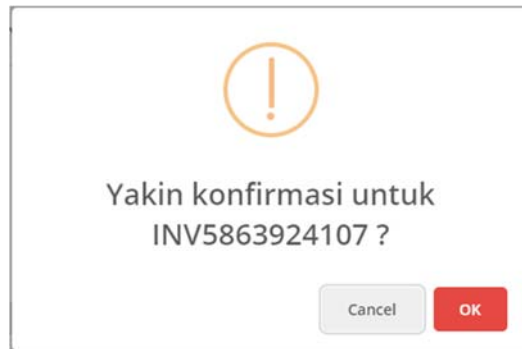
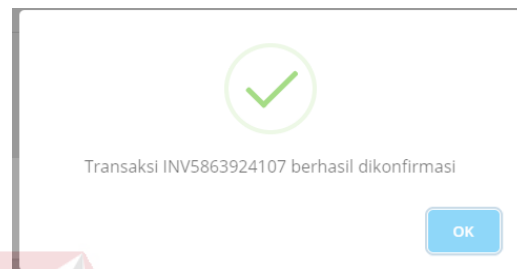
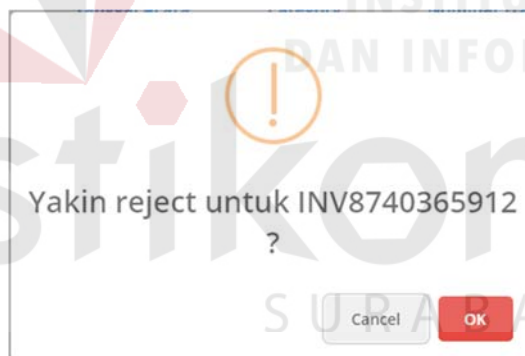
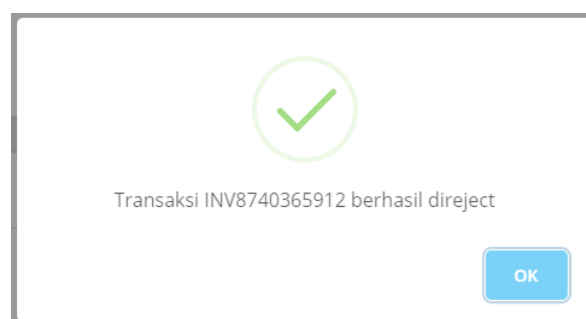
No Transaksi	Tanggal Acara	Category	Nominal Transaksi	Customer	Action
INV8751036924	0000-00-00	Gedung Balai Kota	Rp 15.000.000	Fachreza	<input checked="" type="checkbox"/> <input type="checkbox"/>

No Transaksi	Tanggal Acara	Category	Nominal Transaksi	Customer	Status
INV8751036924	0000-00-00	Anjik's	Rp 3.000.000	Fachreza	Order Diterima

Gambar 4. 65 Implementasi Halaman Konfirmasi *Order* Pada Vendor



Gambar 4. 66 Konfirmasi *Order*Gambar 4. 67 Notifikasi Konfirmasi *Order*Gambar 4. 68 Konfirmasi *Reject Order*Gambar 4. 69 Notifikasi Order Berhasil di *Reject*

## H. Jadwal Vendor

Implementasi halaman jadwal vendor merupakan halaman untuk mencatat jadwal vendor dengan menginputkan nama acara dan tanggal acara yang dilakukan oleh vendor. Berikut adalah gambar dari implementasi halaman jadwal vendor yang dapat dilihat pada Gambar 4.70.

Jadwal Acara Gedung Balai Sarbini

Tambahkan Jadwal Acara Yang Anda Memiliki

Nama Acara

Tanggal Acara

TAMBAH ACARA

No	Nama Acara	Tanggal Acara	Aksi
1	INV5863924107 - Rustic	23-05-2018	 

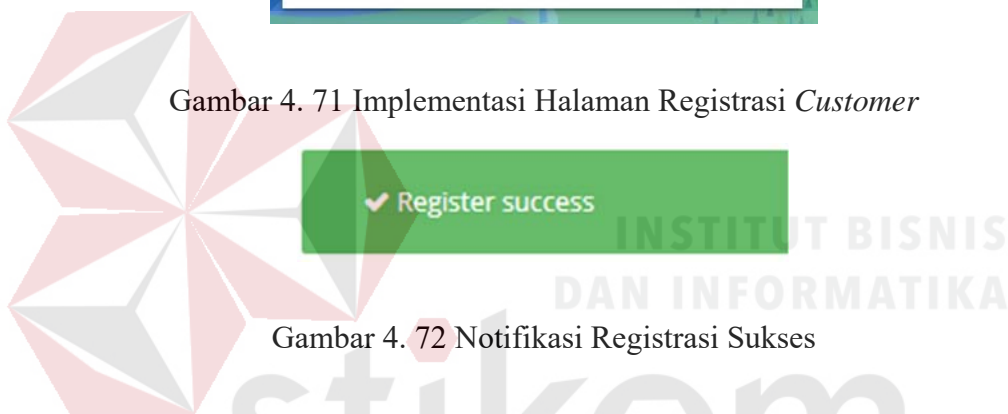
Gambar 4. 70 Implementasi Halaman Jadwal Vendor

### 4.1.6 Implementasi Halaman Pada *Customer*

#### A. Registrasi *Customer*

Implementasi halaman registrasi *customer* merupakan syarat yang harus dilakukan oleh *customer* untuk masuk kedalam aplikasi atau yang akan memesan *vendor* dengan melakukan pendaftaran terlebih dahulu dengan memasukkan identitas, *email* dan *password customer* agar dapat masuk kedalam aplikasi. Berikut adalah gambar dari implementasi halaman registrasi *customer* yang ditunjukkan pada Gambar 4.71.

Gambar 4. 71 Implementasi Halaman Registrasi *Customer*



Gambar 4. 72 Notifikasi Registrasi Sukses

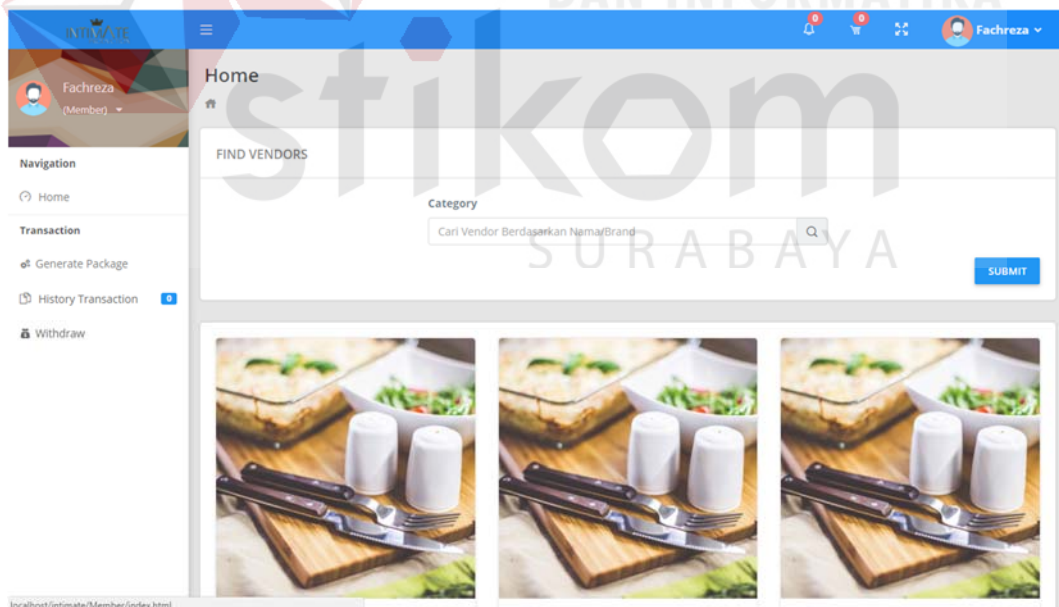
## B. Pengelolaan *Profile Customer*

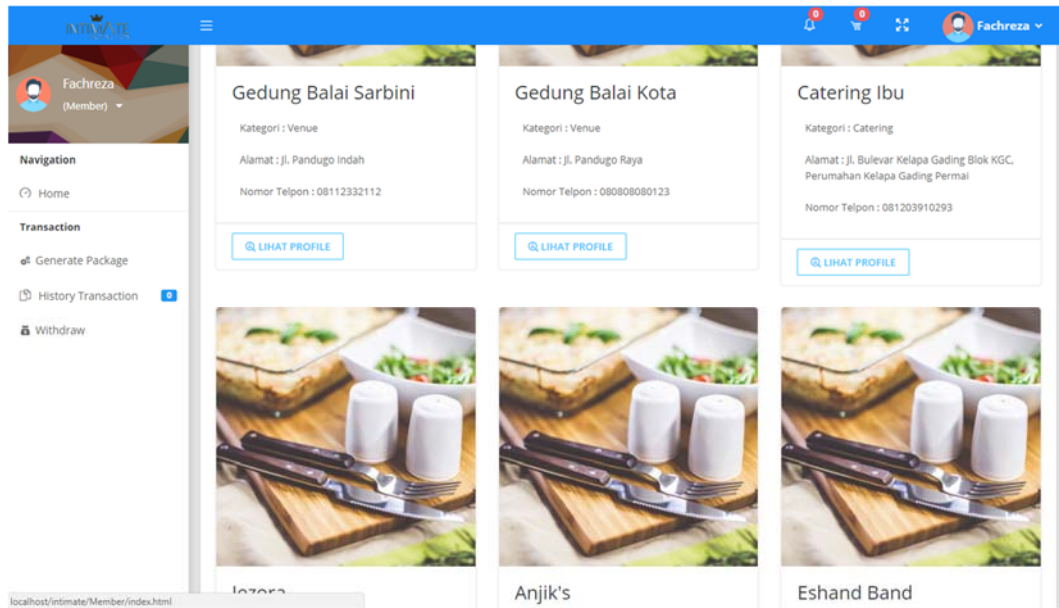
Implementasi halaman pengelolaan profile *customer* merupakan halaman yang digunakan *customer* untuk mengedit identitas, alamat email maupun mengubah *password*. Berikut adalah gambar dari implementasi halaman pengelolaan *profile customer* yang ditunjukkan pada Gambar 4.73. *Change password* merupakan fungsi yang digunakan *vendor* dan *customer* untuk mengubah *password* lama ke dengan *password* baru. Halaman *change password* ini berada di halaman *user profile*. Berikut adalah gambar dari implementasi halaman *change password* yang ditunjukkan pada Gambar 4.73.

Gambar 4. 73 Implementasi Halaman Pengelolaan *Profile Customer*

### C. Pencarian Vendor

Berikut merupakan implementasi halaman utama atau *home* pada *customer* yang menampilkan *list brand* yang ada yang dapat dicari berdasarkan kategori. Implementasi halaman *home* tersebut dapat dilihat pada Gambar 4.74.

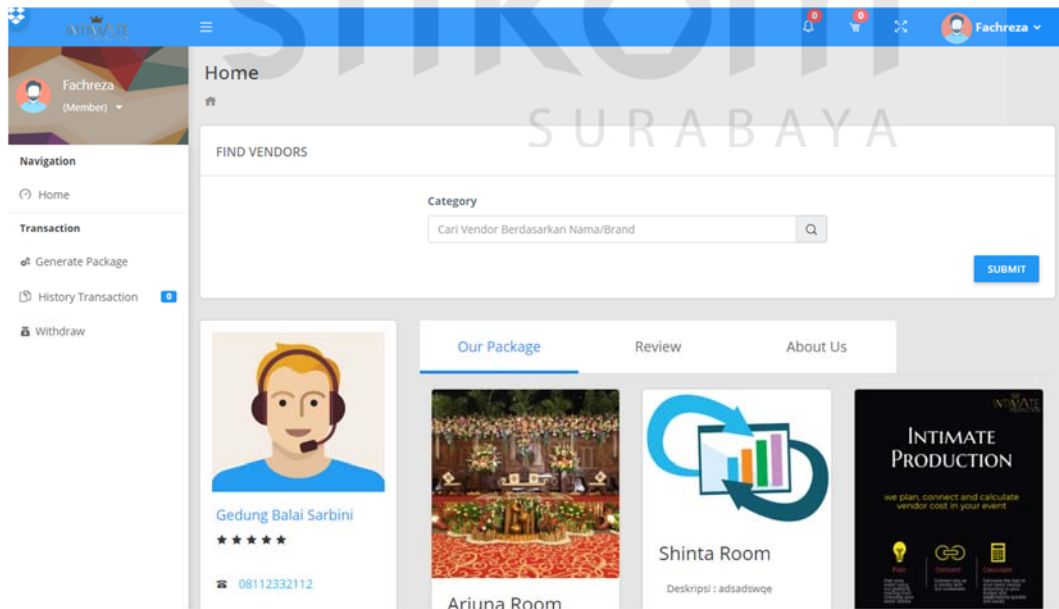


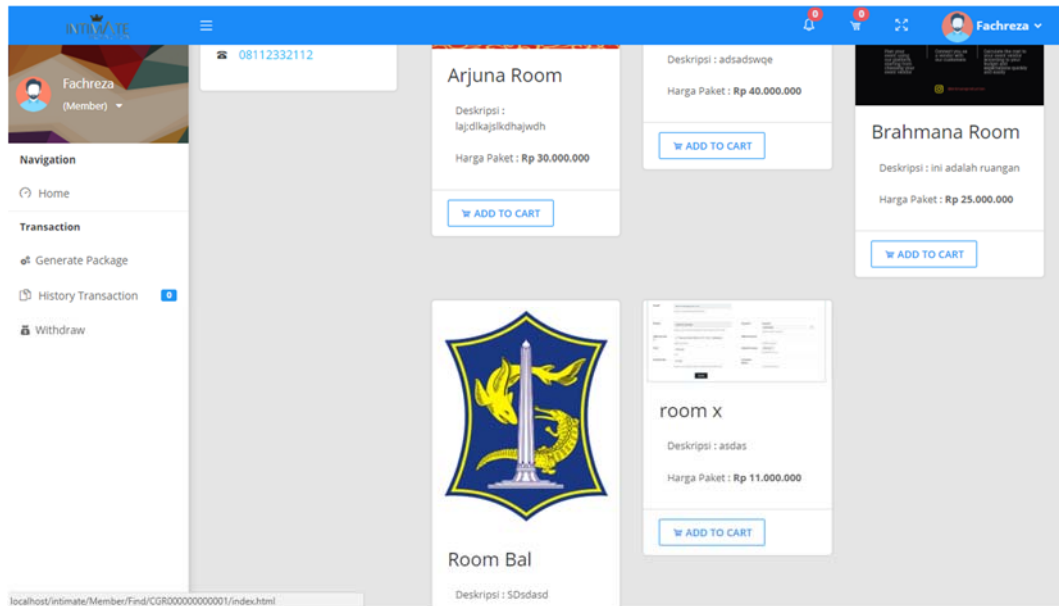


Gambar 4. 74 Implementasi Halaman Pencarian Vendor

#### D. *Customize Order*

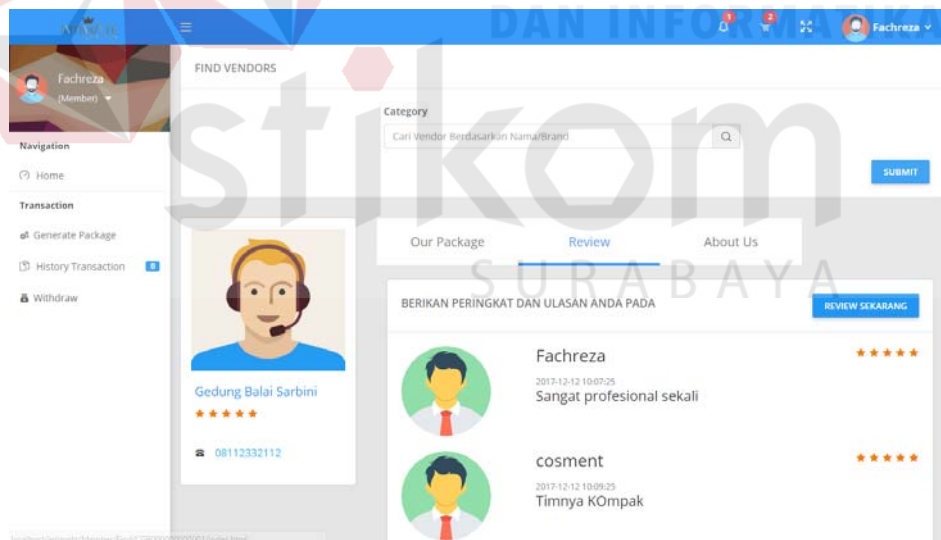
Berikut merupakan implementasi halaman *customize order* yang dilakukan *customer* untuk memilih paket vendor yang bukan direkomendasi oleh sistem. Halaman tersebut dapat dilihat pada Gambar 4.75.





Gambar 4. 75 Implementasi Halaman *Customize Order*

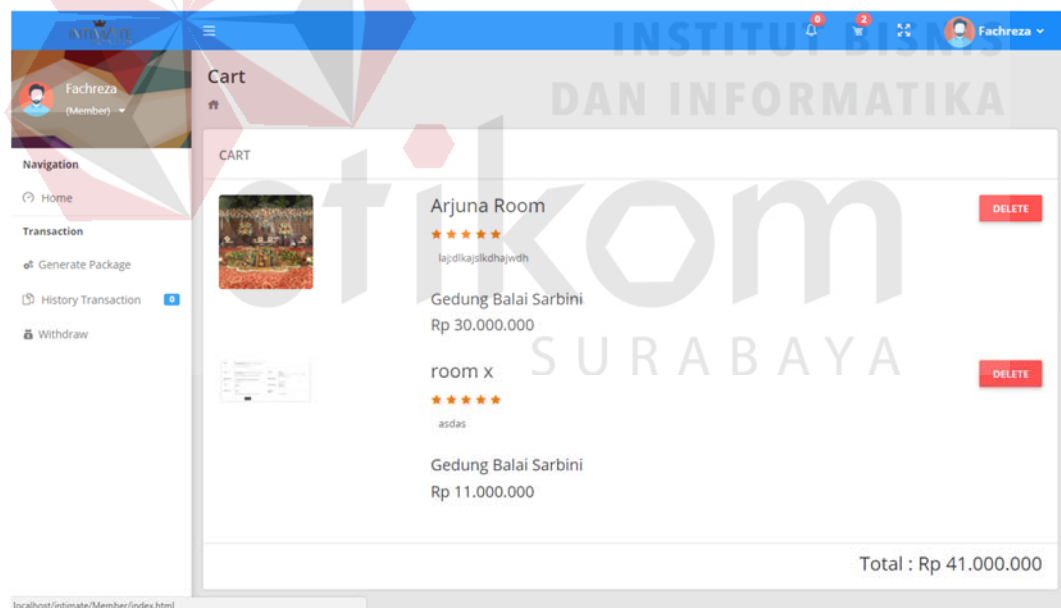
Berikut implementasi halaman *review* vendor yang dapat dilihat pada Gambar 4.76.



Gambar 4. 76 Implementasi Halaman *Review*

Gambar 4. 77 Implementasi Halaman Pemberian Rating/Ulasan

Berikut adalah implementasi halaman *cart* yang merupakan halaman keranjang belanja setelah *customer* melakukan *order*, maka akan masuk kedalam keranjang belanja/*cart*. Implementasi tersebut dapat dilihat pada Gambar 4.77 dan implementasi halaman *form* data pemesanan dapat dilihat pada Gambar 4.78.



Gambar 4. 78 Implementasi Halaman *Cart*

The screenshot displays the 'Form Data Order' interface. At the top, the user profile 'Fachreza (Member)' is visible. The main content area shows a transaction summary for 'Gedung Balai Sarbini' with a price of Rp 11.000.000. The total amount is Rp 41.000.000. Below this is a 'TRANSACTION DETAIL' section with form fields for 'Tanggal Acara', 'Jumlah Undangan' (set to 500), and 'Tema Acara' (set to Rustic / Gothic / Europe / Traditional). A green 'BAYAR' button is at the bottom.

Gambar 4. 79 Implementasi Halaman *Form Data Order*

### E. *Generate Paket dan Pemesanan Paket*

Implementasi halaman *generate paket* merupakan halaman yang digunakan oleh *customer* untuk menentukan vendor pernikahan. *Customer* menginputkan nominal *budget* yang diinginkan, kategori dan kota kemudian tekan tombol submit yang dapat dilihat pada Gambar 4.80.

The screenshot displays the 'Generate Package' interface. The page title is 'Generate Package' and the breadcrumb is 'Generate Package'. The main content area has a 'GENERATE' section with form fields for 'Your Budget Rp' (1.000.000), 'Category' (Select Category), and 'City' (Pilih). A blue 'SUBMIT' button is on the right. A large green 'GENERATE' button is at the bottom.

Gambar 4. 80 Implementasi Halaman *Generate Paket 1*



Setelah ditekan tombol submit maka akan muncul seperti Gambar 4.81 Kemudian *customer* memilih sub kriteria disetiap kategori yang telah dipilih lalu tekan tombol *generate*.

Gambar 4. 81 Implementasi Halaman Generate Paket 2

Setelah di *generate*, sistem akan menampilkan pilihan paket dan *budget* paket untuk dipilih *customer* yang dapat dilihat pada Gambar 4.82.

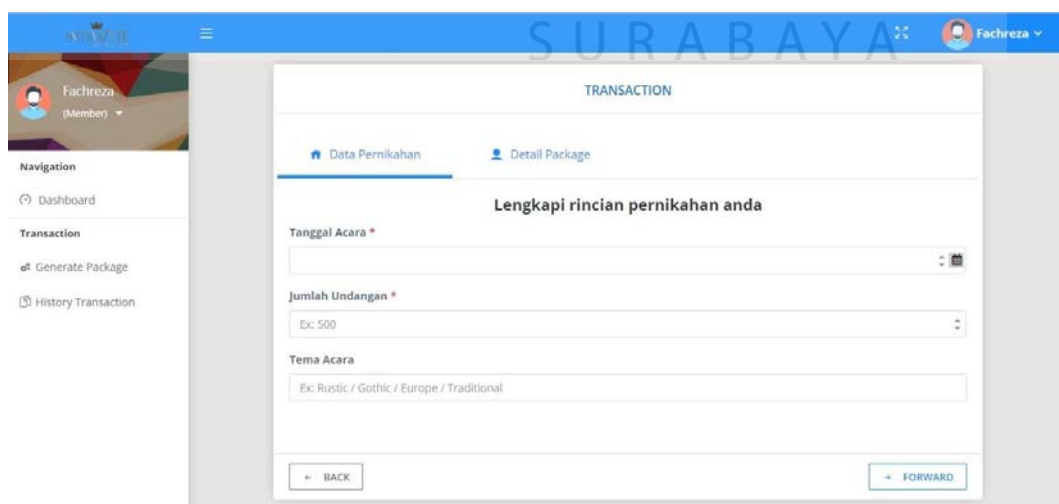
Gambar 4. 82 Implementasi Halaman Hasil *Generate* Paket

Berikut adalah contoh detail dari kriteria yang ada pada paket 1 yang dapat dilihat pada Gambar 4.83.



Gambar 4. 83 Implementasi Halaman Detail Hasil *Generate* Paket

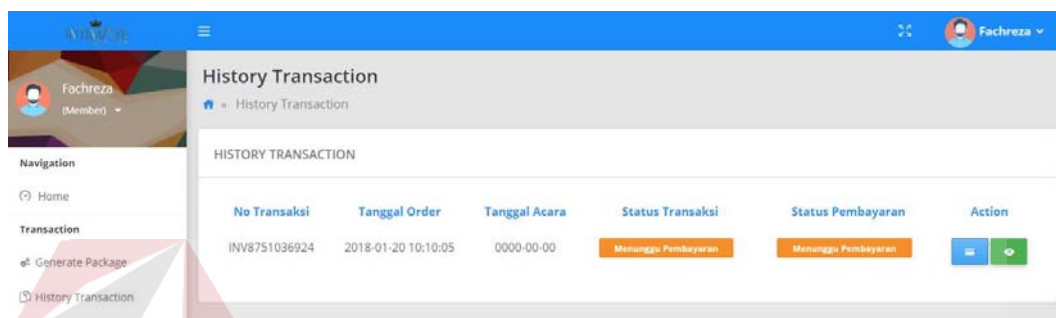
Setelah *customer* memilih paket, maka tahap selanjutnya yaitu pemesanan paket dengan menginputkan tanggal acara, jumlah undangan, dan tema acara. Kemudian *customer* menekan tombol *forward* untuk melanjutkan pemesanan.



Gambar 4. 84 Implementasi Halaman Pemesanan Paket Pada *Customer*

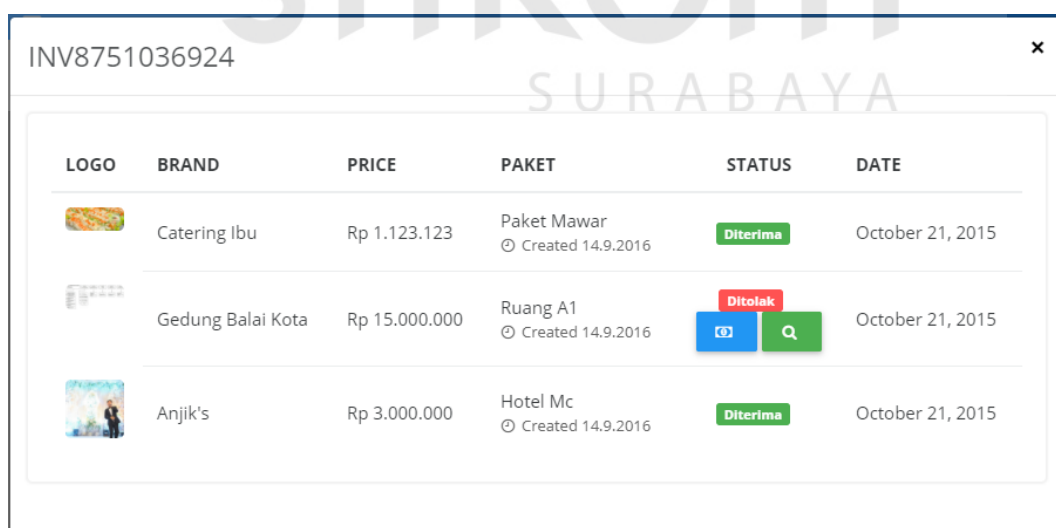
## F. Histori Transaksi

Halaman histori transaksi merupakan halaman untuk melihat riwayat pemesanan atau transaksi yang ada pada *customer*. Didalam riwayat pemesanan juga *customer* dapat melihat status dari transaksi dan pembayaran. Berikut implementasi halaman histori transaksi dapat dilihat pada Gambar 4.85.



Gambar 4. 85 Implementasi Halaman Histori Transaksi

Berikut adalah *pop up* detail transaksi dari aksi *view* yang dapat dilihat pada Gambar 4.86.



Gambar 4. 86 Implementasi Halaman Detail Pemesanan

## G. Konfirmasi Pembayaran

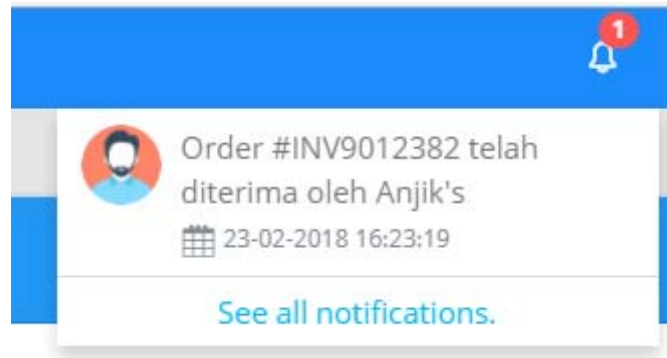
Implementasi halaman konfirmasi merupakan halaman konfirmasi pembayaran yang dilakukan oleh *customer* dengan menginputkan nama *bank*, nomor rekening, jumlah transfer, dan berita transfer. Berikut adalah gambar dari implementasi halaman konfirmasi pembayaran yang ditunjukkan pada Gambar 4.87.

The image shows a web interface for payment confirmation. It is divided into two main sections. The left section, titled 'TRANSFER PEMBAYARAN', displays the following information: 'No Invoice: IP9184730625INV', 'Jumlah Pembayaran: Rp. 5,000,123', 'Transfer Peris Hingga 3 Digit Terakhir', and 'Pembayaran Via Transfer Ke: Rekening DCA: 018 340 46 53'. The right section, titled 'KONFIRMASI PEMBAYARAN', contains a form with the following fields: 'Bank', 'Nomor Rekening: Masukkan Nomor Rekening Anda', 'Jumlah Transfer: Jumlah Uang yang Anda Transfer', 'Tanggal Transfer: [calendar icon]', and 'Berita Transfer: Berita Transfer (jika ada)'. Below the form is an 'Upload File' button and a blue 'KIRIM KONFIRMASI' button. A watermark for 'stikom SURABAYA' is visible across the bottom of the interface.

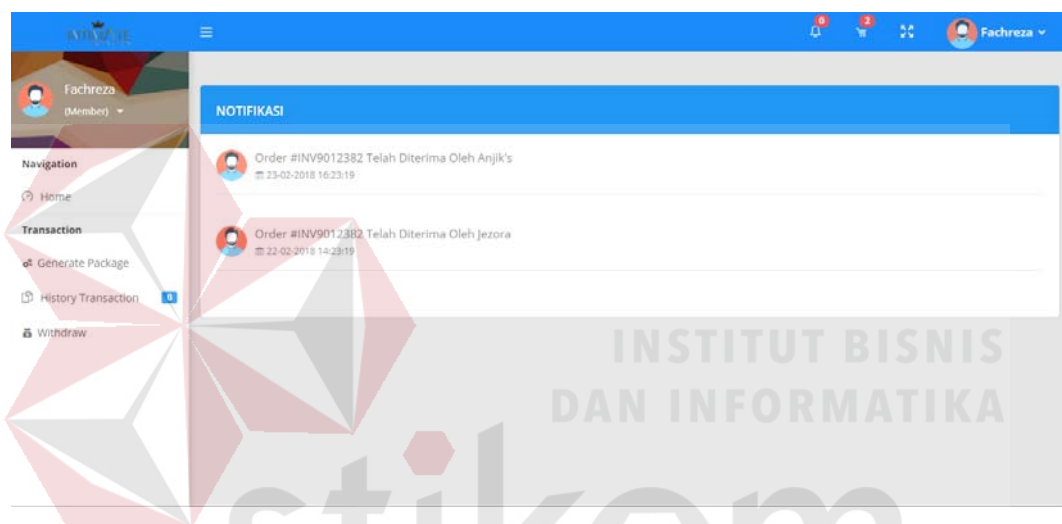
Gambar 4. 87 Implementasi Konfirmasi Pembayaran Pada *Customer*

### 4.1.7 Notification

Berikut merupakan implementasi notifikasi *order* pada *customer* yang terletak pada halaman disebelah kanan atas. Implementasi notifikasi tersebut dapat dilihat pada Gambar 4.88.



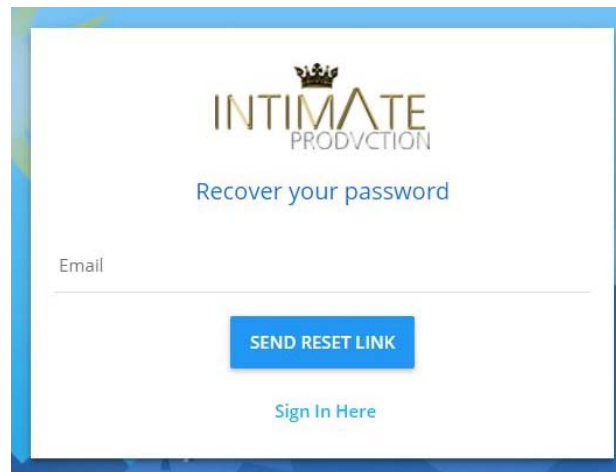
Gambar 4. 88 Notifikasi *Order* Pada *Customer*



Gambar 4. 89 Implementasi Halaman Notifikasi Pada *Customer*

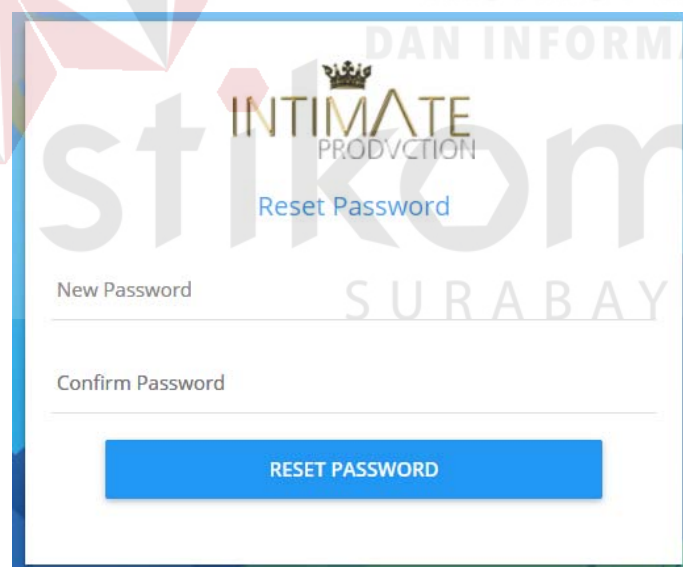
#### 4.1.8 *Forgot Password*

Halaman *forgot password* merupakan halaman untuk mengubah *password* jika *customer*, *vendor* atau admin lupa dengan *passwordnya* atau tidak dapat *login* untuk masuk kedalam aplikasi. *Customer*, *vendor*, atau admin menginputkan *email* kemudian menekan tombol *send reset link*. Berikut adalah gambar dari implementasi halaman *forgot password* yang ditunjukkan pada Gambar 4.89 dan Gambar 4.90.



Gambar 4. 90 Implementasi Halaman *Recover Password*

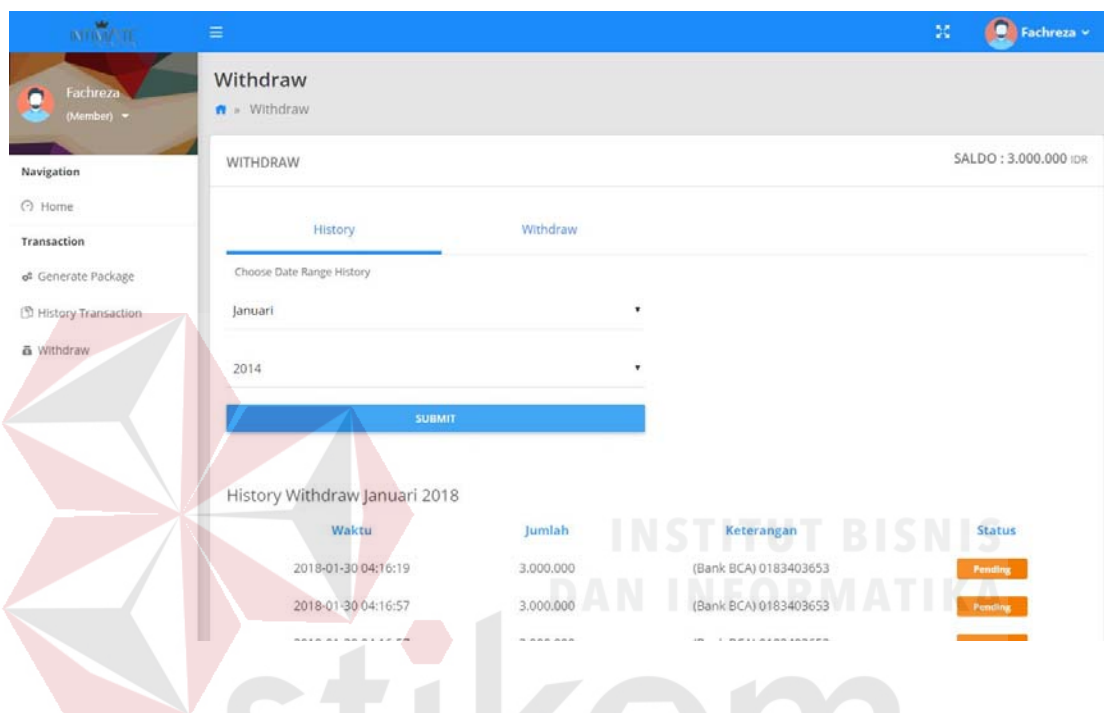
Setelah memasukkan *email* dan menekan tombol *send reset link*, selanjutnya *customer* atau *vendor* memasukkan *password* baru kemudian menekan tombol *reset password*.



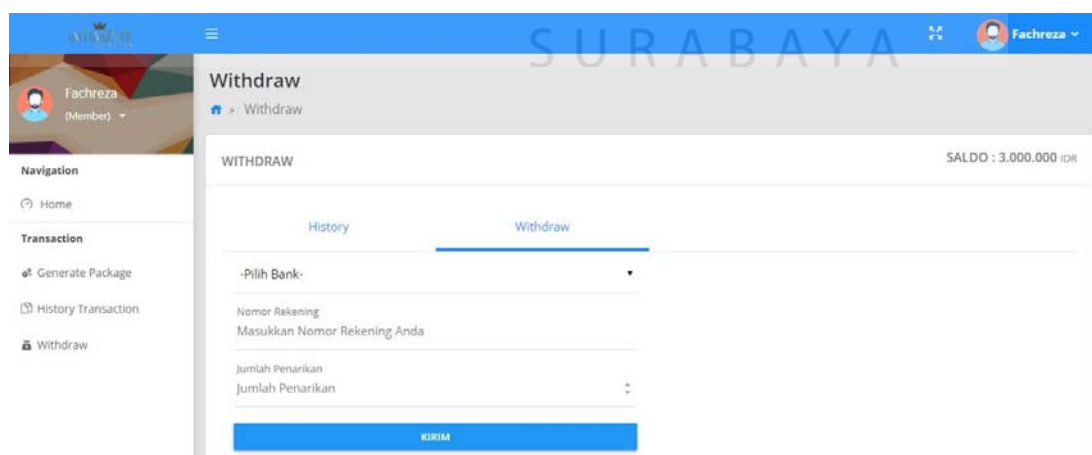
Gambar 4. 91 Implementasi Halaman *Reset Password*

#### 4.1.9 Withdraw

Berikut merupakan implementasi *withdraw* untuk penarikan saldo dan terdapat histori *withdraw* yang dilakukan oleh *customer* dan vendor yang dapat dilihat pada Gambar 4.92.



Gambar 4. 92 Implementasi Halaman Histori *Withdraw*



Gambar 4. 93 Implementasi Halaman *Withdraw*

## 4.2 Uji Coba dan Evaluasi Sistem

Setelah melakukan tahap implementasi, tahapan selanjutnya adalah melakukan proses uji coba dan evaluasi sistem. Tujuan dari melakukan uji coba dan evaluasi sistem adalah supaya mengetahui apakah sistem yang telah dibangun dapat berjalan dengan baik sesuai tujuan dan *output* yang diharapkan. Uji coba dan evaluasi sistem disesuaikan dengan perancangan yang telah dibuat pada bab 3 dan menggunakan *black box testing*. *Black Box Testing* yang dilakukan berfokus pada program untuk memenuhi kebutuhan (*requirement*) yang disebutkan dalam spesifikasi. Pengujian yang dilakukan dengan menjalankan setiap unit atau modul, lalu dilakukan pengamatan pada hasil dari unit atau modul apakah telah sesuai dengan proses bisnis yang diinginkan. Uji coba dan evaluasi dilakukan pada fungsi-fungsi yang ada pada sistem yang ditulis pada rancangan bab 3.

### 4.2.1 Uji Coba *Form Login*

Proses uji coba *login* dimulai dengan memasukkan *email* dan *password* yang telah didaftarkan. Proses *login* dinyatakan berhasil jika sistem telah masuk kehalaman utama. Hasil uji coba *login* dapat dilihat pada Tabel 4.1.

Tabel 4. 1 Hasil Uji Coba *Form Login*

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
1.	Melakukan <i>Login</i>	<i>Email</i> dan <i>Password</i>	<i>Login</i> berhasil dan masuk ke halaman sesuai <i>role</i>	Sukses (Gambar 4.1)
2.	Melakukan <i>Login</i> dengan <i>format</i> salah	<i>Email</i> dan <i>Password</i>	Muncul pesan “ <i>Email</i> atau <i>Password</i> Salah”	Sukses (Gambar 4.2)



## 4.2.2 Pengelolaan Pada *Role Admin*

### a. Hasil Uji Coba Pengelolaan Master Kategori

Proses uji coba pengelolaan master kategori dilakukan terhadap 4 (empat) *test case*. *Test case* tersebut yaitu menampilkan, menginputkan, mengubah, dan menghapus data kategori. Proses dinyatakan berhasil jika *output* sesuai dengan tujuan yang diharapkan. Hasil uji coba pengelolaan master kategori dapat dilihat pada Tabel 4.2.

Tabel 4. 2 Hasil Uji Coba *Form* Master Kategori

Test Case ID	Tujuan	<i>Input</i>	Output yang diharapkan	Status
3.	Menampilkan data kategori	Memilih menu “Manage Kategori”	Data Kategori	Sukses (Gambar 4.3)
4.	Menginputkan data kategori	Nama kategori	Data berhasil disimpan dan ditampilkan pada tabel <i>Manage Kategori</i>	Sukses (Gambar 4.4)
5.	Mengubah Data Kategori	Nama kategori	Data berhasil diubah dan ditampilkan pada tabel <i>Manage Kategori</i>	Sukses (Gambar 4.5)
6.	Menghapus Data Kategori	Memilih kategori yang akan di hapus	Muncul <i>pop up</i> konfirmasi hapus kategori dan jika di konfirmasi akan muncul notifikasi berhasil di hapus	Sukses (Gambar 4.6 & Gambar 4.7)

### b. Hasil Uji Coba Pengelolaan Master Nilai

Proses uji coba pengelolaan master nilai dilakukan terhadap 4 (empat) *test case*. *Test case* tersebut yaitu menampilkan, menginputkan, mengubah, dan

menghapus data nilai. Proses dinyatakan berhasil jika *output* sesuai dengan tujuan yang diharapkan. Hasil uji coba pengelolaan master nilai dapat dilihat pada Tabel 4.3.

Tabel 4. 3 Hasil Uji Coba *Form* Master Nilai

Test Case ID	Tujuan	<i>Input</i>	Output yang diharapkan	Status
7.	Menampilkan Data Nilai	Memilih menu “Manage Nilai”	Data Nilai	Sukses (Gambar 4.8)
8.	Menginputkan Data Nilai	Kriteria, sub kriteria, nilai batas bawah, nilai batas atas, nilai	Data berhasil disimpan dan ditampilkan pada tabel Manage Nilai	Sukses (Gambar 4.9)
9.	Mengubah Data Nilai	Kriteria, sub kriteria, nilai batas bawah, nilai batas atas, nilai	Data berhasil diubah dan ditampilkan pada tabel Manage Nilai	Sukses (Gambar 4.10)
10.	Menghapus Data Nilai	Memilih data Nilai yang akan di hapus	Muncul <i>pop up</i> konfirmasi hapus nilai dan jika di konfirmasi akan muncul notifikasi berhasil di hapus	Sukses (Gambar 4.11 & Gambar 4.12)

### c. Hasil Uji Coba Pengelolaan Sub Kriteria

Proses uji coba pengelolaan sub kriteria dilakukan terhadap 4 (empat) *test case*. *Test case* tersebut yaitu menampilkan, menginputkan, mengubah, dan menghapus data sub kriteria. Proses dinyatakan berhasil jika *output* sesuai dengan tujuan yang diharapkan. Hasil uji coba pengelolaan sub kriteria dapat dilihat pada Tabel 4.4.

Tabel 4. 4 Hasil Uji Coba *Form* Sub Kriteria

Test Case ID	Tujuan	<i>Input</i>	Output yang diharapkan	Status
11.	Menampilkan data sub kriteria	Memilih menu “Manage Sub Kriteria”	Data Sub Kriteria	Sukses (Gambar 4.13)
12.	Menginputkan data sub kriteria	Nama kriteria, nama sub kriteria	Data berhasil disimpan dan ditampilkan pada tabel Manage Sub Kriteria	Sukses (Gambar 4.14)
13.	Mengubah Data Sub Kriteria	Nama kriteria, nama sub kriteria	Data berhasil diubah dan ditampilkan pada tabel Manage Sub Kriteria	Sukses (Gambar 4.15)
14.	Menghapus Data Sub Kriteria	Memilih data Sub Kriteria yang akan di hapus	Muncul <i>pop up</i> konfirmasi hapus sub kriteria dan jika di konfirmasi akan muncul notifikasi berhasil di hapus	Sukses (Gambar 4.16 & Gambar 4.17)

#### d. Hasil Uji Coba Pengelolaan Master *Bank*

Proses uji coba pengelolaan master *bank* dilakukan terhadap 4 (empat) *test case*. *Test case* tersebut yaitu menampilkan, menginputkan, mengubah, dan menghapus data *bank*. Proses dinyatakan berhasil jika *output* sesuai dengan tujuan yang diharapkan. Hasil uji coba pengelolaan master *bank* dapat dilihat pada Tabel 4.5.

Tabel 4. 5 Hasil Uji Coba *Form* Master *Bank*

Test Case ID	Tujuan	<i>Input</i>	Output yang diharapkan	Status
15.	Menampilkan data bank	Memilih menu “Manage Bank”	Data Bank	Sukses (Gambar 4.18)

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
16.	Menginputkan data bank	Kode bank, Nama Bank	Data berhasil disimpan dan ditampilkan pada tabel Manage Bank	Sukses (Gambar 4.19)
17.	Mengubah Data Bank	Kode bank, Nama Bank	Data berhasil diubah dan ditampilkan pada tabel Manage Bank	Sukses (Gambar 4.20)
18.	Menghapus Data Bank	Memilih data Bank yang akan di hapus	Muncul <i>pop up</i> konfirmasi hapus Bank dan jika di konfirmasi akan muncul notifikasi berhasil di hapus	Sukses (Gambar 4.21 & Gambar 4.22)

#### e. Hasil Uji Coba Konfirmasi Pembayaran

Proses uji coba konfirmasi pembayaran dilakukan terhadap 4 (empat) *test case*. *Test case* tersebut yaitu menampilkan *list* pembayaran, menampilkan bukti transfer, melakukan konfirmasi penerimaan pembayaran, dan melakukan *reject* penerimaan pembayaran. Proses dinyatakan berhasil jika *output* sesuai dengan tujuan yang diharapkan. Hasil uji coba konfirmasi pembayaran dapat dilihat pada Tabel 4.6.

Tabel 4. 6 Hasil Uji Coba *Form* Konfirmasi Pembayaran

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
19.	Menampilkan <i>list</i> pembayaran yang akan dikonfirmasi	Memilih menu “Confirmation Payment”	Data <i>Confirmation Payment</i>	Sukses (Gambar 4.23)
20.	Menampilkan bukti transfer	Menekan ikon gambar bukti transfer	Menampilkan gambar bukti transfer secara utuh	Sukses (Gambar 4.24)

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
21.	Melakukan Konfirmasi Penerimaan Pembayaran	Menekan tombol konfirmasi	List bukti transfer yang telah dikonfirmasi menghilang dan jika di konfirmasi akan muncul notifikasi berhasil di konfirmasi	Sukses (Gambar 4.25, Gambar 4.6)
22.	Melakukan <i>Reject</i> Penerimaan Pembayaran	Menekan tombol <i>reject</i>	List bukti transfer yang telah <i>direject</i> menghilang dan jika di konfirmasi akan muncul notifikasi berhasil di <i>reject</i>	Sukses (Gambar 4.27, Gambar 4.28)

#### f. Hasil Uji Coba *History Confirmation*

Proses uji coba *history confirmation* dilakukan terhadap 1 (satu) *test case*. *Test case* tersebut yaitu menampilkan data *history confirmation*. Proses dinyatakan berhasil jika *output* sesuai dengan tujuan yang diharapkan. Hasil uji coba *history confirmation* dapat dilihat pada Tabel 4.7.

Tabel 4. 7 Hasil Uji Coba *History Confirmation*

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
23.	Menampilkan Data <i>History Confirmation</i>	Memilih menu “ <i>History Confirmation</i> ”	Data <i>History Confirmation</i>	Sukses (Gambar 4.29)

#### g. Hasil Uji Coba *Legality Verification*

Proses uji coba *legality verification* dilakukan terhadap 5 (lima) *test case*. *Test case* tersebut yaitu menampilkan data vendor yang melakukan verifikasi, menampilkan *scan*/gambar NPWP, KTP, melakukan konfirmasi *legality*

*verification*, dan melakukan *reject legality verification*. Proses dinyatakan berhasil jika *output* sesuai dengan tujuan yang diharapkan. Hasil uji coba *legality verification* dapat dilihat pada Tabel 4.8.

Tabel 4. 8 Hasil Uji Coba *Legality Verification*

Test Case ID	Tujuan	<i>Input</i>	Output yang diharapkan	Status
24.	Menampilkan Data Vendor yang melakukan verifikasi	Memilih menu “ <i>Legality Verification</i> ”	Data <i>Legality Vendor</i>	Sukses (Gambar 4.30)
25.	Menampilkan Scan / gambar NPWP	Menekan ikon gambar NPWP	Menampilkan gambar NPWP secara utuh	Sukses (Gambar 4.31)
26.	Menampilkan Scan / gambar KTP	Menekan ikon gambar KTP	Menampilkan gambar KTP	Sukses (Gambar 4.32)
27.	Melakukan Konfirmasi <i>Legality Verification</i>	Menekan tombol konfirmasi	List <i>legality vendor</i> yang telah dikonfirmasi menghilang dan jika di konfirmasi akan muncul notifikasi berhasil di konfirmasi	Sukses (Gambar 4.33 & Gambar 4.34)
28.	Melakukan <i>Reject Legality Verification</i>	Menekan tombol <i>reject</i>	List <i>legality vendor</i> yang telah direject menghilang dan jika di konfirmasi akan muncul notifikasi berhasil di <i>reject</i>	Sukses (Gambar 4.35 & Gambar 4.36)

#### h. Hasil Uji Coba *Dashboard Admin*

Proses uji coba *dashboard admin* dilakukan terhadap 1 (satu) *test case*. *Test case* tersebut yaitu menampilkan data *dashboard*. Proses dinyatakan berhasil

jika *output* sesuai dengan tujuan yang diharapkan. Hasil uji coba *dashboard* dapat dilihat pada Tabel 4.9.

Tabel 4. 9 Hasil Uji Coba *Dashboard*

Test Case ID	Tujuan	<i>Input</i>	Output yang diharapkan	Status
29.	Menampilkan Data Dashboard	Memilih menu “Dashboard”	Data <i>Dashboard</i> yang ditampilkan dapat meng- <i>update</i> sesuai dengan adanya transaksi	Sukses (Gambar 4.37)

#### i. Hasil Uji Coba Laporan *Customer* pada Admin

Proses uji coba laporan dilakukan terhadap laporan-laporan yang dihasilkan oleh sistem yaitu laporan *customer*. Uji coba laporan dinyatakan berhasil jika laporan yang dihasilkan sesuai dengan parameter yang dimasukkan kedalam sistem. Hasil uji coba laporan *customer* dapat dilihat pada Tabel 4.10.

Tabel 4. 10 Hasil Uji Coba Laporan *Customer* pada Admin

Test Case ID	Tujuan	<i>Input</i>	Output yang diharapkan	Status
30.	Menampilkan <i>pop up</i> periode laporan <i>customer</i>	Tekan tombol cetak pada <i>card box customer</i>	Pop up Laporan dengan periode	Sukses (Gambar 4.38)
31.	Menampilkan dokumen cetak laporan <i>customer</i>	Memilih periode laporan dan tekan tombol lihat laporan	Dokumen cetak berisi laporan sesuai periode yang dipilih	Sukses (Gambar 4.39)

#### j. Hasil Uji Coba Laporan Vendor pada Admin

Proses uji coba laporan dilakukan terhadap laporan-laporan yang dihasilkan oleh sistem yaitu laporan vendor. Uji coba laporan dinyatakan berhasil

jika laporan yang dihasilkan sesuai dengan parameter yang dimasukkan kedalam sistem. Hasil uji coba laporan vendor dapat dilihat pada Tabel 4.11.

Tabel 4. 11 Hasil Uji Coba Laporan Vendor pada Admin

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
32.	Menampilkan <i>pop up</i> periode laporan <i>vendor</i>	Tekan tombol cetak pada <i>card box vendor</i>	Pop up Laporan dengan periode	Sukses (Gambar 4.40)
33.	Menampilkan dokumen cetak laporan <i>vendor</i>	Memilih periode laporan dan tekan tombol lihat laporan	Dokumen cetak berisi laporan sesuai periode yang dipilih	Sukses (Gambar 4.41)

#### k. Hasil Uji Coba Laporan Transaksi pada Admin

Proses uji coba laporan dilakukan terhadap laporan-laporan yang dihasilkan oleh sistem yaitu laporan transaksi. Uji coba laporan dinyatakan berhasil jika laporan yang dihasilkan sesuai dengan parameter yang dimasukkan kedalam sistem. Hasil uji coba laporan transaksi dapat dilihat pada Tabel 4.12.

Tabel 4. 12 Hasil Uji Coba Laporan Transaksi pada Admin

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
34.	Menampilkan <i>pop up</i> periode laporan transaksi	Tekan tombol cetak pada <i>card box customer</i>	<i>Pop up</i> Laporan dengan periode	Sukses (Gambar 4.42)
35.	Menampilkan dokumen cetak laporan transaksi	Memilih periode laporan dan tekan tombol lihat laporan	Dokumen cetak berisi laporan sesuai periode yang dipilih	Sukses (Gambar 4.43)

#### l. Hasil Uji Coba *Withdraw* / Penarikan

Proses uji coba *withdraw*/penarikan dilakukan untuk penarikan saldo. Uji coba ini terdiri dari 3 (tiga) *test case* yaitu menampilkan list penarikan yang akan



dikonfirmasi, melakukan konfirmasi penarikan, dan melakukan *reject* penarikan. Uji coba dinyatakan berhasil jika *output* sesuai dengan tujuan yang diharapkan. Hasil uji coba *withdraw* dapat dilihat pada Tabel 4.13.

Tabel 4.13 Hasil Uji Coba *Withdraw* Pada Admin

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
36.	Menampilkan list penarikan yang akan dikonfirmasi	Memilih menu “ <i>Confirmation Withdraw</i> ”	Data <i>Confirmation Withdraw</i>	Sukses (Gambar 4.44)
37.	Melakukan Konfirmasi Penarikan	Menekan tombol konfirmasi	List penarikan yang telah dikonfirmasi menghilang dan jika di konfirmasi akan muncul notifikasi berhasil di konfirmasi	Sukses (Gambar 4.45 & Gambar 4.46)
38.	Melakukan <i>Reject</i> Penarikan	Menekan tombol <i>reject</i>	List penarikan yang telah direject menghilang dan jika di konfirmasi akan muncul notifikasi berhasil di <i>reject</i>	Sukses (Gambar 4.47 & Gambar 4.48)

### 4.2.3 Pengelolaan Pada *Role Vendor*

#### a. Hasil Uji Coba Registrasi Vendor

Proses uji coba registrasi vendor dimulai dengan memasukkan data diri dan *email*. Setelah itu sistem akan menyimpan data pengguna untuk kemudian pengguna dapat melakukan *login* kedalam sistem. Hasil uji coba registrasi vendor dapat dilihat pada Tabel 4.14.

Tabel 4. 14 Hasil Uji Coba *Form Registrasi Vendor*

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
39.	Registrasi Vendor	<i>First Name, Last Name, Email, Password, Confirm Password</i>	Registrasi Vendor Berhasil	Sukses (Gambar 4.49)

#### b. Hasil Uji Coba *Legality Vendor*

Proses uji coba *legality vendor* dilakukan untuk menginputkan data pengguna dan mengupload lampiran. Uji coba ini terdiri dari 1 (satu) *test case* yaitu melakukan registrasi legalitas vendor. Uji coba dinyatakan berhasil jika *output* sesuai dengan tujuan yang diharapkan. Hasil uji coba *legality vendor* dapat dilihat pada Tabel 4.15.

Tabel 4. 15 Hasil Uji Coba *Form Legality Vendor*

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
40.	Registrasi Legalitas Vendor	<i>Form data pimpinan dan Form legalitas perusahaan</i>	Notifikasi “Mohon menunggu data anda akan diverifikasi oleh admin”	Sukses (Gambar 4.50)

#### c. Hasil Uji Coba *Dashboard Vendor*

Proses uji coba *dashboard vendor* dilakukan untuk mengetahui apakah informasi fungsi data/grafik dapat ditampilkan dan memberi informasi yang *up to date*. *test case* yang dilakukan terdiri dari 1 (satu) *test case* yaitu menampilkan data *dashboard*. Uji coba dinyatakan berhasil jika *output* sesuai dengan tujuan yang diharapkan. Hasil uji coba *dashboard vendor* dapat dilihat pada Gambar 4.16.

Tabel 4. 16 Hasil Uji Coba *Dashboard*

Test Case ID	Tujuan	<i>Input</i>	Output yang diharapkan	Status
41.	Menampilkan Data Dashboard	Memilih menu “Dashboard”	Data <i>Dashboard</i> yang ditampilkan dapat meng- <i>update</i> sesuai dengan adanya transaksi	Sukses (Gambar 4.51)

#### d. Hasil Uji Coba Pengelolaan Data Master Kategori Vendor

Proses uji coba pengelolaan master kategori vendor dilakukan terhadap 4 (empat) *test case*. *Test case* tersebut yaitu menampilkan data kategori vendor, menginputkan data kategori vendor, menghapus kategori vendor, dan menampilkan halaman *view package*. Proses dinyatakan berhasil jika *output* sesuai dengan tujuan yang diharapkan. Hasil uji coba pengelolaan master kategori vendor dapat dilihat pada Tabel 4.17.

Tabel 4. 17 Hasil Uji Coba *Form* Master Kategori Vendor

Test Case ID	Tujuan	<i>Input</i>	Output yang diharapkan	Status
42.	Menampilkan Data Kategori <i>Vendor</i>	Memilih menu “Category <i>Vendor</i> ”	Data Kategori <i>Vendor</i>	Sukses (Gambar 4.53 & Gambar 4.54)
43.	Menginputkan Data Kategori <i>Vendor</i>	Nama Brand, Jenis Kategori, Alamat Perusahaan, Nomor Telpon	Data berhasil disimpan dan ditampilkan pada card <i>Our Category Business</i>	Sukses (Gambar 4.55)
44.	Menghapus Kategori <i>Vendor</i>	Menekan tombol hapus kategori bisnis	Muncul pop up untuk konfirmasi hapus, jika tekan ok data akan dihapus jika	Sukses (Gambar 4.56 & Gambar 4.57)

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
			cancel data tidak terhapus	
45.	Menampilkan halaman <i>View Package</i>	Menekan tombol <i>view package</i>	Menampilkan halaman <i>view package</i>	Sukses (Gambar 4.58 & Gambar 4.59)

#### e. Hasil Uji Coba Pengelolaan Data Master Paket Vendor

Proses uji coba pengelolaan master paket vendor dilakukan terhadap 3 (tiga) *test case*. *Test case* tersebut yaitu menginputkan data paket *vendor*, mengubah data paket *vendor*, dan menghapus *package*. Proses dinyatakan berhasil jika *output* sesuai dengan tujuan yang diharapkan. Hasil uji coba pengelolaan master paket vendor dapat dilihat pada Tabel 4.18.

Tabel 4. 18 Hasil Uji Coba *Form* Master Paket Vendor

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
46.	Menginputkan Data Paket <i>Vendor</i>	Nama Paket, Harga Paket, Deskripsi Paket, Unggah Gambar	Data berhasil disimpan dan ditampilkan pada card <i>Our Package</i>	Sukses (Gambar 4.60)
47.	Mengubah Data Paket <i>Vendor</i>	Nama Paket, Harga Paket, Deskripsi Paket, Unggah Gambar	Data berhasil diubah dan ditampilkan pada <i>card Our Package</i>	Sukses (Gambar 4.61)
48.	Menghapus <i>Package</i>	Menekan tombol hapus <i>package</i>	Muncul pop up untuk konfirmasi hapus, jika tekan ok data akan dihapus jika cancel data tidak terhapus	Sukses (Gambar 4.62 & Gambar 4.63)

#### f. Hasil Uji Coba Pengelolaan *Profile Vendor*

Proses uji coba data profil vendor dilakukan terhadap 2 (dua) *test case*. *Test case* tersebut yaitu menampilkan *user profile* dan mengubah *profile*. Uji coba dinyatakan berhasil jika *output* sesuai dengan tujuan yang diharapkan. Hasil uji coba pengelolaan *profile* vendor dapat dilihat pada Tabel 4.19.

Tabel 4. 19 Hasil Uji Coba *Form Profile Vendor*

Test Case ID	Tujuan	<i>Input</i>	Output yang diharapkan	Status
49.	Menampilkan <i>User Profile</i>	Memilih menu “ <i>View Profile</i> ”	Data <i>User Profile</i>	Sukses (Gambar 4.64)
50.	Mengubah <i>Profile</i>	<i>Input</i> data profile	Data profile telah diperbarui	Sukses (Gambar 4.64)

#### g. Hasil Uji Coba Konfirmasi *Order*

Proses uji coba konfirmasi *order* terdiri dari 3 (tiga) *test case* yaitu menampilkan daftar *order* baru, melakukan konfirmasi *order*, dan melakukan *reject order*. Uji coba dinyatakan berhasil jika *output* sesuai dengan tujuan yang diharapkan. Hasil uji coba konfirmasi *order* dapat dilihat pada Tabel 4.20.

Tabel 4. 20 Hasil Uji Coba *Form Konfirmasi Order*

Test Case ID	Tujuan	<i>Input</i>	Output yang diharapkan	Status
51.	Menampilkan Daftar <i>Order</i> Baru	Memilih menu “ <i>Incoming Order</i> ”	Data <i>Order</i> Baru	Sukses (Gambar 4.65)
52.	Melakukan Konfirmasi <i>Order</i>	Menekan tombol konfirmasi	List <i>order</i> yang telah dikonfirmasi menghilang dan jika dikonfirmasi akan muncul notifikasi	Sukses (Gambar 4.66 & Gambar 4.67)

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
			berhasil di konfirmasi	
53.	Melakukan <i>Reject Order</i>	Menekan tombol <i>reject</i>	List <i>order</i> yang telah direject menghilang dan jika di konfirmasi akan muncul notifikasi berhasil di <i>reject</i>	Sukses (Gambar 4.68 & Gambar 4.69)

#### h. Hasil Uji Coba Jadwal Vendor

Proses uji coba jadwal vendor dilakukan dengan 1 (satu) *test case* yaitu menampilkan *pop up form* tambah jadwal vendor. Uji coba dinyatakan berhasil jika sistem berhasil menyimpan data yang telah diinputkan. Hasil uji coba jadwal vendor dapat dilihat pada Tabel 4.21.

Tabel 4. 21 Hasil Uji Coba *Form* Jadwal Vendor

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
54.	Menampilkan <i>pop up form</i> tambah jadwal vendor	Pilih tombol jadwal	Menampilkan <i>pop up form</i> jadwal	Sukses (Gambar 4.70)

#### 4.2.4 Pengelolaan Pada *Role Customer*

##### h. Hasil Uji Coba Registrasi *Customer*

Proses uji coba registrasi *customer* dimulai dengan memasukkan data diri dan *email*. Setelah itu sistem akan menyimpan data pengguna untuk kemudian pengguna dapat melakukan *login* kedalam sistem. Hasil uji coba registrasi *customer* dapat dilihat pada Tabel 4.22.

Tabel 4. 22 Hasil Uji Coba *Form Registrasi Customer*

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
55.	Registrasi <i>Customer</i>	<i>Email, Password, Confirm Password</i>	Registrasi <i>Customer</i> Berhasil	Sukses (Gambar 4.71 & Gambar 4.72)

#### i. Hasil Uji Coba Pengelolaan *Profile Customer*

Proses uji coba data profil *customer* dilakukan terhadap 2 (dua) *test case*. *Test case* tersebut yaitu menampilkan *user profile* dan mengubah *profile*. Uji coba dinyatakan berhasil jika *output* sesuai dengan tujuan yang diharapkan. Hasil uji coba pengelolaan *profile customer* dapat dilihat pada Tabel 4.23.

Tabel 4. 23 Hasil Uji Coba *Form Profile Customer*

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
56.	Menampilkan <i>User Profile</i>	Memilih menu "View Profile"	Data <i>User Profile</i>	Sukses (Gambar 4.73)
57.	Mengubah <i>Profile</i>	Input data profile	Data profile telah diperbarui	Sukses (Gambar 4.73)

#### j. Hasil Uji Coba Pencarian Vendor

Proses uji coba pencarian vendor dilakukan untuk mencari dan menampilkan daftar *brand*. Uji coba dinyatakan berhasil apabila data yang dicari dapat ditampilkan oleh sistem. Hasil uji coba pencarian vendor dapat dilihat pada Tabel 4.24.

Tabel 4. 24 Hasil Uji Coba Pencarian Vendor

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
58.	Menampilkan Daftar <i>Brand</i>	Input data nama <i>brand</i>	Data <i>brand</i> sesuai <i>input</i>	Sukses (Gambar 4.74)

#### k. Hasil Uji Coba *Customize Order*

Proses uji coba *customize order* terdiri dari 4 (empat) *test case* yaitu menampilkan profile *brand* (*our package*), menampilkan profile *brand* (*review*), menampilkan form pemberian *rating* dan *review*, dan menampilkan *cart* dan *form* transaksi. Uji coba dinyatakan berhasil jika *output* sesuai dengan tujuan yang diharapkan. Hasil uji coba *customize order* dapat dilihat pada Tabel 4.25.

Tabel 4. 25 Hasil Uji Coba *Customize Order*

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
59.	Menampilkan Profile <i>Brand</i> ( <i>Our package</i> )	Tekan tombol lihat <i>profile</i>	Menampilkan profile <i>brand</i> dan paket yang dimiliki	Sukses (Gambar 4.75)
60.	Menampilkan Profile <i>Brand</i> ( <i>Review</i> )	Tekan tab lihat <i>profile</i>	Menampilkan profile <i>brand</i> dan semua ulasan	Sukses (Gambar 4.76)
61.	Menampilkan form pemberian <i>rating</i> dan <i>review</i>	Tekan tombol <i>review</i> sekarang	Menampilkan <i>pop up form rating and review</i>	Sukses (Gambar 4.77)
62.	Menampilkan <i>cart</i> dan <i>form</i> transaksi	Tekan tombol <i>add to cart</i> pada paket yang dipilih dan tekan tombol keranjang pada bagian atas	Menampilkan daftar paket yang akan dibeli pada <i>cart</i> dan menampilkan form transaksi	Sukses (Gambar 4.78 & Gambar 4.79)



### l. Hasil Uji Coba *Generate Paket dan Pemesanan Paket*

Proses uji coba *generate* paket dan pemesanan paket dinyatakan berhasil apabila sistem dapat memberikan beberapa rekomendasi paket pernikahan sesuai dengan *budget*, kriteria, *rating*, dan pengalaman. Hasil uji coba dapat dilihat pada Tabel 4.26.

Tabel 4. 26 Hasil Uji Coba *Form Generate Paket dan Pemesanan Paket*

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
63.	Menampilkan <i>Generate Paket</i>	Memilih menu “ <i>Generate Package</i> ”	Menampilkan <i>Form Generate Package</i>	Sukses (Gambar 4.80)
64.	Menentukan Kategori yang dibutuhkan dan menentukan prioritas kategori dan sub kriteria	<i>Budget, Category, tanggal acara, kota</i>	list pilihan kategori dan sub kriteria	Sukses (Gambar 4.81)
65.	Membuat pilihan paket pernikahan	Menekan tombol <i>generate</i>	Daftar paket dengan berbagai <i>category vendor</i> , pilihan paket dari <i>vendor</i>	Sukses (Gambar 4.82)
66.	Menampilkan <i>pop up</i> dari detail paket	Tekan tombol (i)	Menampilkan <i>pop up</i> detail paket	Sukses (Gambar 4.83)
67.	Menampilkan <i>form</i> pemesanan	Menekan tombol <i>choose</i>	Menampilkan <i>form</i> pemesanan	Sukses (Gambar 4.84)

### m. Hasil Uji Coba *Histori Transaksi (Pemesanan)*

Proses uji coba histori transaksi dilakukan terhadap 2 (dua) *test case*. *Test case* tersebut yaitu menampilkan histori transaksi dan melihat detail transaksi. Proses dinyatakan berhasil jika *output* sesuai dengan tujuan yang diharapkan. Hasil uji coba histori transaksi dapat dilihat pada Tabel 4.27.

Tabel 4. 27 Hasil Uji Coba *Form* Histori Transaksi

Test Case ID	Tujuan	<i>Input</i>	Output yang diharapkan	Status
68.	Menampilkan Histori Transaksi	Memilih menu “ <i>History Transaction</i> ”	Data <i>history transaction</i>	Sukses (Gambar 4.85)
69.	Melihat Detail Transaksi	Menekan tombol Lihat Detail	Menampilkan pop up daftar <i>category vendor</i>	Sukses (Gambar 4.86)

#### n. Hasil Uji Coba Konfirmasi Pembayaran

Proses uji coba konfirmasi pembayaran dilakukan terhadap 1 (satu) *test case*. *Test case* tersebut yaitu menampilkan konfirmasi pembayaran. Proses dinyatakan berhasil jika *output* sesuai dengan tujuan yang diharapkan. Hasil uji coba konfirmasi pembayaran dapat dilihat pada Tabel 4.28.

Tabel 4. 28 Hasil Uji Coba *Form* Konfirmasi Pembayaran Pada *Customer*

Test Case ID	Tujuan	<i>Input</i>	Output yang diharapkan	Status
70.	Menampilkan konfirmasi pembayaran	Bank, Nomor Rekening, Jumlah Transfer, Tanggal Transfer, Unggah Gambar	Menampilkan pesan “Konfirmasi pembayaran berhasil”	Sukses (Gambar 4.87)

#### 4.2.5 Hasil Uji Coba Notifikasi

Proses uji coba notifikasi dinyatakan berhasil apabila sistem dapat menampilkan pemberitahuan atau *update* status. Hasil uji coba notifikasi dapat dilihat pada Tabel 4.29.

Tabel 4. 29 Hasil Uji Coba Notifikasi

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
71.	Menampilkan notifikasi secara ringkas pada tombol notifikasi	Tekan tombol notifikasi	Notifikasi terbaru pada tombol notifikasi	Sukses (Gambar 4.88)
72.	Menampilkan notifikasi secara keseluruhan	Tekan tombol <i>see all notications</i>	Seluruh list notifikasi	Sukses (Gambar 4.89)

#### 4.2.6 Hasil Uji Coba *Forgot Password*

Proses uji coba *forgot password* dinyatakan berhasil apabila dapat melakukan *update password*. Hasil uji coba *forgot password* dapat dilihat pada Tabel 4.30.

Tabel 4. 30 Hasil Uji Coba *Form Forgot Password*

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
73.	Membuat <i>link reset password</i>	<i>Email</i>	<i>Link reset password</i> berhasil dikirim ke email anda	Sukses (Gambar 4.90)
74.	Membuat <i>Password Baru</i>	<i>New Passwpord, Confirm Password</i>	Menampilkan pesan " <i>Password</i> berhasil dirubah"	Sukses (Gambar 4.91)

#### 4.2.7 Hasil Uji Coba *Withdraw*

Proses uji coba *withdraw* dinyatakan berhasil apabila dapat menampilkan *list history withdraw* dan *form withdraw*. Hasil uji coba *withdraw* dapat dilihat pada Tabel 4.31.

Tabel 4. 31 Hasil Uji Coba *Withdraw*

Test Case ID	Tujuan	Input	Output yang diharapkan	Status
75.	Menampilkan <i>history withdraw</i>	Tekan tombol <i>withdraw</i>	Menampilkan list <i>history withdraw</i>	Sukses (Gambar 4.92)
76.	Menampilkan <i>form withdraw</i>	Tekan tab <i>withdraw</i>	Menampilkan <i>form withdraw</i>	Sukses (Gambar 4.93)

#### 4.2.8 Evaluasi Sistem

Pada hasil uji coba aplikasi Rancang Bangun Aplikasi Penentuan Vendor Pernikahan Untuk Usaha Wedding Organizer Menggunakan Metode SMARTER Berbasis Web, didapatkan hasil bahwa :

1. Aplikasi mampu memberikan rekomendasi paket pernikahan sesuai dengan kriteria yang dibutuhkan oleh *customer* dari vendor yang terbaik. Kriteria tersebut meliputi *budget customer*, *rating*, dan pengalaman. Misalnya jika *customer* memiliki *budget* Rp. 30.000.000, maka rekomendasi paket yang dihasilkan tidak akan lebih dari *budget* tersebut. Hal ini sudah dibuktikan pada *Test Case ID* 63-67.
2. Aplikasi dapat membantu *customer* untuk membuat paket pernikahan secara customisasi (memilih satu persatu sesuai keinginan) dari tiap kategori vendor yang ada tanpa harus menghitung kembali biaya paket satu persatu karena sistem akan menghitung secara otomatis.
3. Aplikasi dapat menjadi wadah untuk mengembangkan potensi vendor dengan cara menawarkan paket yang dimiliki vendor kepada *customer* dengan melakukan *generate* paket sesuai kriteria *customer*. Sehingga aplikasi dapat meningkatkan peluang *order* bagi vendor dari sistem *generate* paket. Selain itu

sistem pembayaran ditampung terlebih dahulu oleh IP dan kemudian diberikan ke vendor setelah transaksi selesai. Hal ini dimaksudkan untuk menjaga keamanan transaksi dengan *customer*.

4. Vendor dapat mengelola setiap *brand* yang dimiliki, dengan menerima atau menolak *order* yang didapat dari *customer*.
5. *Management IP* dapat memantau perkembangan transaksi, *customer*, vendor, paket vendor yang ada di dalam aplikasi, serta dapat memantau omzet dan pendapatan yang ada di dalam aplikasi. Laporan yang didapat terdiri dari laporan member vendor, laporan member *customer*, laporan transaksi, laporan pendapatan yang dapat dicetak secara periodik. Ip juga dapat mengetahui peringkat dari setiap vendor yang ada.
6. *Customer* dapat memberikan peringkat & ulasan pada setiap Brand yang tergabung didalam aplikasi.
7. Vendor dapat memantau perkembangan tiap brand yang dimiliki, mengetahui pemasukan setiap periode (tahun, bulan, hari), dan dapat selalu mengetahui jadwal acara terdekat dari tiap brand kategori dari dashboard.
8. Pada bagian vendor, akan mendapatkan laporan berupa jumlah transaksi yang dimiliki, serta pendapatan setiap periode.
9. Didalam aplikasi ini vendor dapat peluang untuk direkomendasikan mendapat order tambahan dari hasil generate sesuai kriteria *customer*.
10. Aplikasi ini telah menerapkan konsep *crowdsourcing*, yang dapat dilihat dari kebebasan vendor untuk mengelola paket dan kategori yang dimiliki dan *customer* bebas untuk melakukan *order* dari banyak vendor.

## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Dari hasil implementasi, uji coba dan evaluasi sistem pada aplikasi penentuan *vendor* Pernikahan untuk usaha *Wedding Organizer* menggunakan metode SMARTER berbasis web, maka kesimpulan yang didapat adalah sebagai berikut:

1. Aplikasi mampu memberikan rekomendasi paket pernikahan sesuai dengan kriteria yang dibutuhkan oleh *customer* dari vendor yang terbaik dengan menggunakan metode SMARTER. Kriteria tersebut meliputi *budget customer*, *rating*, dan pengalaman.
2. Aplikasi dapat membantu *customer* untuk membuat paket pernikahan secara kustomisasi (memilih satu persatu sesuai keinginan) dari tiap kategori vendor yang ada tanpa harus menghitung kembali biaya paket satu per satu karena sistem akan menghitung secara otomatis.
3. Aplikasi dapat menjadi wadah untuk mengembangkan potensi vendor dengan cara menawarkan paket yang dimiliki vendor kepada *customer* dengan melakukan *generate* paket sesuai kriteria *customer*. Sehingga aplikasi dapat meningkatkan peluang *order* bagi vendor dari sistem *generate* paket. Selain itu sistem pembayaran ditampung terlebih dahulu oleh IP dan kemudian diberikan ke vendor setelah transaksi selesai. Hal ini dimaksudkan untuk menjaga keamanan transaksi dengan *customer*.

## 5.2 Saran

Adapun saran yang dapat diberikan kepada peneliti berikutnya adalah :

1. Aplikasi ini dapat dikembangkan dalam bentuk *Mobile Application* berbasis *android* ataupun *ios*.
2. Sistem ini dapat diintegrasikan dengan metode pembayaran yang beragam.
3. Mengatur UI dan UX berdasarkan UX Researcher.
4. Menggunakan Protokol SSL untuk keamanan transaksi data pada aplikasi.



## DAFTAR PUSTAKA

- Calvin, J. (2015, Oktober 15). *Plus-Minus Crowdsourcing Sebagai Sebuah Konsep Bisnis*. Dipetik Februari 25, 2018, dari Kompasiana: [https://www.kompasiana.com/janrycalvin/plus-minus-crowdsourcing-sebagai-sebuah-konsep-bisnis\\_561fd6b0137b61a5188b456b](https://www.kompasiana.com/janrycalvin/plus-minus-crowdsourcing-sebagai-sebuah-konsep-bisnis_561fd6b0137b61a5188b456b)
- Dian, N. (2013, Februari 21). *Kriteria Website yang Baik*. Dipetik Januari 18, 2018, dari Kompasiana: [https://www.kompasiana.com/speedseo/kriteria-website-yang-baik-dibahas-lengkap-di-sini\\_5529a2b7f17e61f211d623d0](https://www.kompasiana.com/speedseo/kriteria-website-yang-baik-dibahas-lengkap-di-sini_5529a2b7f17e61f211d623d0)
- Dian, N. (2015, Juni 24). *Kriteria Website yang baik*. Dipetik Maret 12, 2018, dari Kompasiana: [https://www.kompasiana.com/speedseo/kriteria-website-yang-baik-dibahas-lengkap-di-sini\\_5529a2b7f17e61f211d623d0](https://www.kompasiana.com/speedseo/kriteria-website-yang-baik-dibahas-lengkap-di-sini_5529a2b7f17e61f211d623d0)
- Guntoro, R. S. (2014). Pembuatan Sistem Pendukung Keputusan untuk Pemilihan Vendor Acara Pernikahan Menggunakan Metode Multi-Attribute Global Inference of Quality (MAGIQ).
- Hakim, L., & Musalini, U. (2004). *Cara Cerdas Menguasai Layout, Desain, dan Aplikasi Web*. Jakarta: PT Elex Media Komputindo.
- Howe, J. (2009). *Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business*. New York: Crown Business.
- Jogiyanto. (2005). *Analisis dan Desain Sistem Informasi pendekatan terstruktur teori dan praktik aplikasi bisnis*. Yogyakarta: Andi Offset.
- Julisman, A. (2014). *Sistem Aplikasi Travel dengan Angular JS & Codeigniter*. Yogyakarta: Lokomedia.
- Kadir, A. (2009). *Membuat Aplikasi Web dengan PHP + Database MySQL*. Yogyakarta: CV Andi Offset.
- Kristin, D. M., & Lisanti, Y. (2014). *Wedding Organizer Order Management. ComTech: Computer, Mathematics and Engineering Applications*.
- Priyolistiyanto, A. (2013, November 23). Implementasi Metode Simple Multi Attribute Rating Technique Exploiting Rank (SMARTER) pada Sistem



Pendukung Keputusan Sanksi Pelanggaran Tata Tertib Sekolah. *Prosiding Seminar Nasional Ilmu Komputer*, 127-123.

Ratna, A. L. (2014, Mei). *Pengertian PHP dan MySQL*. Dipetik Februari 25, 2018, dari Ilmu Teknologi Informasi: <http://ilmuti.org/wp-content/uploads/2014/05/Adis-Lena-Kusuma-ratna-Pengertian-PHP-dan-MySQL.pdf>

Rijazim, G. A. (2016). Pembangunan Modul Penawaran Pada Siapsiapnikah.Com Dengan Metode Extreme Programming Dan Konsep Crowdsourcing. *Jurnal Rekayasa Sistem & Industri*, 1-3.

Romeo. (2003). *Testing dan Implementasi Sistem, Edisi Pertama*. Surabaya: Stikom Surabaya.

Septian, R. F. (2014). *Modul Pelatihan CodeIgniter*. Bandung: Lab Basis Data Ilmu Komputer.

Sutarman. (2003). *Membangun Aplikasi Web dengan PHP dan MySQL*. Yogyakarta: Graha Ilmu.

Teambangun. (2017, Agustus 2). *Fungsi dan Peran Wedding Organizer*. Diambil kembali dari [http://www.kompasiana.com/teambangun/fungsi-dan-peran-wedding-organizer\\_55299be3f17e61aa0dd623d6](http://www.kompasiana.com/teambangun/fungsi-dan-peran-wedding-organizer_55299be3f17e61aa0dd623d6)

