



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - IS184853**

**PENGEMBANGAN APLIKASI FRONTEND LINTAS  
PLATFORM ANDROID DAN IOS UNTUK SISTEM  
MANAJEMEN VENDOR UMKM GIYOMI.ID  
MENGUNAKAN KERANGKA KERJA FLUTTER**

***DEVELOPMENT OF CROSS PLATFORM ANDROID  
AND IOS FRONTEND APPLICATION FOR MSME  
GIYOMI.ID VENDOR MANAGEMENT SYSTEM  
USING FLUTTER FRAMEWORK***

**ALVIAN EKA MAULANA  
NRP 05211640000047**

**DOSEN PEMBIMBING  
Mahendrawathi ER, ST, M.Sc, Ph.D  
Rully Agus Hendrawan, S.Kom, M.Eng**

**DEPARTEMEN SISTEM INFORMASI  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
Surabaya 2020**







**TUGAS AKHIR - IS184853**

**PENGEMBANGAN APLIKASI FRONTEND  
LINTAS PLATFORM ANDROID DAN IOS  
UNTUK SISTEM MANAJEMEN VENDOR UMKM  
GIYOMI.ID MENGGUNAKAN KERANGKA  
KERJA FLUTTER**

**ALVIAN EKA MAULANA  
0521164000047**

**Dosen Pembimbing  
Mahendrawathi ER, ST, M.Sc, Ph.D  
Rully Agus Hendrawan, S.Kom, M.Eng**

**DEPARTEMEN SISTEM INFORMASI  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
Surabaya 2019**





**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**UNDERGRADUATE THESIS - IS184853**

**DEVELOPMENT OF CROSS PLATFORM  
ANDROID AND IOS FRONTEND APPLICATION  
FOR MSME GIYOMI.ID VENDOR  
MANAGEMENT SYSTEM USING FLUTTER  
FRAMEWORK**

**ALVIAN EKA MAULANA**  
0521164000047

**Supervisor**

**Mahendrawathi ER, ST, M.Sc, Ph.D**  
**Rully Agus Hendrawan, S.Kom, M.Eng**

**DEPARTMENT OF INFORMATION SYSTEMS**  
**Faculty of Intelligent Electrical and Informatics Technology**  
**Sepuluh Nopember Institute of Technology**  
**Surabaya 2020**



**LEMBAR PENGESAHAN****PENGEMBANGAN APLIKASI FRONTEND LINTAS  
PLATFORM ANDROID DAN IOS UNTUK SISTEM  
MANAJEMEN VENDOR UMKM GIYOMI.ID  
MENGUNAKAN KERANGKA KERJA FLUTTER****TUGAS AKHIR**

Disusun Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer (S.Kom)

pada

Departemen Sistem Informasi

Fakultas Teknologi Elektro dan Informatika Cerdas (ELECTICS)

Institut Teknologi Sepuluh Nopember

Oleh

**Alvian Eka Maulana**

**05211640000047**

Surabaya, 13 Agustus 2020

**Kepala Departemen Sistem Informasi**

**Dr. Muljahidin, ST., MT.**

**NIP. 197010702003121001**







## LEMBAR PERSETUJUAN

### **PENGEMBANGAN APLIKASI FRONTEND LINTAS PLATFORM BERBASIS ANDROID DAN IOS UNTUK SISTEM MANAJEMEN VENDOR UMKM GIYOMLID MENGUNAKAN KERANGKA KERJA FLUTTER**

#### **TUGAS AKHIR**

Disusun Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Departemen Sistem Informasi  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember

Oleh:


**ALVIAN EKA MAULANA**  
NRP. 0521164000047

Disetujui Tim Penguji: Tanggal Ujian: Juni 2020

Periode Wisuda: September 2020

Mahendrawathi ER, S.T., M.Sc., Ph.D.  (Pembimbing I)

Rully Agus Hendrawan, M.Eng.  (Pembimbing II)

Erma Suryani, S.T., M.T., Ph.D.  (Penguji I)

Andre Parvian Aristio, S.Kom., M.Sc.  (Penguji II)



**PENGEMBANGAN APLIKASI FRONTEND LINTAS  
PLATFORM ANDROID DAN IOS UNTUK SISTEM  
MANAJEMEN VENDOR UMKM GIYOMI.ID  
MENGUNAKAN KERANGKA KERJA FLUTTER**

**Nama Mahasiswa** : Alvian Eka Maulana  
**NRP** : 0521164000047  
**Departemen** : Sistem Informasi FTEIC-ITS  
**Dosen Pembimbing I** : Mahendrawathi E.R., S.T., M.Sc.,  
Ph. D.  
**Dosen Pembimbing II** : Rully Agus Hendrawan, S.Kom.,  
M.Eng.

**ABSTRAK**

*Giyomi.id adalah UMKM yang memproduksi berbagai macam pakaian wanita. Mereka memproduksi produknya dengan merancang ide desain pakaian terlebih dahulu dan menugaskan vendor untuk memproduksi pakaian wanita sesuai dengan ide desain sebelumnya, oleh karena itu Giyomi.id memiliki ketergantungan kepada setiap vendor mereka. Ketergantungan tersebut membuat pengawasan terhadap vendor menjadi faktor sangat penting bagi keberlangsungan bisnis UMKM ini, karena mereka perlu untuk memutuskan keberlangsungan kerja sama terhadap vendor sesuai dengan performa masing-masing vendor, namun saat ini pengawasan terhadap vendor masih menggunakan metode konvensional yaitu melalui telepon, sehingga kurang efektif dan sulit untuk dilakukan. Hal tersebut dapat memberikan dampak buruk yaitu keterlambatan proses produksi dan akhirnya dapat menimbulkan kerugian finansial. Melihat permasalahan pengawasan vendor yang ada di Giyomi.id, diperlukan sebuah sistem manajemen vendor yang dapat memantu untuk mengawasi kinerja setiap vendor.*

*Tugas akhir yang sebelumnya menghasilkan luaran berupa desain prototipe aplikasi yang akan digunakan sebagai referensi sistem yang penulis teruskan. Sistem tersebut dapat dikembangkan dengan menggunakan kerangka kerja Flutter,*

*sistem tersebut hanya berfokus pada bagian pengembangan aplikasi frontend saja. Aplikasi frontend yang dikembangkan dapat berjalan secara cross platform di beberapa sistem operasi yaitu Android dan juga Ios. Proses pengujian kinerja aplikasi frontend dilakukan dengan menggunakan beberapa metode testing yaitu Black Box Testing dan Usability Testing. Pengembangan sistem ini dilakukan dengan metode Systems Development Life Cycle (SDLC) yang berjenis waterfall.*

*Hasil yang didapatkan dari tugas akhir ini adalah aplikasi frontend sistem manajemen vendor yang sudah sesuai dengan desain prototipe, dapat terintegrasi dengan aplikasi backend yang sudah dibuat sebelumnya, dan juga dokumentasi aplikasi frontend yang berguna untuk keperluan tugas akhir selanjutnya. Dengan adanya sistem ini, Giyomi.id dapat melakukan proses pengawasan kinerja setiap vendor yang bekerja sama dengan mereka secara efektif dan efisien.*

***Kata Kunci: UMKM, Sistem Manajemen Vendor, Frontend, Flutter, Cross Platform***

**DEVELOPMENT OF CROSS PLATFORM ANDROID  
AND IOS FRONTEND APPLICATION  
DEVELOPMENT FOR MSME GIYOMI.ID VENDOR  
MANAGEMENT SYSTEM USING FLUTTER  
FRAMEWORK**

**Student Name** : Alvian Eka Maulana  
**NRP** : 05211640000047  
**Departement** : Sistem Informasi FTEIC-ITS  
**Supervisor I** : Mahendrawathi E.R., S.T., M.Sc.,  
Ph. D.  
**Supervisor II** : Rully Agus Hendrawan, S.Kom.,  
M.Eng.

**ABSTRACT**

*Giyomi.id is an MSME that produces various kinds of women's clothing. They produce their products by designing clothing design ideas in advance and assigning vendors to produce women's clothing according to the previous design ideas, therefore Giyomi.id has a dependency on each of their vendors. This dependence makes supervision of vendors a very important factor for the sustainability of this MSME business, because they need to decide on the continuity of cooperation with vendors according to the performance of each vendor, but currently monitoring vendors still use conventional methods, namely by telephone, so it is less effective and difficult to do. This can have a negative impact, namely delays in the production process and ultimately can cause financial losses. Seeing the problem of vendor supervision at Giyomi.id, a vendor management system is needed that can help oversee the performance of each vendor.*

*Previous final project produced an output in the form of an application prototype design that will be used as a system reference that the author continues. The system can be developed using Flutter framework, the system only focuses on the frontend application development section. The frontend*

*application being developed can run cross-platform on several operating systems, namely Android and Ios. The frontend application performance testing process is carried out using several testing methods, namely Black Box Testing and Usability Testing. The development of this system is carried out using the waterfall-type of Systems Development Life Cycle (SDLC) method.*

*The results obtained from this final project is a vendor management system frontend application that is in accordance with the protypte design, can be integrated with the backend application that hhas been made before, and also frontend application documentation which is useful for the purposes of the next final project. With this system, Giyomi.id can carry out the process of monitoring the performance of each vendor who works with them effectively and efficiently.*

***Keywords: MSME, Vendor Management System, Frontend, Flutter, Cross Platform***

## SURAT PERNYATAAN BEBAS PLAGIARISME

Saya yang bertandatangan di bawah ini:

Nama : Alvian Eka Maulana  
NRP : 05211640000047  
Tempat/Tanggal lahir : Blitar, 16 September 1998  
Fakultas/Departemen : FTEIC / Sistem Informasi  
Nomor Telp/Hp/email : 081803276464

Dengan ini menyatakan dengan sesungguhnya bahwa penelitian/makalah/tugas akhir saya yang berjudul

PENGEMBANGAN APLIKASI FRONTEND LINTAS PLATFORM ANDROID DAN IOS UNTUK SISTEM MANAJEMEN VENDOR UMKM GIYOMI.ID MENGGUNAKAN KERANGKA KERJA FLUTTER

**Bebas Dari Plagiarisme Dan Bukan Hasil Karya Orang Lain.**

Apabila dikemudian hari ditemukan seluruh atau sebagian penelitian/makalah/tugas akhir tersebut terdapat indikasi plagiarisme, maka saya bersedia menerima sanksi sesuai peraturan dan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sesungguhnya dan untuk dipergunakan sebagaimana mestinya.

Surabaya, 10 Agustus 2020



Alvian Eka Maulana  
NRP.05211640000047



*Halaman ini sengaja dikosongkan.*

## KATA PENGANTAR

Puji syukur penulis haturkan kepada Allah SWT yang telah melimpahkan rahmat dan hidayahnya sehingga penulis dapat menyelesaikan tugas akhir dengan judul “Pengembangan Aplikasi Frontend Lintas Platform Android Dan IOS Untuk Sistem Manajemen Vendor UMKM Giyomi.id Menggunakan Kerangka Kerja Flutter” sebagai salah satu syarat yang harus dipenuhi untuk mendapatkan gelar sarjana komputer di departemen sistem informasi fakultas teknologi elektro dan informatika cerdas ITS.

Dalam penyusunan tugas akhir ini tentunya penulis tidak menyelesaikannya sendiri dengan mudah, terdapat banyak bantuan yang didapatkan dari berbagai pihak dalam bentuk doa, semangat, arahan, kritik, saran dan berbagai bantuan lainnya. Tanpa mengurangi rasa hormat, penulis ingin menyampaikan ucapan terima kasih kepada :

1. Allah SWT atas segala rahmat dan karunianya telah diberikan selama ini
2. Kedua orang tua penulis yang selalu memberikan dukungan material dan doa demi terselesaikannya pengerjaan tugas akhir ini.
3. Ibu Mahendrawathi E.R., S.T., M.Sc., Ph. D. dan Bapak Rully Agus Hendrawan, S.Kom., M.Eng. selaku pembimbing tugas akhir karena telah memberikan arahan mengenai penyusunan penelitian ini hingga selesai.
4. Ibu Erma Suryani, S.T, M.T, Ph.D dan Bapak Andre Parvian Aristio, S.Kom, M.Sc selaku penguji yang memberikan semua masukan serta koreksi untuk menyempurnakan tugas akhir ini, serta ibu Renny Pradina Kusumawardani S.T, M.T, SCJP yang telah menjadi pembimbing proposal pengajuan tugas akhir ini sehingga proposal pengajuan tugas akhir ini dapat disetujui untuk dikerjakan.

5. Bapak Yudha Prasetyo selaku Owner Giyomi.id karena telah memberikan izin untuk melakukan pengembangan aplikasi di UMKM Giyomi.id
6. Tim pengembang Sistem Manajemen Vendor yaitu Pahang Krisdyan karena telah berkenan untuk berkolaborasi dalam melakukan penelitian ini.
7. Serta seluruh pihak - pihak lain yang tidak dapat disebutkan satu per satu yang telah banyak membantu penulis selama perkuliahan hingga dapat menyelesaikan tugas akhir ini.

Penulis sadar bahwa tugas akhir ini sangat jauh dari kata sempurna, sehingga semua saran dan kritik yang bersifat membangun dari pembaca akan sangat dihargai oleh penulis untuk perbaikan kedepannya. Semoga tugas akhir ini bermanfaat bagi perkembangan ilmu pengetahuan dan teknologi semua pihak.

Surabaya, 31 Mei 2020

Penulis

## DAFTAR ISI

LEMBAR PERSETUJUAN .....	viii
ABSTRAK.....	x
ABSTRACT .....	xii
KATA PENGANTAR.....	xvi
DAFTAR ISI .....	xviii
DAFTAR GAMBAR.....	xxii
BAB 1 PENDAHULUAN.....	1
1.1    Latar Belakang Masalah.....	1
1.2    Rumusan Masalah.....	3
1.3    Batasan Masalah .....	4
1.4    Tujuan Penelitian .....	4
1.5    Manfaat Penelitian .....	4
1.6    Relevansi.....	5
BAB 2 TINJAUAN PUSTAKA.....	7
2.1    Penelitian Sebelumnya.....	7
2.1.1    Pengembangan frontend aplikasi uiiportofolio berbasis web.....	7
2.1.2    Perancangan Prototipe Sistem Manajemen Vendor dengan Metode Iterative Prototyping dan Moderated Usability Test. (Studi Kasus: Giyomi.Id).....	8
2.1.3    Flutter for cross platform App and SDK development.....	9
2.1.4    Comparing Black- and White-box testing environments using mixed research methods and AHP.....	9
2.2    Dasar Teori.....	10
2.2.1    Aplikasi Frontend.....	11
2.2.2    Flutter .....	11
2.2.3    Dart.....	12
2.2.4    Widget.....	12
2.2.5    Routing.....	12
2.2.6    JSON .....	13
2.2.7    Cross Platform.....	13
2.2.8    Blacbox Testing .....	13
2.2.9    Usability Testing .....	13

2.2.10	Software Development Life Cycle Waterfall.....	14
<b>BAB 3</b>	<b>METODOLOGI.....</b>	<b>17</b>
3.1	Diagram Metode Pelaksanaan Tugas Akhir.....	17
3.2	Uraian Metode Pelaksanaan Tugas Akhir.....	19
3.2.1	Identifikasi dan Perumusan Masalah.....	19
3.2.2	Studi Literatur .....	19
3.2.3	Analisis.....	20
3.2.4	Perancangan .....	20
3.2.5	Pengembangan .....	21
3.2.6	Testing.....	22
3.2.7	Penyusunan Laporan Tugas Akhir .....	23
<b>BAB 4</b>	<b>ANALISIS .....</b>	<b>25</b>
4.1	Analisis Kebutuhan User Interface .....	25
4.2	Analisis Kebutuhan Application Programming Interface (API).....	26
<b>BAB 5</b>	<b>PERANCANGAN .....</b>	<b>29</b>
5.1	Perancangan Arsitektur Aplikasi.....	29
5.2	Perancangan Arsitektur Informasi.....	33
<b>BAB 6</b>	<b>PENGEMBANGAN APLIKASI.....</b>	<b>35</b>
6.1	Kebutuhan Dasar Pengembangan.....	35
6.2	Pengembangan User Interface.....	35
6.2.1	Pengembangan Komponen Input .....	36
6.2.2	Pengembangan Komponen Button.....	37
6.2.3	Pengembangan Komponen Counter.....	38
6.2.4	Pengembangan Komponen Progress.....	40
6.2.5	Pengembangan Komponen Card.....	41
6.2.6	Pengembangan Komponen Sort.....	42
6.2.7	Pengembangan Komponen Rating.....	43
6.2.8	Pengembangan Komponen Image.....	44
6.2.9	Pengembangan Komponen AddMore.....	45
6.2.10	Pengembangan Komponen Dropdwon .....	46
6.2.11	Pengembangan Komponen Score .....	47
6.2.12	Pengembangan Komponen Table .....	47
6.2.13	Pengembangan Komponen Header.....	49
6.2.14	Pengembangan Komponen BottomTab.....	50
6.2.15	Navigasi User Interface .....	51
6.3	Integrasi API .....	52
6.3.1	Konfigurasi API .....	52
6.3.2	Penggunaan API.....	53

6.4	Manajemen Vendor.....	56
6.4.1	Penilaian Performa Vendor.....	56
6.4.2	Detail Informasi Vendor .....	57
6.4.3	Keterlambatan Vendor .....	58
<b>BAB 7 PENGUJIAN .....</b>		<b>61</b>
7.1	Blackbox Testing .....	61
7.2	Usability Testing .....	62
7.2.1	Tahap Awal.....	62
7.2.2	Protokol Pengujian .....	63
7.2.3	Skenario Pengujian .....	63
7.2.4	Traceability Matrix .....	65
7.2.5	Karakteristik Responden.....	67
7.2.6	Hasil Pengujian .....	67
7.2.7	Effectivity Matrix.....	71
<b>BAB 8 PENUTUP .....</b>		<b>73</b>
8.1	Kesimpulan .....	73
8.2	Saran.....	74
8.2.1	Saran untuk Penelitian Selanjutnya.....	74
8.2.2	Saran untuk UMKM Giyomi.id .....	74
<b>DAFTAR PUSTAKA.....</b>		<b>75</b>
<b>LAMPIRAN A . USER INTERFACE APLIKASI .....</b>		<b>77</b>
<b>LAMPIRAN B . BLACKBOX TESTING .....</b>		<b>87</b>
<b>LAMPIRAN C . PROTOKOL PENGUJIAN.....</b>		<b>117</b>
<b>LAMPIRAN D . FEEDBACK PENGGUNA.....</b>		<b>119</b>
<b>BIODATA PENULIS.....</b>		<b>123</b>

*Halaman ini sengaja dikosongkan.*

## DAFTAR GAMBAR

Gambar 1.1 Kerangka Kerja Riset Laboratorium Sistem Enterprise .....	5
Gambar 2.1 Paradigma API .....	11
Gambar 2.2 Logo Kerangka Kerja Flutter .....	11
Gambar 2.3 Tahapan SDLC Waterfall.....	14
Gambar 3.1 Diagram Metodologi 1 .....	17
Gambar 3.2 Diagram Metodologi 2 .....	18
Gambar 5.1 Perintah Unduh File Flutter.....	29
Gambar 5.2 Perintah Simpan <i>PATH</i> Flutter.....	29
Gambar 5.3 Perintah Pengecekan Kesiapan Flutter.....	30
Gambar 5.4 Perintah Pengecekan Kesiapan Flutter.....	30
Gambar 5.5 Perintah Pembuatan Direktori Aplikasi .....	30
Gambar 5.6 Struktur Direktori Flutter .....	31
Gambar 5.7 Perintah Menjalankan Simulator IOS .....	32
Gambar 5.8 Perintah Menjalankan Aplikasi.....	32
Gambar 5.9 Aplikasi Giyomi di Simulator IOS.....	33
Gambar 5.10 Arsitektur Informasi .....	34
Gambar 6.1 Kode Sumber Komponen Input .....	36
Gambar 6.2 Tampilan Komponen Input .....	37
Gambar 6.3 Tampilan Komponen Button.....	37
Gambar 6.4 Tampilan Komponen Button.....	38
Gambar 6.5 Kode Sumber Komponen Counter .....	39
Gambar 6.6 Tampilan Komponen Counter.....	39
Gambar 6.7 Kode Sumber Komponen Progres.....	40
Gambar 6.8 Tampilan Komponen Progress .....	40
Gambar 6.9 Kode Sumber Komponen Card .....	41
Gambar 6.10 Tampilan Komponen Card.....	42
Gambar 6.11 Kode Sumber Komponen Sort .....	42
Gambar 6.12 Tampilan Komponen Sort.....	43
Gambar 6.13 Kode Sumber Komponen Rating .....	43
Gambar 6.14 Tampilan Komponen Rating .....	44
Gambar 6.15 Kode Sumber Komponen Image.....	44
Gambar 6.16 Tampilan Komponen Image.....	44
Gambar 6.17 Kode Sumber Komponen AddMore .....	45
Gambar 6.18 Tampilan Komponen AddMore .....	45
Gambar 6.19 Kode Sumber Komponen Dropdown.....	46



Gambar 6.20 Tampilan Komponen Dropdown.....	46
Gambar 6.21 Kode Sumber Komponen Score .....	47
Gambar 6.22 Tampilan Komponen Score .....	47
Gambar 6.23 Kode Sumber Komponen Table .....	48
Gambar 6.24 Tampilan Komponen Table.....	48
Gambar 6.25 Kode Sumber Komponen Header.....	49
Gambar 6.26 Tampilan Komponen Header .....	49
Gambar 6.27 Kode Sumber Komponen BottomTab .....	50
Gambar 6.28 Tampilan Komponen BottomTab.....	50
Gambar 6.29 Kode Sumber Navigasi.....	51
Gambar 6.30 Kode Sumber Penggunaan Navigasi .....	52
Gambar 6.31 Kode Sumber Konfigurasi Package API .....	52
Gambar 6.32 Kode Sumber Konfigurasi Header API.....	53
Gambar 6.33 Kode Sumber Penggunaan API POST .....	53
Gambar 6.34 Kode Sumber Penggunaan API di Komponen .	54
Gambar 6.35 Pengguna Menilai Performa Vendor .....	56
Gambar 6.36 Detail Informasi Vendor di VendorDetail .....	57
Gambar 6.37 Keterlambatan SPK di Home .....	59
Gambar 7.1 Filter SPK berdasarkan Status atau Vendor .....	68
Gambar 7.2 Penambahan Catatan lebih dari dua kali .....	69
Gambar 7.3 Pengambilan dan Penerimaan Barang SPK.....	69
Gambar 7.4 Tambah Varian di SPK.....	70
Gambar 8.1 Login .....	77
Gambar 8.2 Home .....	77
Gambar 8.3 Vendor.....	78
Gambar 8.4 Add Vendor .....	78
Gambar 8.5 Vendor Detail .....	79
Gambar 8.6 SPK.....	79
Gambar 8.7 Add SPK.....	80
Gambar 8.8 SPK Detail.....	80
Gambar 8.9 SPK Progress.....	81
Gambar 8.10 SPK Received.....	81
Gambar 8.11 SPK Evaluation .....	82
Gambar 8.12 Ongoing SPK.....	82
Gambar 8.13 Finished SPK .....	83
Gambar 8.14 SPK Draft .....	83
Gambar 8.15 SPK Validation.....	84
Gambar 8.16 Edit SPK .....	84

Gambar 8.17 Account .....	85
Gambar 8.18 Account List .....	85
Gambar 8.19 Add Account .....	86
Gambar 8.20 Edit Account.....	86

## DAFTAR TABEL

Tabel 2.1 Keterangan Literatur 1 .....	7
Tabel 2.2 Keterangan Literatur 2 .....	8
Tabel 2.3 Keterangan Literatur 3 .....	9
Tabel 2.4 Keterangan Literatur 4 .....	9
Tabel 4.1 Kebutuhan <i>User Interface</i> .....	25
Tabel 4.2 Kebutuhan API.....	26
Tabel 5.1 Keterangan Sub Folder.....	31
Tabel 6.1 <i>Tools</i> dan <i>Platform</i> yang digunakan .....	35
Tabel 6.2 Penggunaan Komponen Input di User Interface ....	37
Tabel 6.3 Penggunaan Komponen Button di User Interface..	38
Tabel 6.4 Penggunaan Komponen Counter di User Interface	39
Tabel 6.5 Penggunaan Komponen Button di User Interface..	41
Tabel 6.6 Penggunaan Komponen Card di User Interface.....	42
Tabel 6.7 Penggunaan Komponen Card di User Interface.....	43
Tabel 6.8 Penggunaan Komponen Rating di User Interface..	44
Tabel 6.9 Penggunaan Komponen Image di User Interface...	45
Tabel 6.10 Penggunaan Komponen AddMore di User Interface .....	45
Tabel 6.11 Penggunaan Komponen Dropdown di User Interface.....	46
Tabel 6.12 Penggunaan Komponen Score di User Interface .	47
Tabel 6.13 Penggunaan Komponen Table di User Interface .	49
Tabel 6.14 Penggunaan Komponen Header di User Interface	49
Tabel 6.15 Penggunaan Komponen BottomTab di User Interface.....	51
Tabel 6.16 Kebutuhan API di <i>User Interface</i> .....	54
Tabel 6.17 Kebutuhan API di <i>User Interface</i> .....	55
Tabel 6.18 Parameter Penilaian Performa Vendor .....	56
Tabel 7.1 Test Case Blackbox Testing.....	61
Tabel 7.2 Test Case Usability Testing .....	63
Tabel 7.3 Traceability Matrix.....	66
Tabel 7.4 Karakteristik Responden .....	67
Tabel 8.1 Login .....	87
Tabel 8.2 Menampilkan Akun.....	88
Tabel 8.3 Menampilkan Halaman Edit Akun.....	89

Tabel 8.4 Merubah Informasi Akun.....	90
Tabel 8.5 Menampilkan Daftar Akun Staf.....	91
Tabel 8.6 Menambah Akun Staf Baru .....	92
Tabel 8.7 Logout.....	93
Tabel 8.8 Menampilkan Daftar Vendor .....	94
Tabel 8.9 Sortir Daftar Vendor Berdasarkan Skor.....	95
Tabel 8.10 Menambah Vendor Baru.....	96
Tabel 8.11 Menampilkan Detail Vendor.....	97
Tabel 8.12 Menghubungi Nomor Telepon Vendor.....	98
Tabel 8.13 Menampilkan Daftar SPK.....	99
Tabel 8.14 Sortir Daftar SPK Berdasarkan Vendor .....	100
Tabel 8.15 Menambah Draf SPK Baru .....	101
Tabel 8.16 Meminta Validasi SPK .....	103
Tabel 8.17 Melakukan Validasi SPK.....	104
Tabel 8.18 Melihat SPK Yang Terlambat.....	105
Tabel 8.19 Melihat Detail Koleksi SPK Berjalan .....	106
Tabel 8.20 Menunda SPK Berjalan.....	107
Tabel 8.21 Melanjutkan SPK Yang Ditunda .....	108
Tabel 8.22 Menambahkan Catatan Proses SPK.....	109
Tabel 8.23 Menyelesaikan Proses SPK.....	110
Tabel 8.24 Melihat Proses SPK Yang Telah Selesai .....	111
Tabel 8.25 Menerima Barang SPK .....	112
Tabel 8.26 Mengembalikan Barang SPK.....	113
Tabel 8.27 Menutup SPK.....	114
Tabel 8.28 Mengevaluasi Vendor .....	115
Tabel 8.29 Melihat Progress SPK.....	116
Tabel 8.30 Protokol Pengujian.....	117
Tabel 8.31 Feedback Pengguna Iterasi Pertama .....	119
Tabel 8.32 Feedback Pengguna Iterasi Kedua .....	120

*Halaman ini sengaja dikosongkan.*



## **BAB 1**

### **PENDAHULUAN**

Bab pendahuluan menjelaskan secara umum mengenai dasar dari penulisan tugas akhir yang diusulkan dalam proposal. Dasar dari penulisan meliputi latar belakang, rumusan masalah, batasan masalah, tujuan dan manfaat serta relevansi dari tugas akhir ini. Dari uraian tersebut diharapkan penjelasan umum permasalahan dan solusi tugas akhir ini dapat dipahami.

#### **1.1 Latar Belakang Masalah**

Giyomi.id adalah UMKM yang memproduksi pakaian wanita untuk dijual kepada pelanggan melalui daring maupun luring. Giyomi.id menggunakan *marketplace* dan media sosial sebagai wahana menjual secara daring sedangkan Giyomi.id menjual produknya secara luring di pasar dan bazar. Giyomi melakukan proses produksi mulai dari perancangan ide desain produk-produk baru, pembuatan sampel produk, menugaskan vendor untuk memproduksi produk, menerima produk jadi dari vendor sampai pemasaran produk [1].

Proses produksi Giyomi.id dimulai dari mengembangkan ide produk baru sampai menghasilkan desain *mock up* baju. Proses dilanjutkan dengan merencanakan produksi, mencakup merencanakan jumlah produk yang akan di produksi, biaya produksi dan lini masa produksi yang harus dilakukan oleh vendor. Selanjutnya mengadakan bahan baku yang dibutuhkan melalui pemasok, bahan baku tersebut dikirim oleh pemasok kepada vendor. Proses selanjutnya adalah memberikan Surat Perintah Kerja (SPK) kepada vendor agar pembuatan produk dapat dimulai. Setelah proses pembuatan produk oleh vendor selesai, vendor mengirimkan produk jadi tersebut kepada Giyomi.id [1].

Proses produksi Giyomi.id sangat bergantung pada pihak ketiga yaitu vendor. Kurangnya pengawasan terhadap vendor dapat menjadi masalah terhadap proses produksi seperti keterlambatan produksi yang mengganggu proses penjualan sampai menciptakan kerugian finansial. Pengawasan kinerja vendor seperti mengingatkan akan batas waktu pembuatan produk menjadi hal yang penting bagi Giyomi.id, namun Giyomi.id masih menggunakan metode konvensional melalui telepon untuk mengawasi kinerja vendor. Kinerja vendor yang kurang baik dapat menyebabkan terganggunya proses produksi, mengingat vendor-vendor Giyomi.id tidak hanya melayani produksi Giyomi.id, mereka juga menerima pesanan jasa produksi dari pihak lain. Giyomi.id perlu untuk mempertimbangkan kinerja vendor sebagai pertimbangan keputusan keberlangsungan kerja sama dengan vendor [2].

Tugas akhir yang dilakukan oleh Bening (2019) telah menghasilkan desain prototipe sistem manajemen vendor sesuai kebutuhan Giyomi.id dalam bentuk dokumentasi visual yang dapat digunakan sebagai referensi untuk penelitian lebih lanjut yaitu pengembangan sistem berupa aplikasi *mobile* sistem manajemen vendor. Tugas akhir tersebut belum melakukan pengembangan aplikasi *mobile* sebagai solusi nyata yang dapat digunakan oleh Giyomi.id [2].

Flutter adalah kerangka kerja yang digunakan dalam proses pengembangan aplikasi *mobile*. Kelebihan kerangka kerja ini adalah aplikasi yang dihasilkan bisa berjalan secara lintas platform pada sistem operasi Android dan IOS. Tujuan diciptakannya kerangka kerja ini adalah untuk mengembangkan aplikasi *mobile* berkinerja tinggi. Flutter dirilis untuk umum pada tahun 2016 oleh Google. Kerangka kerja ini menggunakan bahasa pemrograman Dart [3].

Pengembangan perangkat lunak membutuhkan *Software Development Life Cycle* (SDLC) untuk memastikan kualitas



perangkat lunak, mempermudah kontrol implementasi dan memastikan perangkat lunak dapat memenuhi kebutuhan pengguna [4]. *Waterfall* adalah salah satu jenis SDLC, *waterfall* memiliki kelebihan dapat membuat pengembangan perangkat lunak menjadi tepat waktu [5].

Tugas akhir ini akan berfokus pada pengembangan aplikasi *frontend* sistem manajemen vendor. Aplikasi *frontend* ini bertidak sebagai wadah yang siap mengonsumsi layanan dari aplikasi *backend*. Aplikasi *frontend* akan dikembangkan berdasarkan desain prototipe aplikasi yang telah dibuat pada penelitian yang dilakukan oleh Bening (2019). Penelitian ini akan menggunakan metode SDLC *waterfall*. Aplikasi *frontend* ini diharapkan dapat menjadi aplikasi sistem manajemen vendor yang dapat diintegrasikan dengan penelitian sebelumnya yaitu aplikasi *backend* yang dikembangkan oleh Pahang Krisdyan

## 1.2 Rumusan Masalah

Giyomi.id dalam proses produksinya perlu melakukan pengawasan terhadap vendor dan pengambilan keputusan keberlangsungan kerja sama terhadap vendor berdasarkan performa mereka. Giyomi.id **membutuhkan informasi akurat tentang performa tiap vendor yang dapat digunakan sebagai acuan keberlanjutan kerja sama**. Kondisi saat ini, Giyomi.id melakukan pengawasan terhadap vendor menggunakan metode konvensional melalui telepon **yang masih belum efektif untuk memberikan informasi yang akurat tentang performa tiap vendor**. Kondisi tersebut mengakibatkan permasalahan pada proses produksi Giyomi.id **yaitu keterlambatan proses produksi dan memunculkan kerugian finansial**. Penelitian yang dilakukan oleh Bening (2019) menghasilkan desain prototipe sistem manajemen vendor yang telah dirancang dapat menjadi solusi akan permasalahan tersebut. Perlu adanya pengembangan aplikasi *frontend* sistem manajemen vendor berdasarkan desain prototipe yang telah dibuat oleh

peneliti sebelumnya dan dapat diintegrasikan dengan aplikasi *backend* untuk menjadi sebuah sistem yang dapat digunakan oleh Giyomi.id.

### 1.3 Batasan Masalah

Batasan masalah dari penelitian ini yaitu:

1. Pengembangan aplikasi *frontend* Sistem Manajemen Vendor pada Giyomi.id hanya sampai pada tahap *testing*.
2. Aplikasi *frontend* Sistem Manajemen Vendor hanya dikembangkan untuk sistem operasi Android dan IOS.

### 1.4 Tujuan Penelitian

Tujuan dari tugas akhir ini antara lain:

1. Mengembangkan aplikasi *frontend* berdasarkan desain prototipe aplikasi menggunakan kerangka kerja Flutter.
2. Melakukan integrasi aplikasi *frontend* dengan aplikasi *backend*.
3. Membuat aplikasi *frontend* mampu memberikan informasi akurat tentang performa tiap vendor dalam bentuk skor sehingga kerjasama menjadi lebih efektif
4. Membuat aplikasi *frontend* mampu menampilkan keterlambatan proses produksi tiap vendor sehingga mampu meminimalisir kerugian finansial dan proses produksi menjadi lebih efisien
5. Menguji aplikasi *frontend* dengan menggunakan metode *Blackbox Testing* dan *Usability Testing*.

### 1.5 Manfaat Penelitian

Berikut adalah manfaat yang diharapkan dari pengerjaan tugas akhir ini:

1. Giyomi.id dapat menggunakan sistem yang telah diimplementasikan untuk menjalankan proses produksinya
2. Meningkatkan kemampuan analisa vendor dalam proses pemilihan vendor

## 1.6 Relevansi

Tugas Akhir ini mengangkat topik mengenai *Enterprise Resource Planning*. Topik ini memiliki kontribusi pada tujuan tugas akhir Laboratorium *Sistem Enterprise (SE)* Departemen Sistem Informasi ITS, Surabaya, yaitu untuk meningkatkan pemahaman mengenai bagaimana suatu solusi IT di dalam organisasi atau perusahaan dapat membantu mencapai tujuan bisnis pada suatu organisasi. Tugas Akhir ini memiliki relevansi dengan mata kuliah yang dipelajari di Departemen Sistem Informasi, Institut Teknologi Sepuluh Nopember, yaitu Sistem Enterprise (SE), Manajemen Proses Bisnis (MPB), Rancang Bangun Perangkat Lunak (RBPL), serta Algoritma dan Pemrograman (AP). **Gambar 1.1** menunjukkan tujuan tugas akhir dan topik pengetahuan pada Laboratorium *Sistem Enterprise (SE)* antara lain, *Enterprise Resource Planning (ERP)*, *Business Process Management (BPM)*, *Supply Chain Management (SCM)*, dan *Customer Relationship Management (CRM)*.



**Gambar 1.1** Kerangka Kerja Riset Laboratorium Sistem Enterprise

*Halaman ini sengaja dikosongkan*

## BAB 2 TINJAUAN PUSTAKA

Bab ini berisikan kajian pustaka yang akan digunakan dalam penelitian ini. Berisikan penelitian sebelumnya, dasar teori, dan metode yang akan digunakan sebagai acuan atau landasan dalam pengerjaan tugas akhir ini.

### 2.1 Penelitian Sebelumnya

Dalam penelitian ini, digunakan beberapa penelitian terdahulu sebagai pedoman dan referensi dalam melaksanakan proses-proses dalam penelitian, seperti yang terdapat dalam penelitian di tabel berikut, berisi informasi penelitian sebelum serta hubungan penelitian terhadap tugas akhir ini.

#### 2.1.1 Pengembangan *frontend* aplikasi uiiportofolio berbasis web

**Tabel 2.1 Keterangan Literatur 1**

<b>Judul Penelitian</b>	Pengembangan <i>frontend</i> aplikasi uiiportofolio berbasis web
Penulis, tahun	Fakhrul Hasbi, 2018
Deskripsi umum penelitian	Penelitian ini membahas tentang desain dan pengembangan aplikasi <i>frontend</i> uiiportofolio yang berbasis <i>website</i> . Metode yang digunakan yaitu pengumpulan data dari penggunaan, analisis dan perancangan yang memiliki luaran berupa visualisasi berbentuk diagram, peninjauan dari para ahli, implementasi aplikasi <i>frontend</i> , melakukan integrasi dengan aplikasi <i>backend</i> , dan terakhir melakukan pengujian. [6]
Keterkaitan penelitian	Penelitian ini memiliki keterkaitan yang dekat dengan tugas akhir. Metode

	yang digunakan untuk mengembangkan aplikasi <i>frontend</i> mulai dari analisis, pengembangan sampai <i>testing</i> dapat di aplikasikan ke dalam tugas akhir.
--	--

2.1.2 *Perancangan Prototipe Sistem Manajemen Vendor dengan Metode Iterative Prototyping dan Moderated Usability Test. (Studi Kasus: Giyomi.Id)*

**Tabel 2.2 Keterangan Literatur 2**

<b>Judul Penelitian</b>	Perancangan Prototipe Sistem Manajemen Vendor dengan Metode <i>Iterative Prototyping</i> dan <i>Moderated Usability Test</i> . (Studi Kasus: Giyomi.Id)
Penulis, tahun	Erica Maulidina Bening, 2019
Deskripsi umum penelitian	Penelitian ini berfokus pada pembuatan desain prototipe sistem manajemen vendor sesuai dengan pengguna dan terdokumentasi secara visual dengan baik, demi penelitian selanjutnya yaitu pengembangan aplikasi sesuai desain prototipe yang telah dibuat. Penelitian ini menggunakan metode <i>iterative prototyping</i> yang mengharuskan proses penelitian untuk secara berulang untuk meminta respon balik dari pengguna akan desain aplikasi yang telah dibuat. Penelitian ini menghasilkan desain prototipe aplikasi yang dapat digunakan sebagai panduan bagi penelitian selanjutnya untuk mengembangkan aplikasi agar dapat digunakan secara riil. [2]
Keterkaitan penelitian	Penelitian ini menghasilkan desain prototipe aplikasi yang akan menjadi panduan untuk pengembangan aplikasi <i>frontend</i> . Penelitian ini juga berguna

	untuk tugas akhir agar dapat berfokus pada pengembangan tanpa perlu untuk melakukan perancangan desain prototipe sesuai kebutuhan pengguna.
--	---

### 2.1.3 Flutter for cross platform App and SDK development

**Tabel 2.3 Keterangan Literatur 3**

<b>Judul Penelitian</b>	<i>Flutter for cross platform App and SDK development</i>
Penulis, tahun	Lukas Dagne, 2019
Deskripsi umum penelitian	Penelitian ini berfokus untuk meneliti perbandingan performa antara Flutter dengan <i>multiple</i> JWT untuk digunakan dalam pengamanan <i>web service</i> pada satu atau banyak server. Hasil dari perbandingan tersebut menunjukkan bahwa penggunaan Flutter memiliki performa yang lebih tinggi dibanding kerangka kerja lain dalam hal siklus pengembangan sistem, serta teknik untuk mengakses layanan dari masing-masing sistem operasi Android dan IOS yang membuat <i>user experience</i> lebih mulus. [7]
Keterkaitan penelitian	Tugas akhir akan menggunakan Flutter untuk mengembangkan aplikasi <i>frontend</i> . Penelitian ini memberikan perbandingan performa antara penggunaan Flutter dengan kerangka kerja <i>cross platform</i> lain yang dapat menjadi acuan tugas akhir dalam menentukan penggunaan Flutter pada tugas akhir.

### 2.1.4 Comparing Black- and White-box testing environments using mixed research methods and AHP

**Tabel 2.4 Keterangan Literatur 4**

<b>Judul Penelitian</b>	<i>Comparing Black- and White-box testing environments using mixed research methods and AHP</i>
Penulis, tahun	Amiryasin Fallah Daryavarsari, 2013
Deskripsi umum penelitian	Penelitian ini membahas tentang pengurangan biaya, fleksibilitas dalam manajemen, dan meningkatkan efisiensi dengan menggunakan dua metode pengujian yang selama ini dianggap tumpang tindih yaitu <i>Black Box</i> dan <i>White Box Testing</i> . Menghapus satu metode atau mengintegrasikan salah satu jenis pengujian dengan pengujian yang lain dapat menghemat sumber daya yang digunakan. Dalam proses penelitian ini akan digunakan dua metode penilaian yaitu secara kualitatif dan kuantitatif. Dua kuesioner digunakan untuk mengungkap pendapat setiap penguji sehingga dapat memberikan penilaian yang adil ke setiap metode pengujian. [8]
Keterkaitan penelitian	Tugas akhir akan menggunakan <i>Black Box Testing</i> untuk menguji aplikasi <i>frontend</i> yang dikembangkan. Penelitian ini memberikan perbandingan antara <i>Black Box</i> dan <i>White Box Testing</i> yang dapat menjadi acuan tugas akhir. Selain itu penelitian ini juga memberitahukan cara yang tepat untuk perancangan dan implementasi <i>Black Box Testing</i> .

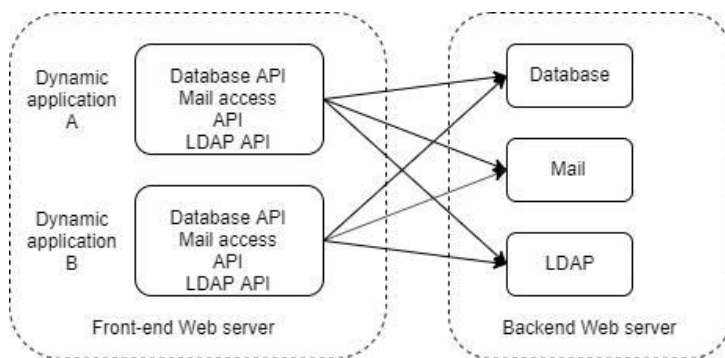
## 2.2 Dasar Teori

Dasar teori digunakan untuk menjelaskan beberapa metode, formula dan istilah yang dipakai untuk melakukan penelitian ini. Berikut beberapa teori yang akan mendukung pembaca dalam memahami penelitian ini dan mendukung untuk melakukan tugas akhir ini :



### 2.2.1 Aplikasi Frontend

Aplikasi *frontend* adalah aplikasi yang berjalan pada bagian klien yang bertujuan sebagai wadah untuk mengonsumsi layanan dari bagian server. Aplikasi *frontend* mengakses aplikasi *backend* melalui *application programming interface* (API) [9]. Paradigma yang diadopsi pada klien dapat dijelaskan oleh **Gambar 2.1**. Paradigma API



**Gambar 2.1 Paradigma API**

### 2.2.2 Flutter



**Gambar 2.2 Logo Kerangka Kerja Flutter**

Flutter adalah kerangka kerja yang digunakan dalam proses pengembangan aplikasi *mobile*. Kelebihan kerangka kerja ini adalah aplikasi yang dihasilkan bisa berjalan secara lintas platform pada sistem operasi Android dan IOS. Tujuan diciptakannya kerangka kerja ini adalah untuk mengembangkan aplikasi *mobile* berkinerja tinggi. Flutter dirilis untuk umum pada tahun 2016 oleh Google. Kerangka kerja ini menggunakan bahasa pemrograman Dart [3].

### 2.2.3 *Dart*

Pada kerangka kerja Flutter, semua aplikasi ditulis dengan Dart. Dart adalah bahasa pemrograman yang dikembangkan dan dikelola oleh Google. Ini banyak digunakan di dalam Google dan telah terbukti memiliki kemampuan untuk mengembangkan aplikasi web berskala besar. Selain itu Dart memiliki sintaksis yang lebih mirip bahasa pemrograman Java. [3]

### 2.2.4 *Widget*

*Widget* adalah elemen terpenting dalam kerangka kerja Flutter. *Widget* harus menarik dan masuk akal karena pengguna melihat dan merasakannya secara langsung di dalam aplikasi. *Widget* tidak hanya mengontrol dan memengaruhi tampilan, tetapi juga menangani dan merespons tindakan pengguna. Oleh karena itu, sangat penting bahwa *widget* harus berkinerja cepat, termasuk *rendering* dan tampilan animasi. *Widget* milik Flutter ini terkenal sangat mudah di modifikasi daripada kerangka kerja *cross platform* yang lain. [3]

### 2.2.5 *Routing*

*Routing* adalah salah satu spesifikasi paling penting untuk membuat struktur aplikasi. *Routing* merupakan istilah yang digunakan untuk merancang jalur rute ketika aplikasi berjalan, pengembang dapat memisahkan seluruh aplikasi

menjadi beberapa bagian dan memperjelas logika setiap navigasi terlebih dahulu. Flutter menyediakan *widget* tersendiri untuk membuat navigasi aplikasi yang bernama *Navigator*.

### 2.2.6 *JSON*

JSON (JavaScript Object Notation) adalah sebuah format data yang digunakan untuk pertukaran dan penyimpanan data. JSON merupakan bagian (*subset*) dari Javascript. JSON bisa dibaca dengan berbagai macam bahasa pemrograman seperti C, C++, C#, Java, Javascript Perl, Python, dan banyak lagi. Hal ini membuat JSON menjadi bahasa yang ideal untuk pertukaran data antar aplikasi.

### 2.2.7 *Cross Platform*

*Cross platform* adalah proses pengembangan yang membuat pengembang cukup menuliskan basis kode tunggal tetapi bisa berjalan di berbagai macam *platform* seperti Android dan IOS bagi pengembang aplikasi seluler. Daya tarik utama pengembangan *cross platform* dibandingkan pengembangan *native* adalah pengembang bisa meminimalkan ukuran, waktu, dan sumber daya tim mereka. [7]

### 2.2.8 *Blacbox Testing*

*Blackbox Testing* adalah metode yang digunakan untuk menguji perilaku aplikasi yang dikembangkan. Pengujian ini tidak menggunakan informasi dari struktur internal maupun prosedur suatu perangkat lunak. *Black Box Testing* lebih berfokus pada fungsionalitas aplikasi. Metode ini digunakan oleh penguji aplikasi untuk memberikan masukan yang sesuai dan memeriksa apakah aplikasi memenuhi persyaratan fungsional atau tidak. [10]

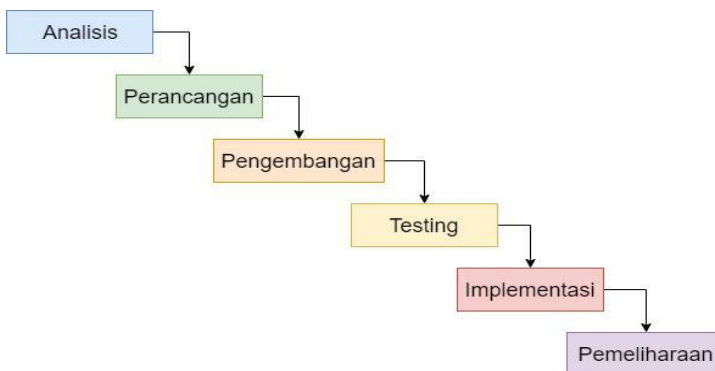
### 2.2.9 *Usability Testing*

*Usability Testing* adalah salah satu contoh proses dalam penelitian pengguna. Penelitian pengguna adalah studi

sistematis tentang tujuan, kebutuhan, dan kemampuan pengguna sehingga dapat menentukan desain, konstruksi, atau peningkatan alat untuk mendapatkan manfaat bagaimana pengguna bekerja dan hidup. *Usability Testing* tetap merupakan cara utama untuk menentukan apakah pengguna mencapai tujuan mereka ketika menggunakan aplikasi. Pengujian ini menggunakan banyak metrik yang sama dengan teknik penelitian pengguna lainnya (misalnya: Tingkat penyelesaian dapat ditemukan di mana saja). [2]

### 2.2.10 Software Development Life Cycle Waterfall

SDLC *Waterfall* adalah SDLC *sequential* dimana kebutuhan harus selesai sebelum dilanjutkan ke tahap selanjutnya. Pengujian pada *waterfall* dilakukan pada saat pengembangan telah selesai sepenuhnya dilakukan. Setiap tahap pada SDLC *waterfall* harus diselesaikan dalam jangka waktu tertentu dan setiap tahap harus memiliki dokumentasi atau keluaran [11]. Tahapan pada SDLC *waterfall* dapat divisualisasikan oleh **Gambar 2.3**.



**Gambar 2.3 Tahapan SDLC Waterfall**

Berdasarkan gambar tersebut terlihat bahwa tahapan SDLC *waterfall* berjumlah 6 tahap yaitu analisis, perancangan, pengembangan, *testing*, implementasi dan pemeliharaan. Tahapan tersebut harus dilaksanakan secara berurutan mulai

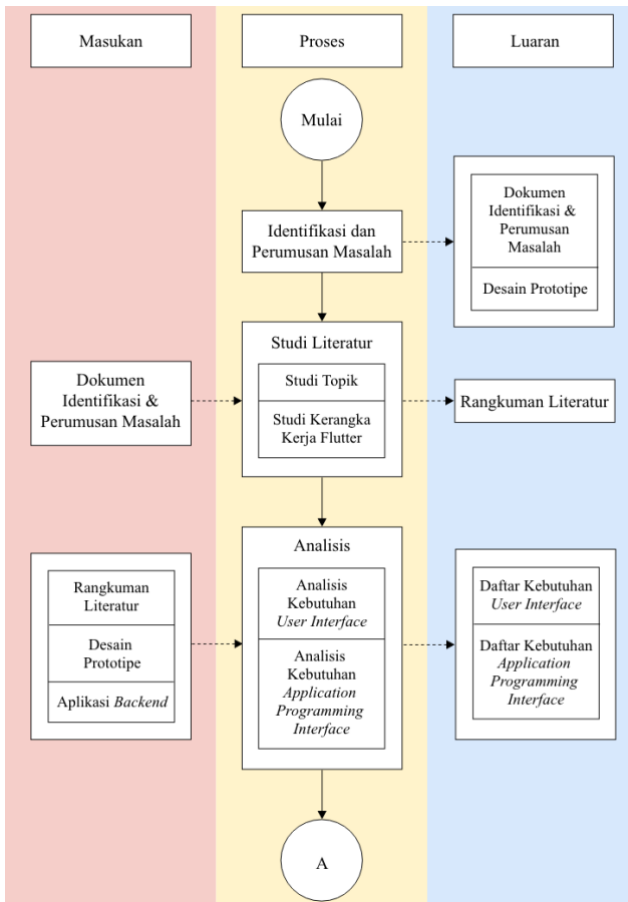
dari atas dan berjalan menurun bawah. Inilah yang menjadi pembeda siklus pengembangan perangkat lunak *waterfall* dengan siklus pengembangan yang lain. Tetapi pada penelitian ini hanya berfokus pada tahapan analisis, perancangan, pengembangan dan *testing*. Siklus ini memiliki kekurangan yaitu sulit untuk beradaptasi ketika terjadi banyak perubahan dalam proses pengembangan perangkat lunak.

*Halaman ini sengaja dikosongkan*

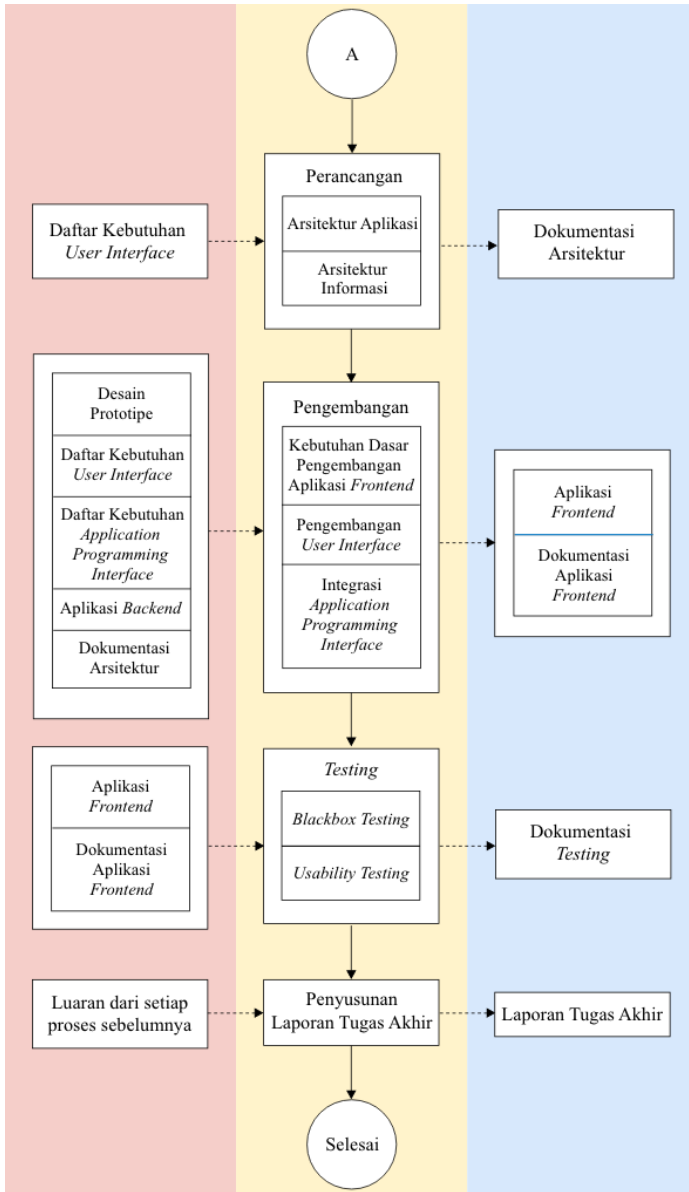
## BAB 3 METODOLOGI

Pada bab 3 akan dibahas mengenai metode apa yang akan digunakan dan tahap – tahap untuk melakukan penelitian ini. Berikut akan dijabarkan mengenai metode beserta tahap – tahap penelitian ini.

### 3.1 Diagram Metode Pelaksanaan Tugas Akhir



**Gambar 3.1 Diagram Metodologi 1**



Gambar 3.2 Diagram Metodologi 2



## **3.2 Uraian Metode Pelaksanaan Tugas Akhir**

### **3.2.1 Identifikasi dan Perumusan Masalah**

Tahap awal ini dimulai dengan melakukan identifikasi masalah melalui studi penelitian yang dilakukan oleh Bening pada tahun 2019 dan Kinasih pada tahun 2019 [1], [2]. Tahap ini dilakukan dengan tujuan untuk mengamati dan menemukan masalah yang akan diteliti dalam tugas akhir dan merumuskan masalah yang harus diselesaikan, batasan penelitian, tujuan penelitian, manfaat penelitian, dan relevansi. Luaran pada tahap ini adalah dokumen identifikasi masalah dan perumusan masalah serta desain prototipe sistem manajemen vendor.

### **3.2.2 Studi Literatur**

Pada tahap ini dilakukan dengan mengumpulkan informasi mengenai topik yang akan diangkat untuk memahami topik dan kerangka kerja yang digunakan untuk tugas akhir. Fase ini berguna untuk membangun pemahaman dasar dalam mengerjakan tugas akhir dan mencari penelitian sebelumnya pada topik terkait. Luaran pada tahap ini adalah rangkuman literatur.

#### **a. Studi Topik**

Studi topik dilakukan dengan melakukan studi pada literatur terkait topik tugas akhir dalam bentuk buku, *paper*, publikasi dan penelitian sebelumnya.

#### **b. Studi Kerangka Kerja Flutter**

Studi kerangka kerja dilakukan dengan melakukan studi literatur kerangka kerja Flutter yang digunakan pada tugas akhir ini dalam bentuk penelitian sebelumnya dan dokumentasi kerangka kerja.

### 3.2.3 Analisis

Tahap ini akan dilakukan penggalian kebutuhan mengenai aplikasi *frontend* yang akan dibuat. Tahap ini memerlukan masukan berupa desain prototipe sistem manajemen vendor dan akan menghasilkan keluaran berupa daftar kebutuhan antarmuka dan komponen Flutter.

#### a. Analisis Kebutuhan *User Interface*

Tahap ini adalah tahap menarik kebutuhan antarmuka dari desain prototipe. Tahap ini akan menghasilkan tabel yang menjelaskan daftar kebutuhan antarmuka. Kolom tabel tersebut terdiri *ID* kebutuhan antarmuka dan nama antarmuka.

#### b. Analisis Kebutuhan *Application Programming Interface*

Tahap ini adalah tahap menganalisis kebutuhan API. Pada satu kebutuhan antarmuka dapat dilayani oleh satu atau banyak API. Tahap ini akan menghasilkan tabel daftar kebutuhan komponen API dengan kolom yang terdiri dari *ID* API dan *endpoint* nya.

### 3.2.4 Perancangan

Menyusun perancangan perlu dilakukan untuk mempersiapkan rencana pekerjaan yang akan dilakukan berdasarkan dari hasil rancangan yang akan dicapai. Rancangan ini ini akan dipecah menjadi dua bagian kecil yang mudah untuk dikerjakan.

#### a. Arsitektur Aplikasi

Sebelum masuk ke perancangan arsitektur informasi, perlu adanya tahap perancangan arsitektur aplikasi yang berpedoman dengan arsitektur kerangka kerja Flutter. Tahap ini akan mempermudah proses pengembangan aplikasi *frontend* karena sudah diatur dengan baik letak

kode sumber setiap *user interface*. Proses pengembangan juga menjadi lebih cepat.

#### **b. Arsitektur Informasi**

Penulis pada bagian ini akan membuat arsitektur informasi yang menjadi dasar penulis dalam merancang setiap desain prototipe ke dalam bentuk aplikasi *frontend*. Arsitektur informasi akan menginformasikan bagaimana tampilan *user interface* aplikasi itu sebaiknya dibuat untuk memudahkan pemahaman pengguna ketika menggunakan aplikasi dan memudahkan pengembang dalam mengatur konten pada setiap *user interface*

### **3.2.5 Pengembangan**

Pada tahap ini, penulis akan melakukan pengembangan aplikasi *frontend* sistem manajemen vendor Giyomi.id berdasarkan desain prototipe penelitian sebelumnya, daftar kebutuhan *user interface* serta komponen API, dan aplikasi *backend*.

#### **a. Kebutuhan Dasar Pengembangan Aplikasi**

Tahap ini adalah tahapan yang pertama kali harus dilakukan di dalam fase pengembangan ini. Penulis akan menyiapkan beberapa kebutuhan yang digunakan selama pengembangan aplikasi *frontend*. Tujuan dilakukan tahap ini adalah untuk mempercepat dan mempermudah proses pengembangan.

#### **b. Pengembangan *User Interface***

Pengembangan ini akan menghasilkan semua antarmuka berdasarkan desain prototipe. Tahap ini akan memanfaatkan banyak sekali komponen yang sudah ada dalam kerangka kerja Flutter. Proses modifikasi terhadap komponen yang ada juga dilakukan untuk meningkatkan kesesuaian antarmuka aplikasi *frontend* dengan desain

prototipe. Selanjutnya juga akan dilakukan *routing* setiap antarmuka sehingga bisa terhubung satu sama lain dan bisa menghasilkan navigasi yang mulus.

### c. **Integrasi *Application Programming Interface***

Pada tahapan ini penulis melakukan integrasi antara aplikasi *frontend* dengan aplikasi *backend*. Integrasi yang dilakukan di sini adalah proses penggabungan setiap antarmuka dengan *endpoint* yang disediakan oleh aplikasi *backend*. Setiap *endpoint* harus bekerja secara mulus dengan antarmuka sehingga menghasilkan aplikasi yang berjalan sesuai fungsi yang diharapkan.

## 3.2.6 Testing

Pada tahap *testing* ini, sistem akan diuji dalam beberapa metode dengan tujuan untuk mengetahui kemampuan sistem dalam mengatasi masalah sesuai dengan fungsi yang diharapkan. Sistem diharapkan dapat menyelesaikan permasalahan yang terjadi pada *Giyomi.id*. Pengujian juga akan dilakukan dengan melibatkan pihak *Giyomi.id*.

### a. **Blackbox Testing**

Tahap ini dilakukan setelah aplikasi selesai dikembangkan. Aplikasi akan diuji dengan cara menjalankan skenario pengujian untuk memastikan bahwa fungsional aplikasi sudah dapat dijalankan secara keseluruhan. Dilakukan Blackbox Testing untuk setiap fungsional aplikasi yang dimodifikasi menyesuaikan kebutuhan untuk memastikan bahwa semua fungsi yang dikembangkan telah berjalan baik dan sesuai dengan proses bisnis *Giyomi.id*. Berikutnya akan dilakukan compatibility test dengan menjalankan aplikasi di system operasi Android dan Ios untuk memastikan aplikasi dapat berjalan dengan baik. Hasil pengujian akan didokumentasikan berupa dokumen Blackbox Testing dan dianalisa. Apabila terdapat fitur

yang tidak berjalan, maka proses kembali ke pengembangan aplikasi. Luaran dari proses ini adalah dokumentasi skenario serta hasil dari Blackbox testing.

**c. *Usability Testing***

Usability Testing dilakukan untuk melihat kualitas dari pengalaman yang dilakukan calon pengguna pada saat menggunakan aplikasi. Tahapan ini perlu dilakukan Untuk mengetahui terlebih dahulu kemungkinan kendala penggunaan aplikasi yang akan dihadapi oleh pengguna. Usability Testing dilakukan dengan memberikan aplikasi pada calon pengguna aplikasi dan meminta calon pengguna untuk melakukan tugas tertentu. Hasil observasi dari pelaksanaan tugas oleh calon pengguna akan didokumentasikan dalam dokumen Usability Test serta dianalisa seberapa mudah calon pengguna dalam mengenali dan menggunakan fungsional aplikasi. Penilaian dilakukan berdasarkan jumlah langkah serta konfigurasi yang diperlukan oleh pengguna dalam menyelesaikan tugas. Selain itu, kepuasan pengguna dinilai berdasarkan saran perbaikan yang diberikan oleh calon pengguna setelah melakukan rangkaian pengujian. Luaran dari proses ini adalah skenario usability test, hasil pengujian, serta saran perbaikan dari calon Pengguna.

**3.2.7 Penyusunan Laporan Tugas Akhir**

Penyusunan laporan tugas akhir merupakan tahap terakhir dari penelitian ini yang akan menghasilkan buku tugas akhir. Tahap penyusunan laporan tugas akhir dilakukan dokumentasi mengenai hasil dari semua proses penelitian. Dalam laporan ini akan dituliskan mengenai kesimpulan dan saran untuk dijadikan masukan pada penelitian yang akan datang.

*Halaman ini sengaja dikosongkan.*

## BAB 4 ANALISIS

Pada bab ini akan dilakukan analisis kebutuhan sebelum masuk ke fase perancangan aplikasi *frontend* dengan kerangka kerja Flutter.

### 4.1 Analisis Kebutuhan *User Interface*

Penulis telah mengumpulkan data kebutuhan *User Interface* berdasarkan desain prototipe. Terdapat 20 *User Interface* yang harus dilakukan pengembangannya dalam penelitian ini. Daftar kebutuhan *User Interface* ditampilkan secara lengkap di **Tabel 4.1**.

**Tabel 4.1 Kebutuhan *User Interface***

<b>ID User face</b>	<b>Nama <i>User Interface</i></b>
UI1	Login
UI2	Home
UI3	Vendor
UI4	AddVendor
UI5	VendorDetail
UI6	SPK
UI7	AddSPK
UI8	SPKDetail
UI9	SPKProgress
UI10	SPKReceived
UI11	SPKEvaluation
UI12	OngoingSPK

UI13	FinishedSPK
UI14	SPKDraft
UI15	SPKValidation
UI16	EditSPK
UI17	Account
UI18	AccountList
UI19	AddAccount
UI20	EditAccount

#### 4.2 Analisis Kebutuhan *Application Programming Interface* (API)

Data kebutuhan API telah penulis dapatkan dari aplikasi *backend* yang dikerjakan pada penelitian sebelumnya oleh Pahang. Terdapat 18 API yang akan digunakan selama pengemabangan aplikasi *frontend*. Setiap *User Interface* bisa memiliki lebihh dari satu API. Tabel kebutuhan API akan terlihat seperti **Tabel 3.2**

**Tabel 4.2 Kebutuhan API**

ID API	Endpoint
API1	<a href="https://giyomi.herokuapp.com/rest/auth/login">https://giyomi.herokuapp.com/ rest/auth/login</a>
API2	<a href="https://giyomi.herokuapp.com/rest/users">https://giyomi.herokuapp.com/ rest/users</a>
API3	<a href="https://giyomi.herokuapp.com/rest/spks">https://giyomi.herokuapp.com/ rest/spks</a>
API4	<a href="https://giyomi.herokuapp.com/">https://giyomi.herokuapp.com/</a>



	rest/cdn
API5	<a href="https://giyomi.herokuapp.com/restful/vendors">https://giyomi.herokuapp.com/ restful/vendors</a>
API6	<a href="https://giyomi.herokuapp.com/restful/spks">https://giyomi.herokuapp.com/ restful/spks</a>
API7	<a href="https://giyomi.herokuapp.com/restful/spkVarians">https://giyomi.herokuapp.com/ restful/spkVarians</a>
API8	<a href="https://giyomi.herokuapp.com/restful/spkProseses">https://giyomi.herokuapp.com/ restful/spkProseses</a>
API9	<a href="https://giyomi.herokuapp.com/restful/spkStatuses">https://giyomi.herokuapp.com/ restful/spkStatuses</a>
API10	<a href="https://giyomi.herokuapp.com/restful/koleksis">https://giyomi.herokuapp.com/ restful/koleksis</a>
API11	<a href="https://giyomi.herokuapp.com/restful/jenisKoleksis">https://giyomi.herokuapp.com/ restful/jenisKoleksis</a>
API12	<a href="https://giyomi.herokuapp.com/restful/variens">https://giyomi.herokuapp.com/ restful/variens</a>
API13	<a href="https://giyomi.herokuapp.com/restful/panjangKains">https://giyomi.herokuapp.com/ restful/panjangKains</a>
API14	<a href="https://giyomi.herokuapp.com/restful/satuanPanjangKains">https://giyomi.herokuapp.com/ restful/satuanPanjangKains</a>
API15	<a href="https://giyomi.herokuapp.com/restful/ukuranJumlahs">https://giyomi.herokuapp.com/ restful/ukuranJumlahs</a>
API16	<a href="https://giyomi.herokuapp.com/restful/ukuranJumlah">https://giyomi.herokuapp.com/ restful/ukuranJumlah</a>

	Diterimas
API17	<a href="https://giyomi.herokuapp.com/restful/catatans">https://giyomi.herokuapp.com/ restful/catatans</a>
API18	<a href="https://giyomi.herokuapp.com/restful/nilaispks">https://giyomi.herokuapp.com/ restful/nilaispks</a>

## BAB 5 PERANCANGAN

Pada bab ini dilakukan perancangan sebelum masuk ke fase pengembangan aplikasi *frontend* dengan kerangka kerja Flutter.

### 5.1 Perancangan Arsitektur Aplikasi

Pada bagian ini penulis memaparkan mengenai penyusunan arsitektur kerangka kerja Flutter yang akan digunakan selama pengembangan aplikasi *frontend*. Ada beberapa proses yang harus dilakukan, yaitu ada instalasi, pembuatan direktori, dan proses menjalankan aplikasi. Selama proses perancangan dan pengembangan, penulis menggunakan Sistem Operasi MAC OSX.

#### 5.1.1 Instalasi Kerangka Kerja

Proses instalasi kerangka kerja Flutter tergolong sangat mudah dan cepat. Hanya membutuhkan 3 langkah sebelum kerangka kerja siap digunakan. Pertama, penulis harus mengunduh kelengkapan *file* kerangka kerja dari situs resminya yaitu <https://flutter.dev/docs/get-started/>. Pengunduhan dilakukan dengan melakukan *clone* dengan GIT. Perintah untuk melakukan *clone* ditampilkan pada *terminal* yang ada pada **Gambar 5.1**.

```
MBP:~ alvianem$ git clone https://github.com/flutter/flutter.git
```

**Gambar 5.1 Perintah Unduh File Flutter**

Kedua, penulis menjalankan perintah di *terminal* yang berguna untuk membuat *PATH flutter* di simpan dan dikenali oleh sistem operasi. Perintah *flutter* akan selalu digunakan selama pengembangan aplikasi.

```
MBP:~ alvianem$ export PATH="$PATH:`pwd`/flutter/bin"
```

**Gambar 5.2 Perintah Simpan PATH Flutter**

Terakhir, setelah mengunduh file Flutter dan menyimpan *PATH* nya. Maka diperlukan perintah untuk melakukan pengecekan

bahwa kerangka kerja siap digunakan. Perintah ini akan menampilkan beberapa *checklist*, seperti kesiapan *Code Editor* untuk menulis kode sumber aplikasi, Android SDK untuk menjalankan aplikasi di sistem operasi Android, dan Xcode Command untuk menjalankan aplikasi versi IOS nya. Perintah yang penulis tulis ditampilkan pada **Gambar 5.3**.

```
MBP:~ alvianem$ flutter doctor
```

**Gambar 5.3 Perintah Pengecekan Kesiapan Flutter**

Hasil dari perintah pengecekan adalah seperti yang ditampilkan pada **Gambar 5.4**. Disini terlihat bahwa kerangka kerja Flutter telah siap digunakan dan memenuhi semua *checklist*.

```

[✓] Flutter (Channel stable, v1.12.13+hotfix.5, on Mac OS X 10.14.6 18G4032, locale en-ID)
[✓] Android toolchain - develop for Android devices (Android SDK version 29.0.2)
[✓] Xcode - develop for iOS and macOS (Xcode 11.3.1)
[✓] Android Studio (version 3.5)
[✓] VS Code (version 1.45.1)

```

**Gambar 5.4 Perintah Pengecekan Kesiapan Flutter**

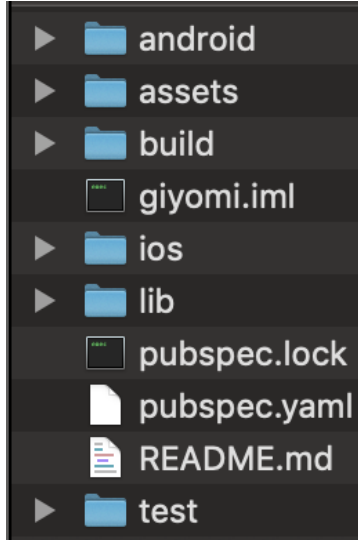
### 5.1.2 Struktur Direktori Kerangka Kerja

Proses instalasi telah berhasil dilakukan. Selanjutnya yang dibutuhkan adalah proses pembuatan folder aplikasi *frontend* yang berisi *file* kerangka kerja Flutter dan siap digunakan. Folder ini tidak akan muncul dengan sendirinya setelah instalasi kerangka kerja Flutter. Ada perintah sendiri yang harus dijalankan di *terminal* untuk memunculkan folder aplikasi. Perintah yang dijalankan penulis ada di **Gambar 5.5**. Dengan menekan tombol *enter* maka akan otomatis melakukan pembuatan folder.

```
MBP:~ alvianem$ flutter create giyomi
```

**Gambar 5.5 Perintah Pembuatan Direktori Aplikasi**

Direktori otomatis telah terbentuk dan sudah bisa masuk ke dalam direktori untuk melihat isi direktori. Struktur dari direktori ini ditampilkan di **Gambar 5.6**



**Gambar 5.6 Struktur Direktori Flutter**

Keterangan dari setiap isi direktori ditampilkan secara lengkap pada **Tabel 5.1**.

**Tabel 5.1 Keterangan Sub Folder**

No	Nama Sub Folder	Keterangan
1.	<i>android</i>	Berisi <i>file</i> yang akan digunakan oleh Flutter untuk mengkonversi kode sumber Dart menjadi kompatibel dengan sistem operasi Android
2.	<i>assets</i>	Digunakan untuk menyimpan <i>file</i> berupa gambar dan <i>font</i> yang akan digunakan pada aplikasi
3.	<i>build</i>	Berisi seluruh <i>file</i> kerangka kerja Flutter yang akan digunakan selama membuat aplikasi

4.	<i>ios</i>	Berisi <i>file</i> yang akan digunakan oleh Flutter untuk mengkonversi kode sumber Dart menjadi kompatibel dengan sistem operasi IOS
5.	<i>lib</i>	Berisi <i>source code</i> aplikasi Sistem Manajemen Vendor dengan bahasa Dart
6.	<i>test</i>	Berisi <i>file</i> yang akan dipakai Flutter untuk melakukan pengujian terhadap <i>Widget</i> yang dipakai penulis selama mengembangkan aplikasi

### 5.1.3 Menjalankan Kerangka Kerja

Menjalankan aplikasi dengan kerangka kerja Flutter dibutuhkan *simulator* yang berjalan di desktop. *Simulator* ada dua jenis, yaitu yang memiliki sistem operasi Android atau sistem operasi IOS. Untuk saat ini penulis akan menjalankan *Simulator* versi IOS dengan perintah seperti pada **Gambar 5.7**.

```
MBP:~ alvianem$ open -a simulator
```

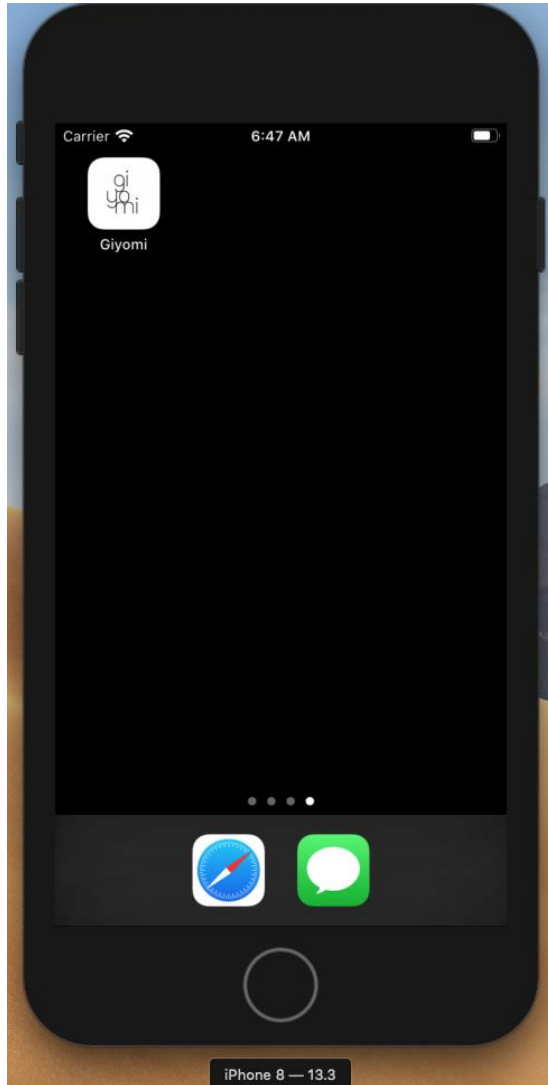
**Gambar 5.7 Perintah Menjalankan Simulator IOS**

Simulator akan otomatis terbuka setelah perintah nya dijalankan di *terminal*. Tetapi ini masih menjalankan *Simulator* saja, belum masuk ke proses menjalankan aplikasi. Untuk menjalankan aplikasi dibutuhkan perintah tersendiri. Perintah yang ditulis oleh Penulis ditampilkan pada **Gambar 5.8**

```
MBP:giyomi alvianem$ flutter run
```

**Gambar 5.8 Perintah Menjalankan Aplikasi**

Aplikasi yang berjalan di *Simulator* ditampilkan pada **Gambar 5.9**

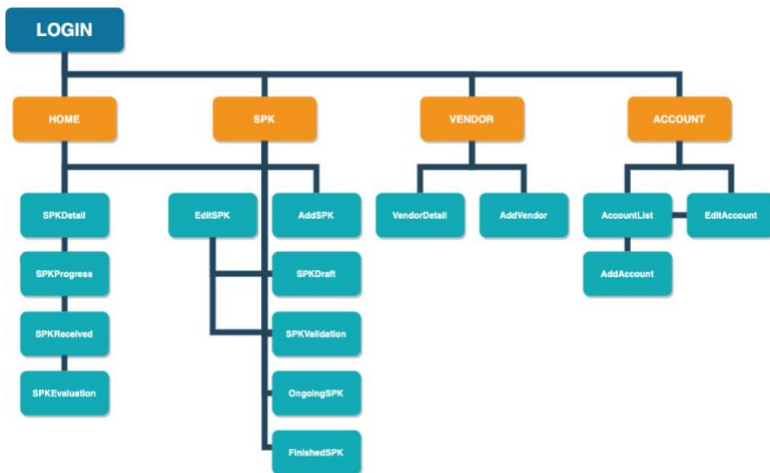


Gambar 5.9 Aplikasi Giyomi di Simulator IOS

## 5.2 Perancangan Arsitektur Informasi

Aplikasi *frontend* Sistem Manajemen Vendor terdiri dari 4 halaman induk yaitu Home, Account, SPK, dan Vendor. Masing-

masing dari halaman induk mempunyai sub halaman. Total dari sub halaman adalah sejumlah 15 dan jika ditambah dengan halaman Login maka total halaman yang dibuat adalah sejumlah 20 halaman. Untuk mempermudah proses pengembangan aplikasi, seluruh halaman dibuat dengan menggunakan arsitektur pada **Gambar 5.1**. Selain mempermudah proses pengembangan, penggunaan arsitektur ini akan memudahkan penulis pula untuk proses *debugging* dan mempercepat proses pengembangan aplikasi.



**Gambar 5.10** Arsitektur Informasi



## BAB 6 PENGEMBANGAN APLIKASI

Pada bab pengembangan ini penulis akan menjabarkan terkait pengembangan aplikasi *frontend* yang sesuai dengan analisis kebutuhan yang telah dijabarkan oleh penulis pada bab 4 buku tugas akhir ini.

### 6.1 Kebutuhan Dasar Pengembangan

Pada tahapan ini disiapkan kebutuhan dasar berupa sejumlah *tools* maupun *platform* untuk membantu dan mendukung proses pengembangan aplikasi yang memiliki kerangka kerja Flutter agar bisa dikembangkan dengan lebih mudah dan cepat. Kebutuhan dasar yang perlu disiapkan dapat dilihat pada **Error! Reference source not found.**

**Tabel 6.1 Tools dan Platform yang digunakan**

Nama Tools dan Platform	Kegunaan
Mac OS	Sistem operasi untuk menjalankan <i>simulator</i> aplikasi <i>frontend</i> yang berbasis Android dan IOS
Visual Studio Code	<i>Code Editor</i> untuk menuliskan kode selama pengembangan aplikasi berlangsung
Pub Dev	<i>Package Manager</i> yang berisi kumpulan <i>Flutter Library</i> untuk mempermudah proses pengembangan
Emulator Android & IOS	<i>Simulator</i> untuk menjalankan aplikasi melalui <i>desktop</i>

### 6.2 Pengembangan User Interface

Pada tahapan ini penulis melakukan pengembangan *User Interface* aplikasi *frontend* yang menggunakan *Widget* kerangka kerja *Flutter* dan berdasarkan prototipe penelitian sebelumnya sebagaimana hasilnya dilampirkan secara lengkap pada **LAMPIRAN A**. Pengembangan aplikasi dibuat berdasarkan arsitektur informasi yang telah dijelaskan pada bab 4 buku ini. Flutter mempunyai dokumentasi lengkap pengembangan *User*

*Interface* yang tertera di halaman resminya (<https://flutter.dev/docs/development/ui/widgets-intro>).

### 6.2.1 Pengembangan Komponen Input

Komponen Input sangat dibutuhkan oleh aplikasi *frontend* di banyak *User Interface*. Salah satu nya adalah yang paling awal dibuka oleh pengguna, yaitu Login. Pengguna harus bisa memasukkan *username* maupun *password* nya untuk masuk ke dalam aplikasi sehingga bisa menggunakan fitur yang disediakan. Flutter memberikan kemudahan untuk membuat komponen ini dengan menggunakan *Widget* yang bernama *TextField*. *Widget* ini memiliki kemampuan kustomisasi yang yang banyak, salah satunya adalah memberikan dekorasi sehingga lebih enak di pandang oleh Pengguna dengan menggunakan *InputDecoration*. Untuk penggunaan *TextField* di dalam aplikasi bisa dilihat pada **Gambar 6.1**.

```
return Column(
  crossAxisAlignment: CrossAxisAlignment.start,
  children: <Widget>[
    SizedBox(height: 10.0),
    Container(
      alignment: Alignment.centerLeft,
      decoration: BoxDecoration(
        color: Color(0xFFEC407A),
        borderRadius: BorderRadius.circular(10.0),
        boxShadow: [
          BoxShadow(
            color: Colors.black12,
            blurRadius: 6.0,
            offset: Offset(0, 2),
          ), // BoxShadow
        ],
      ), // BoxDecoration
      height: 55.0,
      child: TextField(
        keyboardType: TextInputType.text,
        style: TextStyle(
          color: Colors.white,
          fontFamily: 'OpenSans',
        ), // TextStyle
        decoration: InputDecoration(
          border: InputBorder.none,
          contentPadding: EdgeInsets.only(top: 14.0),
          prefixIcon: Icon(
            Icons.person,
            color: Colors.white,
          ), // Icon
          hintText: 'Username',
          hintStyle: TextStyle(
            color: Colors.white54,
            fontFamily: 'OpenSans',
          ),
        ),
      ),
    ),
  ],
);
```

**Gambar 6.1** Kode Sumber Komponen Input

InputDecoration membuat komponen TextField bisa dikustomisasi sesuai keinginan, salah satunya adalah untuk memberikan warna di bagian tepi komponen. Hasil dari pembuatan komponen ini ditampilkan pada **Gambar 6.2**.



**Gambar 6.2** Tampilan Komponen Input

Komponen Input yang penulis buat tidak hanya digunakan di *User Interface* Login saja, tetapi juga di *User Interface* yang lain seperti yang ditampilkan pada **Tabel 6.2**.

**Tabel 6.2** Penggunaan Komponen Input di User Interface

Nama Komponen	User Interface yang Menggunakan
Input	UI1, UI4, UI7, UI9, UI10, UI11, UI16, UI19, UI20

### 6.2.2 Pengembangan Komponen Button

Pengguna membutuhkan tombol ketika ingin masuk ke dalam aplikasi. *Widget* yang digunakan oleh penulis untuk membuat komponen Button bernama RaisedButton. Untuk penggunaan RaisedButton di dalam aplikasi bisa dilihat pada **Gambar 6.3**

```
return Container(
  padding: EdgeInsets.symmetric(
    vertical: 40.0,
  ), // EdgeInsets.symmetric
  width: double.infinity,
  child: RaisedButton(
    elevation: 5.0,
    onPressed: () {
      navigateToHomeScreen(context);
    },
    padding: EdgeInsets.all(15.0),
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(10.0),
    ), // RoundedRectangleBorder
    color: Colors.white,
    child: Text(
      'MASUK',
      style: TextStyle(
        color: Color(0xFFE91E63),
        letterSpacing: 1.0,
        fontSize: 18.0,
        fontWeight: FontWeight.bold,
        fontFamily: 'OpenSans',
      ),
    ),
  ),
);
```

**Gambar 6.3** Tampilan Komponen Button

RaisedButton akan menampilkan animasi ketika pengguna menekan tombol ini sehingga pengguna lebih memilih *Widget* ini daripada yang lain. Hasil dari pembuatan komponen ini ditampilkan pada **Gambar 6.4**.



**Gambar 6.4** Tampilan Komponen Button

Komponen Button digunakan di banyak *User Interface* seperti yang ditampilkan pada **Tabel 6.3**.

**Tabel 6.3** Penggunaan Komponen Button di User Interface

Nama Komponen	User Interface yang Menggunakan
Button	UI1, UI4, UI7, UI9, UI10, UI11, UI16, UI17, UI19, UI20

### 6.2.3 Pengembangan Komponen Counter

Pada *User Interface* Beranda terdapat komponen Counter untuk menampilkan jumlah SPK Berjalan, SPK Selesai, Draf SPK, dan SPK yang memerlukan validasi. *Widget* yang digunakan oleh penulis untuk membuat komponen Button bernama RaisedButton. Penulis menggunakan *Widget* ini karena Counter ini juga bisa di tekan oleh Pengguna untuk masuk ke *User Interface* lain. Selain RaisedButton juga terdapat *Widget* Text untuk menampilkan nama dari masing-masing Counter. Masing-masing dari komponen Counter akan dimasukkan ke dalam Container sehingga membuat tampilan menjadi lebih rapi. Di dalam Container akan diberikan kustomisasi bernama *EdgeInsets* untuk memberikan jarak antara teks dengan ujung tepi tombol Untuk penggunaan RaisedButton untuk komponen Counter bisa dilihat pada **Gambar 6.5**

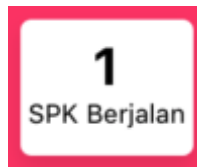
```

Container(
  width: MediaQuery.of(context).size.width / 4,
  padding: EdgeInsets.only(right: 2.0, left: 8.0, top: 6.0),
  child: RaisedButton(
    padding: EdgeInsets.all(5),
    elevation: 2.5,
    color: Colors.white,
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(5.0)), // RoundedRectangleBorder
    child: Column(
      mainAxisAlignment: MainAxisAlignment.center,
      children: <Widget>[
        Text(
          _spkBerjalan,
          style: TextStyle(
            color: Colors.black,
            fontSize: 20.0,
            fontWeight: FontWeight.bold), // TextStyle
        ), // Text
        SizedBox(
          height: 1,
        ), // SizedBox
        Text(
          'SPK Berjalan',
          style: TextStyle(
            color: Colors.black, fontSize: 11.0), // TextStyle
        ), // Text
      ], // <Widget>[]
    ), // Column
    onPressed: () =>
      navigateToOngoingWorkOrderScreen(context)
  )
)

```

**Gambar 6.5** Kode Sumber Komponen Counter

Hasil dari pembuatan komponen Counter ini ditampilkan pada **Gambar 6.6**.



**Gambar 6.6** Tampilan Komponen Counter

Komponen Counter hanya digunakan pada satu *User Interface* seperti yang ditampilkan pada **Tabel 6.4**.

**Tabel 6.4** Penggunaan Komponen Counter di User Interface

Nama Komponen	User Interface yang Menggunakan
Button	UI2

### 6.2.4 Pengembangan Komponen Progress

Komponen Progress digunakan untuk menampilkan sejauh mana SPK sudah menyelesaikan proses Cutting, Jahit, dan Finishing. *Widget* yang digunakan oleh penulis untuk membuat komponen Progress bernama `IntervalProgressBar`. Dokumentasi `IntervalProgressBar` tertera secara lengkap di halaman ini (<https://pub.dev/packages/intervalprogressbar>). Penulis menggunakan *Widget* ini karena mudah dalam pemakaiannya. Penulis bisa menentukan berapa jumlah interval sekaligus warna nya langsung dalam *Widget* ini. Untuk penggunaan `IntervalProgressBar` untuk komponen Progress bisa dilihat pada **Gambar 6.7**

```
return Padding(
  padding: EdgeInsets.only(right: 10),
  child: IntervalProgressBar(
    direction: IntervalProgressDirection
      .horizontal,
    max: 3,
    progress: i,
    intervalSize: 3,
    size: Size(200, 7),
    highlightColor: Colors.grey,
    defaultColor: Colors.redAccent[700],
    intervalColor: Colors.white,
    intervalHighlightColor:
      Colors.white,
    reverse: true,
    radius: 5)); // IntervalProgressBar // Padding
```

**Gambar 6.7** Kode Sumber Komponen Progres

Komponen Progress akan merubah warna interval nya sesuai dengan ketepatan waktu SPK. Jika telat maka akan berwarna merah, dan ketika tepat waktu akan otomatis berubah berwarna biru. Hasil dari pembuatan komponen ini ditampilkan pada **Gambar 6.8**.



**Gambar 6.8** Tampilan Komponen Progress

Komponen Progress digunakan di dua *User Interface* seperti yang ditampilkan pada **Tabel 6.5**.

**Tabel 6.5 Penggunaan Komponen Button di User Interface**

Nama Komponen	User Interface yang Menggunakan
Progress	UI1, UI9

### 6.2.5 Pengembangan Komponen Card

Komponen Card digunakan untuk membuat tampilan daftar SPK dan Vendor. Komponen dimasukkan kedalam *Widget* ListView sehingga bisa tampil dalam bentuk daftar yang dinamis dan bisa di urut. *Widget* yang digunakan oleh penulis untuk membuat komponen Card memiliki nama yang sama dengan komponen yaitu Card. Untuk pembuatan komponen Card bisa dilihat pada **Gambar 6.9**

```
return GestureDetector(
  child: Card(
    margin: EdgeInsets.all(8),
    elevation: 2.5,
    child: InkWell(
      onTap: () =>
        navigateToProductDetailScreen(context, spks[index]['id'], koleksis[index]['id'], 8),
      child: Container(
        padding: EdgeInsets.all(16),
        child: Row(
          children: <Widget>[
            Column(
              crossAxisAlignment:
                CrossAxisAlignment.start,
              children: <Widget>[
                Row(
                  children: <Widget>[
                    Text(
                      spks[index],
                      style: TextStyle(
                        fontSize: 12,
                        fontWeight:
                          FontWeight.w400,
                        color: Colors.pink,
                        decoration:
                          TextDecoration
                            .underline), // TextStyle
                    ), // Text
                    SizedBox(
                      width: 145,
                    ), // SizedBox
                    _buildStatus(statuses[index]
                      ['statusId'])
                  ], // <Widget>[]
                ), // Row
              ], // Column
            ), // Row
          ], // <Widget>[]
        ), // Container
      ), // InkWell
    ), // Card
  ), // GestureDetector
```

**Gambar 6.9 Kode Sumber Komponen Card**

Komponen Card juga mengandung *Widget InkWell* sehingga ketika pengguna menekan komponen ini bisa memunculkan efek *splash*. Hasil dari pembuatan komponen ini ditampilkan pada **Gambar 6.10**.



**Gambar 6.10** Tampilan Komponen Card

Komponen Progress digunakan di dua *User Interface* seperti yang ditampilkan pada **Tabel 6.6**.

**Tabel 6.6** Penggunaan Komponen Card di User Interface

Nama Komponen	User Interface yang Menggunakan
Card	UI2, UI3, UI5, UI6, UI12, UI13, UI14, UI15, UI18

### 6.2.6 Pengembangan Komponen Sort

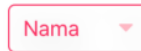
Pengguna ketika melakukan sortir daftar SPK maupun Vendor akan menggunakan Komponen Sort. Widget yang penulis gunakan untuk membuat komponen ini bernama *OutlineButton*. Sengaja penulis bedakan dengan Komponen Button yang menggunakan *RaisedButton*. Tujuannya adalah agar bisa membedakan antara satu sama lain. Untuk pembuatan komponen Sort bisa dilihat pada **Gambar 6.11**

```
child: OutlineButton(
  color: Colors.white,
  borderSide: BorderSide(
    color: Colors.pink,
    style: BorderStyle.solid,
    width: 1.0,
  ), // BorderSide
  shape: RoundedRectangleBorder(
    borderRadius: BorderRadius.circular(5.0)),
  child: Row(
    mainAxisAlignment: MainAxisAlignment.center,
    children: <Widget>[
      Text(
        'Nama',
        style: TextStyle(color: Colors.pink),
      ), // Text
      Icon(
        Icons.arrow_drop_down,
        color: Colors.pink,
        size: 20.0,
```

**Gambar 6.11** Kode Sumber Komponen Sort



Komponen Sort juga mengandung *Widget Icons* karena ada tampilan ikon yaitu anak panah yang menghadap ke bawah sehingga pengguna tahu kalau komponen Sort digunakan untuk menyortir daftar. Hasil dari pembuatan komponen ini ditampilkan pada **Gambar 6.12**.



**Gambar 6.12 Tampilan Komponen Sort**

Ada beberapa *User Interface* yang menggunakan komponen Sort, yaitu seperti yang ditampilkan pada **Tabel 6.7**.

**Tabel 6.7 Penggunaan Komponen Card di User Interface**

Nama Komponen	User Interface yang Menggunakan
Sort	UI3, UI5, UI6

### 6.2.7 Pengembangan Komponen Rating

Komponen Rating digunakan pengguna untuk menampilkan nilai suatu SPK atau Vendor. Komponen ini juga berguna untuk menilai SPK di *User Interface* SPK Evaluation. Widget yang penulis gunakan untuk membuat komponen ini bernama *SmoothStarRating*. Dokumentasi *SmoothStarRating* tertera di halaman ini ([https://pub.dev/packages/smooth\\_star\\_rating](https://pub.dev/packages/smooth_star_rating)). Untuk pembuatan komponen bisa dilihat pada **Gambar 6.13**

```
Row(children: <Widget>|
  SmoothStarRating(
    allowHalfRating:
      true,
    starCount: 5,
    rating: _rating,
    size: 25.0,
    filledIconData:
      Icons.star,
    halfFilledIconData:
      Icons.star_half,
    color:
      Colors.yellow,
    borderColor:
      Colors.yellow,
    spacing: 0.0), // SmoothStarRating
  SizedBox(
    width: 4,
  ), // SizedBox
  Text(
    _rating.toString(),
    style: TextStyle(
      fontSize: 16,
      fontWeight:
        FontWeight
          .bold), // TextStyle
```

**Gambar 6.13 Kode Sumber Komponen Rating**

Hasil dari pembuatan komponen Rating ditampilkan pada **Gambar 6.14**.



**Gambar 6.14** Tampilan Komponen Rating

*User Interface* yang menggunakan komponen Rating ditampilkan pada **Tabel 6.8**.

**Tabel 6.8** Penggunaan Komponen Rating di User Interface

Nama Komponen	User Interface yang Menggunakan
Rating	UI3, UI5, UI11

### 6.2.8 Pengembangan Komponen Image

Komponen Image digunakan untuk menampilkan segala macam bentuk gambar atau foto di dalam aplikasi. *Widget* yang penulis gunakan untuk membuat komponen ini bernama `ClipRRect` dan `Image`. `ClipRRect` berguna untuk membuat gambar memiliki bentuk atau sudut lingkaran. Untuk pembuatan komponen bisa dilihat pada **Gambar 6.15**

```
ClipRRect(
  borderRadius: BorderRadius.circular(25.0),
  child: Image.asset(
    items[index].img,
    width: 50,
  ), // Image.asset
), // ClipRRect
```

**Gambar 6.15** Kode Sumber Komponen Image

Hasil dari pembuatan komponen Image ditampilkan pada **Gambar 6.16**.



**Gambar 6.16** Tampilan Komponen Image

Hampir semua *User Interface* yang ada di aplikasi menggunakan komponen Image, seperti yang ditampilkan pada **Tabel 6.9**.

**Tabel 6.9 Penggunaan Komponen Image di User Interface**

Nama Komponen	User Interface yang Menggunakan
Image	UI1, UI4, UI5, UI6, UI7, UI8, UI9, UI10, UI11, UI12, UI13, UI14, UI15, UI16, UI17, UI19, UI20

### 6.2.9 Pengembangan Komponen AddMore

Komponen AddMore digunakan oleh pengguna sebagai tombol untuk menambahkan komponen Input yang baru. *Widget* yang penulis gunakan untuk membuat komponen ini bernama `FloatingActionButton`. Komponen ini sangat mudah digunakan dan praktis. Untuk pembuatan komponen bisa dilihat pada **Gambar 6.17**

```
floatingActionButton: FloatingActionButton(
  onPressed: () {
    addMore(context);
  },
  child: Icon(Icons.add),
  backgroundColor: Colors.pink,
) // FloatingActionButton
```

**Gambar 6.17 Kode Sumber Komponen AddMore**

Hasil dari pembuatan komponen AddMore ditampilkan pada **Gambar 6.18**.



**Gambar 6.18 Tampilan Komponen AddMore**

Komponen ini digunakan di *User Interface* AddSPK seperti yang ditampilkan pada **Tabel 6.10**.

**Tabel 6.10 Penggunaan Komponen AddMore di User Interface**

Nama Komponen	User Interface yang Menggunakan
AddMore	UI7

### 6.2.10 Pengembangan Komponen Dropdwon

Komponen Dropdown memiliki tampilan yang sejenis dengan Komponen Input. Tetapi komponen ini digunakan pengguna untuk memasukkan data dengan memilih terlebih dahulu. *Widget* yang penulis gunakan untuk membuat komponen ini bernama `DropDownButton`. Untuk pembuatan komponen bisa dilihat pada **Gambar 6.19**

```
child: DropDownButton(
  hint: Text("Pilih Vendor..."),
  isExpanded: true,
  value: _valVendor,
  items: _dataVendor.map((item) {
    return DropDownMenuItem(
      child: Text(item['nama']),
      value: item['id'],
    ); // DropDownMenuItem
  }).toList(),
  onChanged: (value) {
    setState(() {
      _valVendor = value;
    });
  },
),
```

**Gambar 6.19** Kode Sumber Komponen Dropdown

Hasil dari pembuatan komponen Dropdown ditampilkan pada **Gambar 6.20**.



**Gambar 6.20** Tampilan Komponen Dropdown

Komponen ini digunakan di *User Interface* seperti yang ditampilkan pada **Tabel 6.11**.

**Tabel 6.11** Penggunaan Komponen Dropdown di User Interface

Nama Komponen	User Interface yang Menggunakan
Dropdown	UI7, UI16

### 6.2.11 Pengembangan Komponen Score

Komponen Score dibuat oleh penulis dengan memanfaatkan *Widget* yang bernama `CircularPercentIndicator`. Untuk dokumentasi `CircularPercentIndicator` tertera secara lengkap di halaman ini ([https://pub.dev/packages/percent\\_indicator](https://pub.dev/packages/percent_indicator)). Pembuatan komponen bisa dilihat pada **Gambar 6.21**

```
CircularPercentIndicator(
  radius: 120.0,
  lineWidth: 13.0,
  animation: true,
  percent: 0.0,
  center: new Text(
    _skorVendor,
    style: new TextStyle(
      fontWeight: FontWeight.bold, fontSize: 20.0),
  ), // Text
  circularStrokeCap: CircularStrokeCap.round,
  backgroundColor: Colors.grey,
  progressColor: Colors.yellow,
```

**Gambar 6.21** Kode Sumber Komponen Score

Hasil dari pembuatan komponen Score ditampilkan pada **Gambar 6.22**.



**Gambar 6.22** Tampilan Komponen Score

Komponen ini digunakan di *User Interface* yang ditampilkan pada **Tabel 6.12**.

**Tabel 6.12** Penggunaan Komponen Score di User Interface

Nama Komponen	User Interface yang Menggunakan
Score	UI5

### 6.2.12 Pengembangan Komponen Table

Komponen Table dibuat oleh penulis dengan memanfaatkan *Widget* yang bernama `Table` dan `TableRow`. Keduanya

dimanfaatkan secara bersamaan ketika ingin membuat komponen yang berbentuk Tabel dimana ada kolom dan baris di dalamnya. Untuk pembuatan komponen bisa dilihat pada **Gambar 6.23**

```

Container(
  margin: EdgeInsets.only(left: 23, right: 7, top: 6),
  child: Table(
    border: TableBorder.all(),
    columnWidths: {
      0: FractionColumnWidth(.3),
      1: FractionColumnWidth(.1),
      2: FractionColumnWidth(.1),
      3: FractionColumnWidth(.1),
      4: FractionColumnWidth(.1),
      5: FractionColumnWidth(.1),
      6: FractionColumnWidth(.2)
    },
  ),
  children: [
    TableRow(children: [
      Container(
        color: Colors.lightBlue,
        child: Column(children: [
          SizedBox(
            height: 4,
          ), // SizedBox
          Text('Varian', style: TextStyle(color: Colors.white)),
          SizedBox(
            height: 4,
          ), // SizedBox
        ]), // Column
    ]), // Container
  ],
)

```

**Gambar 6.23** Kode Sumber Komponen Table

Hasil dari pembuatan komponen Table ditampilkan pada **Gambar 6.24**.

Varian	XS	S	M	L	XL	XXL	Total
Hijau	1	1	2	1	0	0	5
Biru	1	2	1	3	1	2	10
Kuning	1	1	1	0	2	0	5

**Gambar 6.24** Tampilan Komponen Table

Komponen ini digunakan hanya digunakan di dua *User Interface* yang ditampilkan pada **Tabel 6.13**.

**Tabel 6.13 Penggunaan Komponen Table di User Interface**

Nama Komponen	User Interface yang Menggunakan
Table	UI2, UI3, UI6, UI17

### 6.2.13 Pengembangan Komponen Header

Komponen Header dibuat oleh penulis dengan memanfaatkan *Widget* yang AppBar. Untuk pembuatan komponen bisa dilihat pada **Gambar 6.25**

```

appBar: AppBar(
  title: Text('SPK'),
  backgroundColor: Colors.pink,
  actions: <Widget>[
    IconButton(
      icon: Icon(Icons.add_circle),
      onPressed: () {
        navigateToAddSPKScreen(context);
      },
    ), // IconButton
  ], // <Widget>[]
), // AppBar

```

**Gambar 6.25 Kode Sumber Komponen Header**

Hasil dari pembuatan komponen Table ditampilkan pada **Gambar 6.26**.



**Gambar 6.26 Tampilan Komponen Header**

Komponen Header hampir digunakan di semua *User Interface*. Komponen ini sering dipakai karena hampir semua *User Interface* membutuhkan tampilan Header. *User Interface* yang membutuhkan komponen ini ditampilkan pada **Tabel 6.14**.

**Tabel 6.14 Penggunaan Komponen Header di User Interface**

Nama Komponen	User Interface yang Menggunakan
Header	UI2, UI3, UI4, UI5, UI6, UI7, UI8, UI9, UI10, UI11, UI12, UI13, UI14, UI15, UI16, UI17, UI18, UI19, UI20

### 6.2.14 Pengembangan Komponen BottomTab

Komponen Bottom dibuat oleh penulis dengan memanfaatkan *Widget* yang `BottomNavigationBar`. *Widget* ini memiliki pilihan untuk memilih tipe dengan `BottomNavigationBarType`. Untuk item yang ada di dalam komponen ini dimasukkan dalam *Widget* `BottomNavigationBarItem`. Untuk pembuatan komponen bisa dilihat pada **Gambar 6.27**

```
bottomNavigationBar: BottomNavigationBar(
  type: BottomNavigationBarType.fixed,
  currentIndex: _selectedTab,
  selectedItemColor: Colors.pink,
  onTap: (int index) {
    setState(() {
      _selectedTab = index;
    });
  },
  items: [
    BottomNavigationBarItem(
      icon: Icon(Icons.home),
      title: Text('Beranda'),
    ), // BottomNavigationBarItem
    BottomNavigationBarItem(
      icon: Icon(Icons.folder),
      title: Text('SPK'),
    ), // BottomNavigationBarItem
    BottomNavigationBarItem(
      icon: Icon(Icons.people),
      title: Text('Vendor'),
    ), // BottomNavigationBarItem
    BottomNavigationBarItem(
      icon: Icon(Icons.settings),
      title: Text('Akun'),
    ),
  ],
)
```

**Gambar 6.27** Kode Sumber Komponen BottomTab

Hasil dari pembuatan komponen BottomTab ditampilkan pada **Gambar 6.28**.



**Gambar 6.28** Tampilan Komponen BottomTab



Komponen BottomTab digunakan di empat *User Interface* utama. *User Interface* yang membutuhkan komponen ini ditampilkan pada **Tabel 6.15**.

**Tabel 6.15 Penggunaan Komponen BottomTab di User Interface**

Nama Komponen	User Interface yang Menggunakan
BottomTab	UI2, UI3, UI4, UI5, UI6, UI7, UI8, UI9, UI10, UI11, UI12, UI13, UI14, UI15, UI16, UI17, UI18, UI19, UI20

### 6.2.15 Navigasi *User Interface*

Navigasi dibutuhkan ketika pengguna melakukan perpindahan antar *User Interface*. Setiap *User Interface* di dalam aplikasi membutuhkan komponen ini sehingga pengguna tidak terjebak di dalam satu *User Interface* saja. Kerangka kerja Flutter memiliki *Widget* yang bernama Navigator. Flutter mempunyai dokumentasi lengkap Navigator yang tertera di halaman resminya (<https://api.flutter.dev/flutter/widgets/Navigator-class.html>). *Widget* ini memerlukan *Widget* lain sehingga dapat digunakan. Pertama adalah *PageRouteBuilder* untuk menentukan pengguna menuju ke *User Interface* yang mana dan *FadeTransition* untuk memberikan efek pergatian *User Interface* yang menarik. Untuk pemanfaatan *Widget* ini bisa di lihat di dalam *function* yang ada pada **Gambar 6.29**

```
Future navigateToAddVendorScreen(context) async {
  Navigator.of(context)
    .push(new PageRouteBuilder(pageBuilder: (BuildContext context, __, ___) {
      return new AddVendorScreen();
    }, transitionsBuilder: (_, Animation<double> animation, ___, Widget child) {
      return new FadeTransition(opacity: animation, child: child);
    })); // PageRouteBuilder
}
```

**Gambar 6.29 Kode Sumber Navigasi**

*Function* yang bernama `navigateToAddVendorScreen` harus dipanggil terlebih dahulu di dalam komponen lain sehingga bisa digunakan. *Function* ini bisa digunakan di berbagai komponen, salah satunya adalah di Komponen Header seperti pada **Gambar 6.30**

```
actions: <Widget>[
  IconButton(
    icon: Icon(Icons.add_circle),
    onPressed: () {
      navigateToAddVendorScreen(context);
    },
  ),
]
```

Gambar 6.30 Kode Sumber Penggunaan Navigasi

## 6.3 Integrasi API

Pada tahapan ini penulis melakukan proses integrasi antara *User Interface* yang sudah dikembangkan dengan API yang tersedia sehingga setiap fitur di dalam aplikasi bisa berjalan. Flutter mempunyai dokumentasi lengkap integrasi API yang tertera di halaman resminya (<https://flutter.dev/docs/development/data-and-backend>). Proses integrasi ini dibuat berdasarkan daftar *endpoint* API yang telah dijelaskan pada bab 3 buku ini. Sedangkan hasil dari integrasi API bisa dilihat berdasarkan Blackbox Testing di **LAMPIRAN B**.

### 6.3.1 Konfigurasi API

API pada kerangka kerja Flutter tidak bisa digunakan secara langsung. Dibutuhkan konfigurasi terlebih dahulu sebelum pemakaian. Pertama adalah lakukan *import package* di setiap antarmuka. Ada dua *package* yang harus di *import*, yaitu *convert* dan *http*. *Convert* digunakan untuk mengkonversi data ke dalam maupun dari format JSON. Sedangkan *http* digunakan untuk melakukan *request* ke *endpoint* API yang sudah tersedia. Proses *import* bisa dilihat pada **Gambar 6.31**

```
import 'dart:convert';
import 'package:http/http.dart' as http;
```

Gambar 6.31 Kode Sumber Konfigurasi Package API

Setelah melakukan *import*, dibutuhkan beberapa Header sehingga API bisa digunakan. Pertama adalah *Content-Type*, digunakan untuk memberikan label ke data bahwa data mempunyai format JSON. Ini dibutuhkan untuk mempermudah

API dan Server mengetahui jenis data yang di kirim. Kedua adalah *Authorization*, digunakan untuk menaruh *Secret Key* sehingga API yang di *request* tidak ditolak oleh sistem keamanan dari Server. Header di dalam kerangka kerja Flutter bisa dilihat pada **Gambar 6.32**.

```
import 'dart:convert';
import 'package:http/http.dart' as http;
```

**Gambar 6.32** Kode Sumber Konfigurasi Header API

### 6.3.2 Penggunaan API

Penggunaan API dengan memanfaatkan *package* http, terdapat berbagai macam metode. Penulis hanya membutuhkan tiga metode yaitu *POST*, *PUT*, *GET*. Metode *POST* penulis gunakan untuk mengirim data dari aplikasi *frontend* ke dalam server sehingga bisa di simpan di server. Metode ini membutuhkan *body* untuk menggunakan API. *Body* berisikan data yang akan di simpan ke server dengan format JSON. Karena data dari Flutter tidak menggunakan format JSON, dibutuhkan konversi dengan menggunakan fungsi *encode*. Penggunaan *endpoint* API dengan metode *POST* di dalam aplikasi bisa dilihat pada **Gambar 6.33**

```
final response = await http.post(
  'https://giyomi.herokuapp.com/restful/vendors',
  headers: headers,
  body: json.encode(body));
if (response.statusCode == 200) {
  return true;
} else {
  return false;
}
```

**Gambar 6.33** Kode Sumber Penggunaan API POST

Metode *PUT* dan *GET* penggunaanya di kerangka kerja Flutter sama seperti metode *POST*. Perbedaanya hanya di penggunaan *package* http nya yang harus disesuaikan dengan jenis metode. Apabila menggunakan *GET* maka harus menggunakan fungsi *http.get*. Selain itu khusus untuk *GET*, tidak diperlukan *body* didalam API karena metode ini hanya digunakan untuk mengambil data dari Server. Tidak untuk mengirim atau

melakukan *update* data yang ada di server. Untuk penggunaan API di dalam komponen *User Interface*, bisa dilakukan dengan memanggil nama *function* yang telah dibuat dan ada API di dalamnya. Jika metodenya adalah *POST* atau *PUT*, maka komponen juga harus menyiapkan *body* yang dibutuhkan seperti pada **Gambar 6.34**.

```
onPressed: () {
  final body = {
    "nama": _namaVendor.text,
    "email": _emailVendor.text,
    "telepon": _teleponVendor.text,
    "alamat": _alamatVendor.text,
    "skor": 0,
  };
  addVendor(body).then((success) {
    if (success) {
      showDialog(
        builder: (context) => AlertDialog(
          title: Text(
            'Vendor telah berhasil ditambahkan!'),
          actions: <Widget>[
            FlatButton(
              onPressed: () {
                Navigator.pop(context);
                _namaVendor.text = '';
                _emailVendor.text = '';
                _teleponVendor.text = '';
                _alamatVendor.text = '';
              },
              child: Text('OK'),
            ) // FlatButton
          ], // <Widget>[]
        ), // AlertDialog
        context: context,
      );
      return;
    }
  });
}
```

**Gambar 6.34** Kode Sumber Penggunaan API di Komponen

Ada 18 API yang telah tersedia berdasarkan bab 3 di buku ini. Setiap API terdiri dari berbagai macam metode. Begitu pula dengan *User Interface* yang sudah dibuat. Satu *User Interface* bisa memiliki banyak API seperti pada **Tabel 6.17**.

**Tabel 6.17** Kebutuhan API di *User Interface*

ID UI	Nama UI	API yang dibutuhkan
UI1	Login	API1, API2
UI2	Home	API3
UI3	Vendor	API3, API5
UI4	AddVendor	API5
UI5	VendorDetail	API3, API5, API18
UI6	SPK	API3, API5
UI7	AddSPK	API4, API5, API6, API7, API8, API9, API10, API11, API12, API13, API14, API15
UI8	SPKDetail	API3, API9
UI9	SPKProgress	API3, API8, API9, API17
UI10	SPKReceived	API3, API7, API9, API10, API12, API15, API16
UI11	SPKEvaluation	API3, API9, API18
UI12	OngoingSPK	API3, API5
UI13	FinishedSPK	API3, API5
UI14	SPKDraft	API3, API5
UI15	SPKValidation	API3, API5
UI16	EditSPK	API4, API5, API6, API7, API8, API9, API10, API11, API12, API13, API14, API15
UI17	Account	API2
UI18	AccountList	API2
UI19	AddAccount	API2 dan API4
UI20	EditAccount	API2 dan API4

Semua *User Interface* sudah diintegrasikan dengan berbagai macam jenis API yang tersedia. Hasil dari integrasi ini bisa dilihat pada pengujian Blackbox yang ditampilkan secara lengkap pada **LAMPIRAN B**.

## 6.4 Manajemen Vendor

Pada tahapan penulis menjelaskan sistem manajemen vendor di dalam aplikasi setelah dikembangkannya *User Interface* dan Integrasi API. Sistem manajemen vendor akan menampilkan performa setiap vendor secara detail sehingga kerja sama dengan vendor tersebut dapat dilaksanakan kembali. Keterlambatan pengerjaan vendor juga ditampilkan di dalam aplikasi sehingga mampu meminimalisir kerugian finansial.

### 6.4.1 Penilaian Performa Vendor

Parameter yang digunakan untuk menilai performa vendor ada 4 jenis. Setiap Parameter mempunyai rentang nilai seperti pada **Tabel 6.18**.

**Tabel 6.18** Parameter Penilaian Performa Vendor

Nama Parameter	Rentang Nilai
Kerapihan	0-5
Kualitas Pengerjaan	0-5
Ketepatan Waktu	0-5
Kesesuaian Jumlah	0-5

Pengguna melakukan penilaian terhadap performa vendor setelah vendor menyelesaikan tugasnya. *User Interface* yang digunakan untuk menilai performa vendor adalah *SPKEvaluation*. Pengguna dapat menilai vendor dengan memilih bintang yang memiliki rentang nilai dari 0 sampai 5 seperti pada **Gambar 6.35**



**Gambar 6.35** Pengguna Menilai Performa Vendor

Pengguna menilai performa vendor dari semua parameter yang ada yaitu Kerapihan, Kualitas, Ketepatan dan Kesesuaian. Kemudian sistem akan melakukan perhitungan sehingga menghasilkan suatu nilai akhir untuk performa vendor. Berikut ini perhitungan yang digunakan untuk menilai performa vendor dalam mengerjakan tugasnya:

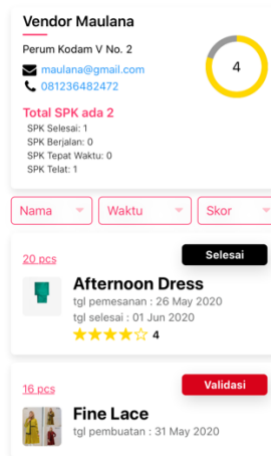
$$N = \frac{\text{Kerapihan} + \text{Kualitas} + \text{Ketepatan} + \text{Kesesuaian}}{4}$$

Berdasarkan **Gambar 6.35**, maka performa untuk Vendor Maulana setelah mengerjakan Afternoon Dress adalah 4. Berikut ini perhitungan performa vendor Maulana:

$$N = \frac{4 + 4 + 3 + 5}{4} = 4$$

#### 6.4.2 Detail Informasi Vendor

Semua informasi yang berhubungan dengan Vendor ada di dalam *User Interface* VendorDetail. Terdapat Informasi kontak vendor seperti nama, alamat, email, dan nomor telepon. Ada juga informasi seputar performa vendor dalam mengerjakan tugasnya. *User Interface* VendorDetail ditampilkan pada **Gambar 6.36**.



**Gambar 6.36** Detail Informasi Vendor di VendorDetail

Setiap vendor memiliki nilai akhir dari setiap pengerjaan SPK yang digunakan oleh pengguna untuk mengetahui performa vendor. Performa vendor digunakan sebagai patokan apakah kerja sama dilanjutkan atau tidak. Berikut ini perhitungan yang digunakan untuk menghasilkan nilai akhir vendor dari semua SPK yang pernah dikerjakan:

$$N = \frac{\text{Total Nilai semua SPK yang pernah dikerjakan}}{\text{Jumlah semua SPK yang pernah dikerjakan}}$$

Berdasarkan **Gambar 6.36**, maka nilai akhir untuk Vendor Maulana setelah mengerjakan semua SPK adalah 4. Berikut ini perhitungan performa vendor Maulana:

$$N = \frac{4}{1} = 4$$

### 6.4.3 Keterlambatan Vendor

Pengguna setelah masuk ke aplikasi dan melawati proses *login* maka aplikasi akan menampilkan *User Interface Home*. Pada halaman ini terdapat Informasi proses pengerjaan semua SPK yang sedang berjalan oleh Vendor. Setiap proses yang ada di SPK Berjalan ada indikator berupa warna yang digunakan oleh pengguna untuk mengetahui keterlambatan vendor selama mengerjakan tugasnya. Warna biru digunakan untuk memberi tanda bahwa SPK tepat waktu dan warna merah untuk SPK yang terlambat. Setiap SPK memiliki 3 buah proses. Ada Cutting, Jahit, dan Finishing. Setiap proses terdapat durasi pengerjaan yang direncanakan oleh pengguna ketika membuat SPK. Sistem di dalam aplikasi juga menyimpan durasi nyata pengerjaan SPK. Apabila ada salah satu proses yang durasi nyata nya lebih lama dibandingkan dengan durasi yang sudah direncanakan di awal, maka SPK tersebut akan digolongkan menjadi SPK yang terlambat. Tampilan SPK di Home akan berubah warnanya menjadi warna merah. Proses perubahan warna ini berlangsung secara *real time* ketika pergantian hari. Sehingga ketika pengguna masuk ke Home akan otomatis menampilkan tampilan yang paling terbaru. *User Interface Home* ditampilkan pada **Gambar 6.37**.





**Gambar 6.37 Keterlambatan SPK di Home**

*Halaman ini sengaja dikosongkan*

## BAB 7 PENGUJIAN

Pada bab ini dijelaskan hasil pengujian yang dilakukan. Proses pengujian dilakukan sebanyak dua kali. Pengujian pertama dilaksanakan sebelum kustomisasi dilakukan, dan pengujian kedua dilaksanakan setelah kustomisasi dilakukan. Pengujian kedua dilaksanakan untuk memastikan bahwa saran dan kendala yang dihadapi pengguna pada pengujian pertama sudah dapat diselesaikan.

### 7.1 Blackbox Testing

Pada tahap ini, dokumentasi blackbox testing dibuat untuk menguji dan memastikan apakah fungsional Aplikasi Giyomi.id dapat digunakan untuk mengakomodasi proses bisnis berdasarkan kasus yang diberikan. Blackbox testing dijalankan menggunakan *environment test* yaitu Android Emulator dan IOS Simulator. Hasil Blackbox testing menunjukkan bahwa aplikasi dapat berjalan dengan lancar. Hasil Blackbox testing dapat dilihat di **LAMPIRAN B**. Blackbox testing ini terdapat 29 *test case*. Masing-masing *test case* ditampilkan pada **Tabel 7.1**.

**Tabel 7.1 Test Case Blackbox Testing**

No	Test Case
A1	Login ke Aplikasi
A2	Menampilkan Akun
A3	Menampilkan Edit Akun
A4	Merubah Informasi Akun
A5	Menampilkan Daftar Akun Staf
A6	Menambah Akun Staf Baru
A7	Logout dari Aplikasi
V1	Menampilkan Daftar Vendor
V2	Sortir Daftar Vendor berdasarkan Skor
V3	Menambah Vendor Baru
V4	Menampilkan Detail Vendor

V5	Menghubungi Nomor Telepon Vendor
S1	Menampilkan Daftar SPK
S2	Sortir Daftar SPK berdasarkan Vendor
S3	Menambah Draf SPK Baru
S4	Meminta Validasi SPK
S5	Melakukan Validasi SPK
S6	Melihat SPK yang Terlambat
S7	Melihat Detail Koleksi SPK Berjalan
S8	Menunda SPK Berjalan
S9	Melanjutkan SPK yang Ditunda
S10	Menambahkan Catatan Proses SPK
S11	Menyelesaikan Proses SPK
S12	Melihat Proses SPK yang telah Selesai
S13	Menerima Barang SPK
S14	Mengembalikan Barang SPK
S15	Menutup SPK
S16	Melakukan Evaluasi SPK
S17	Melihat Progress SPK

## 7.2 Usability Testing

Usability Testing dilakukan dengan menggunakan aplikasi *frontend* yang telah ditampilkan dalam **LAMPIRAN A**. Pengguna diberi sejumlah *test case* untuk mengukur seberapa baik aplikasi yang telah dibuat. Bab ini menyajikan protokol pengujian, skenario pengujian, dan implementasi pengujian di setiap iterasi.

### 7.2.1 Tahap Awal

Pengguna diuji menggunakan *test case* yang telah dibuat. Peneliti yang bertindak seperti moderator mencatat setiap kesulitan, atau keraguan pengguna. Data yang terkait dengan jumlah kesalahan dan kepuasan pengguna dikumpulkan dan kemudian dianalisis untuk memutuskan seberapa baik aplikasi dalam hal kegunaan. Kekurangan aplikasi ditunjukkan selama

tes, dan mereka digunakan sebagai referensi untuk meningkatkan kualitas aplikasi.

### 7.2.2 Protokol Pengujian

Protokol pengujian bermanfaat sebagai pedoman dalam melakukan proses uji kegunaan. Protokol ini berisi langkah-langkah dan persiapan yang harus dilakukan untuk mendapatkan hasil maksimal. Peneliti bertindak sebagai moderator dalam menjalani sesi pengujian. Protokol pengujian dirangkum dalam **LAMPIRAN C**.

### 7.2.3 Skenario Pengujian

Peneliti sebagai moderator *usability testing* memberikan 12 *test case* kepada pengguna. Moderator menjelaskan tujuan pengujian yaitu untuk memvalidasi aplikasi *frontend* yang dibuat apakah cukup jelas untuk digunakan oleh pengguna.

*Test case* dibuat berdasarkan tampilan-tampilan yang telah dibuat di bab sebelumnya agar selaras dengan solusi yang dibutuhkan Giyomi. Ada 12 *test case*, masing-masing ditunjukkan pada **Tabel 7.2**.

**Tabel 7.2 Test Case Usability Testing**

No	Test Case
TC 1	Pengguna menampilkan daftar semua SPK dan sebutkan berapa banyak, setelah itu pengguna mengubah filter ke SPK yang sedang berlangsung dan pengguna sebutkan berapa
TC 2	Pengguna melihat aksesori detail Ameera Sweater SPK yang sedang berlangsung
TC 3	Pengguna melihat progres Ameera Sweater SPK dari Eka dan selesaikan semua proses. Pengguna menyebutkan batas waktu untuk setiap tahap. Pengguna memberikan catatan di setiap tahap dengan detail berikut ini:

	<p>Dalam proses cutting: <i>Pengerjaan lebih cepat satu hari.</i></p> <p>Dalam proses jahit: <i>benang habis. mohon kirim benang 2 roll.</i></p> <p>Dalam proses finishing: <i>masih dalam proses setrika dan packing.</i></p>																																																
TC 4	<p>Pengguna mencatat tanda terima barang dari SPK Sweater Ameera nya Eka dengan perincian berikut:</p> <table border="1"> <thead> <tr> <th>Nama Varian</th> <th>XS</th> <th>S</th> <th>M</th> <th>L</th> <th>XL</th> <th>XXL</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>Maroon</td> <td>0</td> <td>15</td> <td>10</td> <td>5</td> <td>5</td> <td>5</td> <td><i>accept</i></td> </tr> <tr> <td>Maroon</td> <td>0</td> <td>10</td> <td>5</td> <td>0</td> <td>0</td> <td>0</td> <td><i>return</i></td> </tr> <tr> <td>Maroon</td> <td>0</td> <td>10</td> <td>5</td> <td>0</td> <td>0</td> <td>0</td> <td><i>accept</i></td> </tr> <tr> <td>Biru</td> <td>15</td> <td>5</td> <td>10</td> <td>5</td> <td>5</td> <td>0</td> <td><i>accept</i></td> </tr> <tr> <td>Hitam</td> <td>0</td> <td>15</td> <td>20</td> <td>10</td> <td>5</td> <td>0</td> <td><i>accept</i></td> </tr> </tbody> </table>	Nama Varian	XS	S	M	L	XL	XXL	Status	Maroon	0	15	10	5	5	5	<i>accept</i>	Maroon	0	10	5	0	0	0	<i>return</i>	Maroon	0	10	5	0	0	0	<i>accept</i>	Biru	15	5	10	5	5	0	<i>accept</i>	Hitam	0	15	20	10	5	0	<i>accept</i>
Nama Varian	XS	S	M	L	XL	XXL	Status																																										
Maroon	0	15	10	5	5	5	<i>accept</i>																																										
Maroon	0	10	5	0	0	0	<i>return</i>																																										
Maroon	0	10	5	0	0	0	<i>accept</i>																																										
Biru	15	5	10	5	5	0	<i>accept</i>																																										
Hitam	0	15	20	10	5	0	<i>accept</i>																																										
TC 5	<p>Pengguna menutup SPK Ameera Sweater dan Memberikan evaluasi vendor dengan rincian berikut ini:</p> <ol style="list-style-type: none"> <li><i>1. Kerapihan: 5</i></li> <li><i>2. Kualitas Pengerjaan: 4</i></li> <li><i>3. Ketepatan Waktu: 4</i></li> <li><i>4. Kesesuaian Jumlah: 5</i></li> </ol> <p>Dan berikan komentar dengan rincian berikut: <i>Kualitas pengerjaan masih ada yang kurang dan sempat ada yang dikembalikan</i></p>																																																
TC 6	<p>Menambahkan vendor baru dengan detail berikut:</p> <p><i>Nama: Lila</i>  <i>Alamat email: lila@gmail.com</i>  <i>No. Telepon: 08112345678</i>  <i>Alamat: Jalan Gebang Wetan 1</i></p>																																																
TC 7	<p>Pengguna mengurutkan daftar vendor dari tertinggi ke terendah skor untuk menganalisis vendor terbaik untuk SPK baru.</p>																																																

TC 8	<p>Pengguna membuat SPK baru dan menyimpan SPK sebagai draf. Detail untuk SPK adalah sebagai berikut:  <i>Nama Koleksi: Sahara Dress</i>  <i>Vendor: Vendor dengan nilai paling tinggi</i>  <i>Jenis produk: Atasan</i>  <i>Detail aksesoris: Renda di leher</i></p>
TC 9	<p>Pengguna mengakses halaman draf untuk membuka SPK yang baru saja disimpan. Pengguna terus merubah SPK dan mengirimkannya untuk proses validasi. Informasi detail untuk draf adalah sebagai berikut:  <i>Nama Varian: Putih</i>  <i>Satuan Panjang Kain: Meter</i>  <i>Panjang Kain: 100</i>  <i>XS: 5, S: 10, M: 20, L: 20, XL: 5, XXL: 5</i>  <i>Nama Varian: Creme</i>  <i>Satuan Panjang Kain: Meter</i>  <i>Panjang Kain: 100</i>  <i>XS: 5, S: 10, M: 20, L: 10, XL: 5, XXL: 5</i></p>
TC 10	<p>Pengguna melakukan validasi SPK dengan mengubah jumlah ukuran S di Varian Creme dari 10 menjadi 20 pcs, dan pengguna mengirimkannya ke Vendor.</p>
TC 11	<p>Pengguna menampilkan daftar SPK dari Vendor Maulana.</p>
TC 12	<p>Pengguna membuat akun staf baru dengan detail:  <i>Username: Kiki</i>  <i>Email: kiki@gmail.com</i>  <i>Nomor Telepon: 08123332451</i>  <i>Password: pass_kiki</i></p> <p>Lalu merubah detail akunnya menjadi:  <i>Nomor Telepon: 08112341245</i></p>

#### 7.2.4 Traceability Matrix

Traceability Matrix dibuat seperti yang ditunjukkan pada **Tabel 7.3** untuk memetakan kelengkapan hubungan antara *test case* dan tampilan aplikasi *frontend* yang sudah dibuat. Satu *test case*





### 7.2.5 Karakteristik Responden

Dalam penelitian ini, responden untuk setiap iterasi yang dipilih adalah pengguna yang bekerja di Giyomi dan memiliki smartphone yang sistem operasinya adalah Android atau IOS. Responden yang akan melakukan pengujian di iterasi pertama dan kedua sama, ini memiliki tujuan agar *feedback* yang di dapatkan sebelumnya telah terealisasi dengan baik di iterasi berikutnya. Karakteristik pengguna dalam *usability testing* ini ditunjukkan pada **Tabel 7.4**.

**Tabel 7.4 Karakteristik Responden**

<b>Nama</b>	<b>Jabatan di Giyomi</b>	<b>Sistem Operasi</b>	<b>Jenis Smartphone</b>
Yudha Prasetyo	Owner	IOS	Iphone 8 Plus

### 7.2.6 Hasil Pengujian

Ada sedikit perbedaan antara hasil dari iterasi pertama dan yang kedua. Ini menunjukkan tren peningkatan dalam hal efektivitas aplikasi.

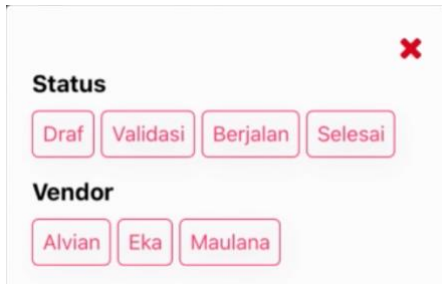
#### 1. Iterasi Pertama

Dalam iterasi pertama ada beberapa *test case* yang tidak terselesaikan oleh responden, ringkasan dari formulir *usability testing* ditunjukkan pada **LAMPIRAN D**. *Test Case* yang gagal dilakukan adalah:

*Pengguna menampilkan daftar semua SPK dan sebutkan berapa banyak, setelah itu pengguna mengubah filter ke SPK yang sedang berlangsung dan pengguna sebutkan berapa*

Pengguna hanya bisa melakukan filter daftar SPK berdasarkan nama dan waktu. Padahal *test case* nya jelas menugaskan untuk melakukan filter SPK dengan status sedang berlangsung. Pengguna kebingungan mencari tombol untuk melakukan filter tersebut. Akhirnya pengguna hanya bisa menyebutkan jumlah SPK yang ada tetapi tanpa filter. Sehingga dibutuhkan perbaikan dengan menambahkan *User Interface* untuk

mengubah filter berdasarkan Status. *User Interface* yang baru seperti pada



**Gambar 7.1 Filter SPK berdasarkan Status atau Vendor**

Pengguna kemudian lanjut ke *test case* berikutnya. *Test case* berikutnya berjalan dengan lancar tetapi mulai ada kendala ketika memasuki *test case* 3:

*Pengguna melihat progres Ameera Sweater SPK dari Eka dan selesaikan semua proses. Pengguna menyebutkan batas waktu untuk setiap tahap. Pengguna memberikan catatan di setiap tahap dengan detail berikut ini:*

*Dalam proses cutting:  
Pengerjaan lebih cepat satu hari.*

*Dalam proses jahit:  
benang habis. mohon kirim benang 2 roll.*

*Dalam proses finishing:  
masih dalam proses setrika dan packing.*

Pengguna mampu melihat progres SPK dari tahap Cutting, Jahit sampai Finishing. Tanggal rencana penyelesaian proses juga ditampilkan di masing-masing tampilan proses sehingga pengguna bisa menyebutkan setiap tanggalnya. Kemudian pengguna menambahkan catatan. Tetapi ketika mencoba memasukkan catatan untuk kedua kalinya yaitu di proses jahit, pengguna malah gagal dan aplikasi menampilkan peringatan kalau terjadi kesalahan atau *error*. Dibutuhkan perbaikan yaitu untuk menambahkan catatan hingga kedua kalinya seperti pada Gambar

**Catatan Step Cutting:**

yudha:  
"Ada kain yg robek"  
27/05 3:06

yudha:  
"tes lagi"  
27/05 3:20

**Gambar 7.2 Penambahan Catatan lebih dari dua kali**

Pengguna mencatat tanda terima barang dari SPK Sweater Ameera nya Eka dengan perincian berikut:

Nama Varian	XS	S	M	L	XL	XXL	Status
Maroon	0	15	10	5	5	5	accept
Maroon	0	10	5	0	0	0	return
Maroon	0	10	5	0	0	0	accept
Biru	15	5	10	5	5	0	accept
Hitam	0	15	20	10	5	0	accept

Pengguna sangat lama sekali ketika mengerjakan *test case* nomor 4. Setiap melakukan penerimaan atau penambahan, aplikasi menampilkan tampilan *loading* yang bisa melebihi 30 detik. Ketika selesai aplikasi menampilkan *error* dan proses penerimaan barang gagal. Sehingga dibutuhkan perbaikan agar pengguna bisa melakukan penerimaan maupun pengembalian barang seperti pada **Gambar**

**Rincian Penerimaan Barang**

Varian Belum Dikirim

Hitam 0 pcs

Putih 0 pcs

**Tutup SPK**

tanggal penerimaan: 07 Jun 2020

Varian	XS	S	M	L	XL	XXL	Total	Status
Hitam	1	1	1	1	2	1	7	accept
Putih	1	1	1	1	2	2	8	accept

**Gambar 7.3 Pengambilan dan Penerimaan Barang SPK**

Terakhir pengguna gagal menjalankan *test case* nomor 9 yaitu:

*Pengguna mengakses halaman draf untuk membuka SPK yang baru saja disimpan. Pengguna terus merubah SPK dan mengirimkannya untuk proses validasi. Informasi detail untuk draf adalah sebagai berikut:*

*Nama Varian: Putih*

*Satuan Panjang Kain: Meter*

*Panjang Kain: 100*

*XS: 5, S: 10, M: 20, L: 20, XL: 5, XXL: 5*

*Nama Varian: Creme*

*Satuan Panjang Kain: Meter*

*Panjang Kain: 100*

*XS: 5, S: 10, M: 20, L: 10, XL: 5, XXL: 5*

Pada bagian ini, pengguna kebingungan mencari tombol untuk menambahkan varian. Sehingga pengguna hanya bisa menambahkan varian dengan nama Putih. Sedangkan varian dengan nama Creme tidak berhasil ditambahkan karena tidak ada kolom dan tombol untuk mengisi varian lebih dari satu. Dibutuhkan kolom dan tombol baru untuk pengguna menambahkan varian lebih dari satu. Hasilnya adalah seperti pada **Gambar 7.4**

**Detail Varian**

Nama Varian

Satuan Panjang Kain

Panjang Kain

Ukuran

XS  S  M

L  XL  XXL



**Gambar 7.4** Tambah Varian di SPK

## 2. Iterasi Kedua

Hasil dari iterasi kedua menunjukkan tren yang berbeda dari yang pertama. Seperti yang ditunjukkan dalam **LAMPIRAN D**. Pengguna dapat menyelesaikan tugas yang diberikan setelah desain telah direvisi berdasarkan *Feedback* yang diperoleh dari iterasi sebelumnya. Karena responden dapat menyelesaikan semua tugas dan tidak ada kesulitan signifikan yang dihadapi, penulis merasa cukup untuk melakukan pengujian kegunaan hanya dua kali.

### 7.2.7 Effectivity Matrix

Peneliti mengukur kemampuan pengguna untuk menyelesaikan tugas. Efektivitas dapat dihitung dari tingkat penyelesaian *test case* [12]. Peneliti memberikan nilai 1 untuk tugas yang telah diselesaikan oleh pengguna, dan 0 untuk tugas yang gagal. Berikut ini perhitungan yang digunakan untuk mengukur efektivitas:

$$N = \frac{\text{Completed Test Case}}{\text{Total Test Case}} \times 100\%$$

#### 1. Iterasi Pertama

Efektivitas menunjukkan tingkat keberhasilan pengguna untuk semua tugas yang dikerjakan. Misalnya, pengguna gagal menyelesaikan satu tugas dari semua 12 *test case* yang diberikan. Keefektifannya dapat dihitung sebagai berikut:

$$N = \frac{8}{12} \times 100\%$$

Berdasarkan perhitungan tersebut, disimpulkan bahwa pengguna di iterasi pertama mendapat tingkat keberhasilan 66,67%.

#### 2. Iterasi Kedua

Karena pengguna dalam iterasi kedua mampu menyelesaikan semua *test case*, sehingga tingkat keberhasilannya adalah 100%. Jika kita membandingkan keefektifan dari dua iterasi, ditemukan peningkatan signifikan yang menunjukkan bahwa aplikasi *frontend* revisi terakhir telah dipahami dengan baik oleh pengguna.

## **BAB 8**

### **PENUTUP**

Bab ini berisi mengenai kesimpulan dan saran mengenai hasil penelitian. Kesimpulan berisi rangkuman dari setiap tahapan pengerjaan, dimulai dari analisis kesenjangan dengan kemampuan aplikasi hingga pengujian. Saran berisi rekomendasi dari penulis terhadap keberlanjutan penelitian agar dapat dijadikan referensi untuk dilanjutkan atau penelitian lain yang berkaitan.

#### **8.1 Kesimpulan**

Bedasarkan hasil penelitian terkait pengembangan aplikasi *frontend* Sistem Manajemen Vendor, berikut beberapa kesimpulan yang dapat diambil:

1. Informasi yang didapatkan dari tahap analisis kebutuhan *User Interface* dan API sistem manajemen vendor Giyomi.id adalah sebagai berikut:
  - a. Terdapat 20 *User Interface* yang dibutuhkan untuk membuat aplikasi *frontend* sistem manajemen vendor Giyomi.id
  - b. Terdapat 18 kebutuhan API yang harus digunakan oleh aplikasi *frontend* untuk menjalankan fitur di sistem manajemen vendor Giyomi.id
2. Aplikasi *frontend* sudah terintegrasi dengan aplikasi *backend* sehingga menjadi aplikasi sistem manajemen vendor yang dapat dimanfaatkan oleh Giyomi.id untuk:
  - a. Memberikan informasi akurat tentang performa tiap vendor sehingga bisa dijadikan acuan untuk keberlanjutan kerja sama
  - b. Menampilkan keterlambatan proses produksi sehingga mampu meminimalisir kerugian finansial
3. Pengujian-pengujian yang dilakukan terhadap beberapa aspek dari aplikasi *frontend* berstatus lolos uji, pengujian-pengujian yang dilakukan sebagai berikut:
  - a. Blackbox Testing  
Pengujian yang dilakukan untuk memastikan setiap fitur di dalam aplikasi *frontend* bekerja dengan baik
  - b. Usability Testing

Pengujian yang dilakukan untuk memastikan aplikasi *frontend* telah digunakan dan memenuhi kebutuhan pihak Giyomi.id

## 8.2 Saran

Berdasarkan kesimpulan hasil penelitian, terdapat beberapa saran yang dapat dijadikan masukan.

### 8.2.1 Saran untuk Penelitian Selanjutnya

Beberapa saran yang dapat dijadikan masukan untuk penelitian selanjutnya adalah sebagai berikut :

1. Aktivitas pengujian kurang efektif bila dilakukan secara daring, sehingga disarankan melakukan pengujian secara tatap muka.
2. Melakukan pengujian lebih dari satu kali selama pengembangan sangat disarankan karena mempermudah proses pengembangan.

### 8.2.2 Saran untuk UMKM Giyomi.id

Beberapa saran yang dapat dijadikan masukan untuk pihak UMKM Giyomi.id adalah sebagai berikut:

1. Pihak UMKM dapat mengimplementasikan aplikasi *frontend* Giyomi.id dengan melakukan migrasi secara bertahap agar tidak mengganggu proses yang sedang berjalan secara konvensional.
2. Pihak UMKM perlu memastikan bahwa aplikasi sistem manajemen vendor selama digunakan dalam kegiatan sehari-hari dapat berjalan dengan baik.



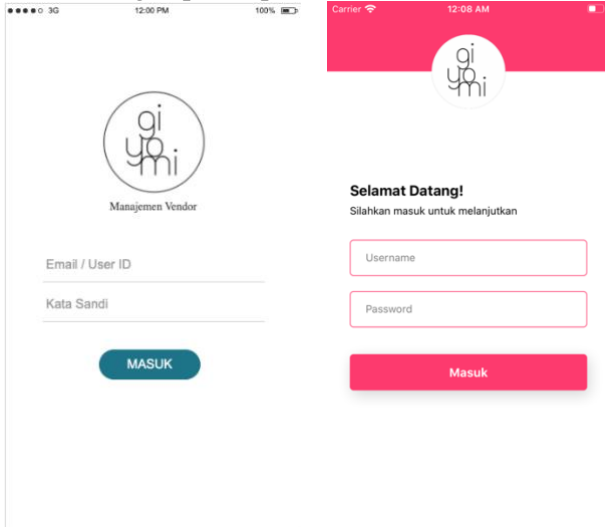
## DAFTAR PUSTAKA

- [1] A. Kinasih, "Analisis dan Usulan Rancang Ulang Proses Bisnis Produksi pada Usaha Garmen Make to Stock untuk Meningkatkan Ketepatan Waktu Produksi (Studi Kasus: UMKM Giyomi.id)", Tugas Akhir Departemen Sistem Informasi Fakultas Teknologi Informasi dan Komunikasi ITS Surabaya, 2019.
- [2] E. M. Bening, Perancangan Prototipe Sistem Manajemen Vendor dengan Metode Iterative Prototyping dan Moderated Usability Test. (Studi Kasus : UMKM Giyomi.Id), Tugas Akhir Departemen Sistem Informasi Fakultas Teknologi Informasi dan Komunikasi ITS, Surabaya, 2019.
- [3] W. Wu, "React Native vs Flutter, cross-platform mobile application frameworks," Thesis of Information Technology Department Metropolia University of Applied Sciences, 2018.
- [4] T. Y. T. Wong, Software Development Life Cycle, 2016.
- [5] Tutorialspoint, "SDLC - Waterfall Model," [Online]. Available: [https://www.tutorialspoint.com/sdlc/sdlc\\_waterfall\\_model.htm](https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm). [Accessed 5 November 2019].
- [6] F. Hasbi, "Pengembangan Front-End Aplikasi UIIPORTOFOLIO Berbasis Web," Tugas Akhir Departemen Teknik Informatika Fakultas Teknologi Industri UII, Yogyakarta, 2018.
- [7] L. Dagne, "Flutter for cross-platform App and SDK Development," Thesis of Information Technology Department Metropolia University of Applied Sciences, 2019.
- [8] A. F. Daryavarsari, "Comparing Black- and White-box testing environments using mixed research methods and AHP," Thesis of Informatics Department University of Oslo, 2013.
- [9] H. C. a. P. Mohapatra, "Using service brokers for accessing backend servers for web applications," J. Netw. Comput. Appl., 2005.
- [10] G. W. Setiawan, "Pengujian Perangkat Lunak Menggunakan Metode Black Box Studi Kasus Exelsa Universitas Sanata

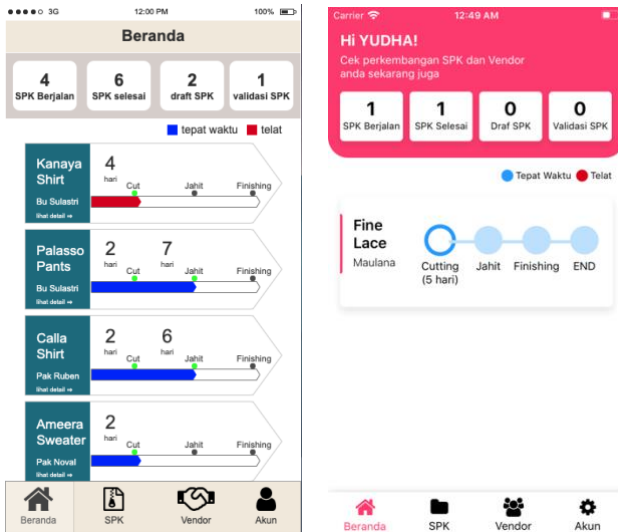
- Dharma,” Tugas Akhir Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Sanata Dharma, Yogyakarta,, 2011.
- [11] S. B. a. M. S. Murugaiyan, “Waterfall vs. V-Model vs. Agile: A comparative study on SDLC,” *Int. J. Inf. Technol. Bus. Manag.*, 2012.
- [12] J. S. a. J. R. Lewis, *Quantifying the user experience: Practical statistics for user research.*, Waltham: Morgan Kaufmann, 2016.

## LAMPIRAN A. USER INTERFACE APLIKASI

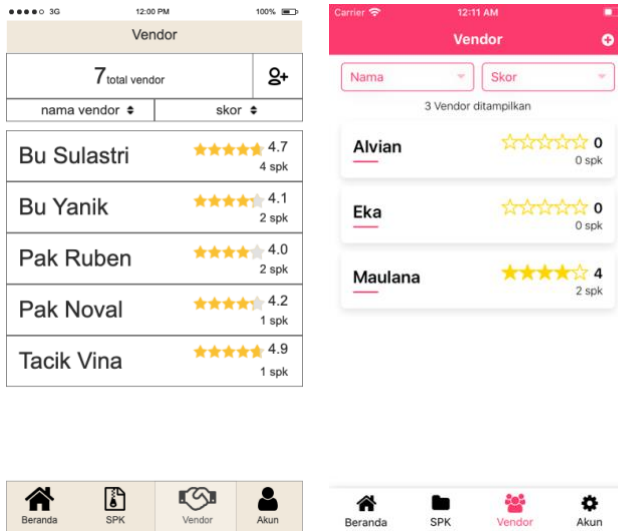
Dokumentasi perbandingan antara prototipe dengan aplikasi *frontend* dibuat untuk memastikan apakah tampilan aplikasi sudah sesuai dengan prototipe.



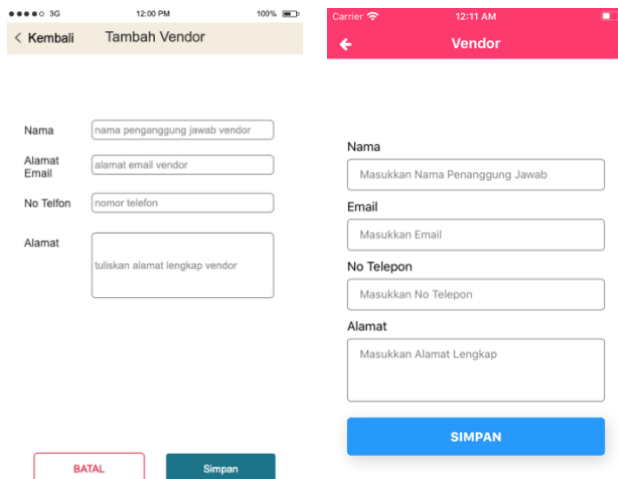
Gambar 8.1 Login



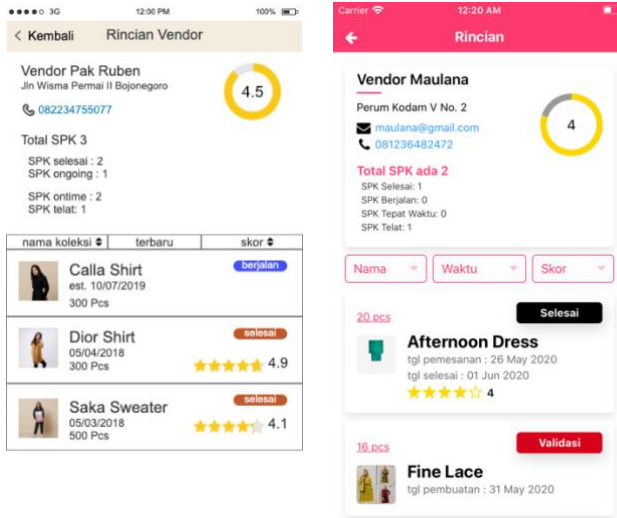
Gambar 8.2 Home



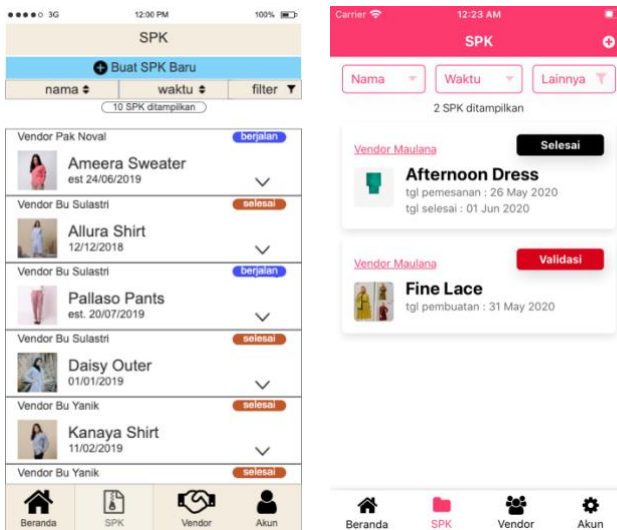
Gambar 8.3 Vendor



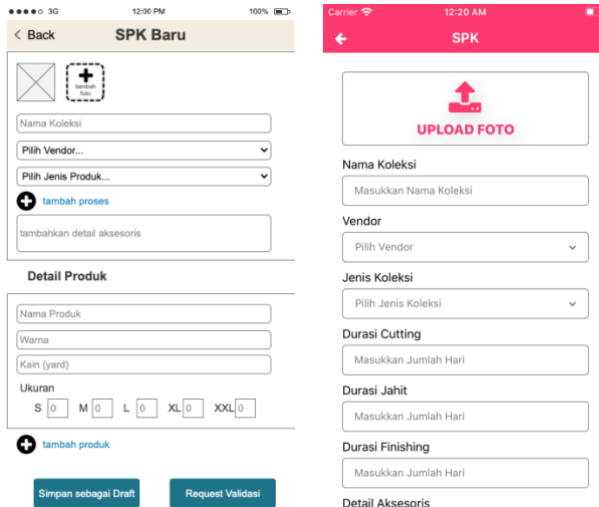
Gambar 8.4 Add Vendor



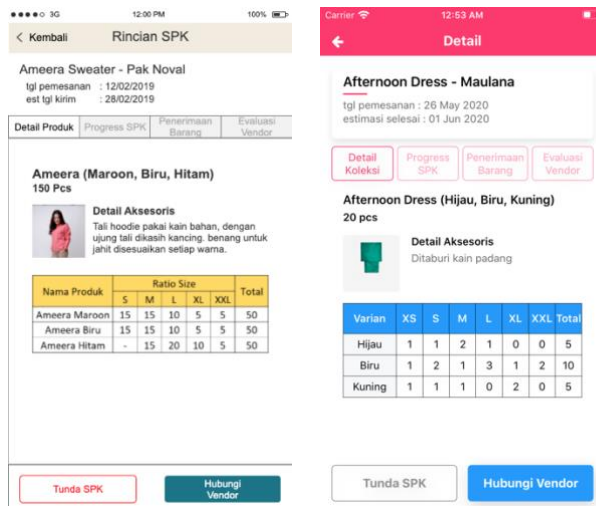
Gambar 8.5 Vendor Detail



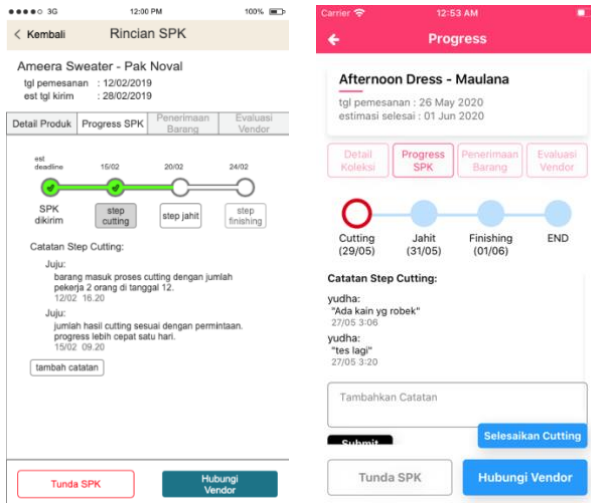
Gambar 8.6 SPK



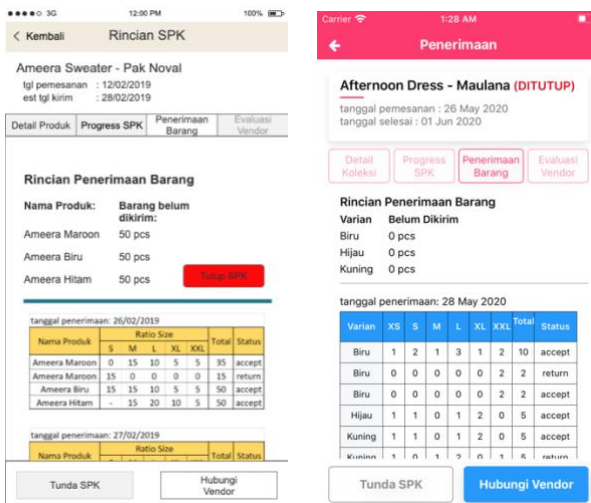
Gambar 8.7 Add SPK



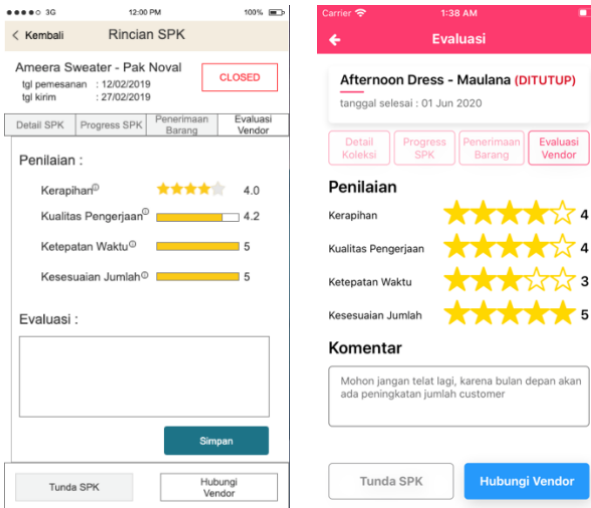
Gambar 8.8 SPK Detail



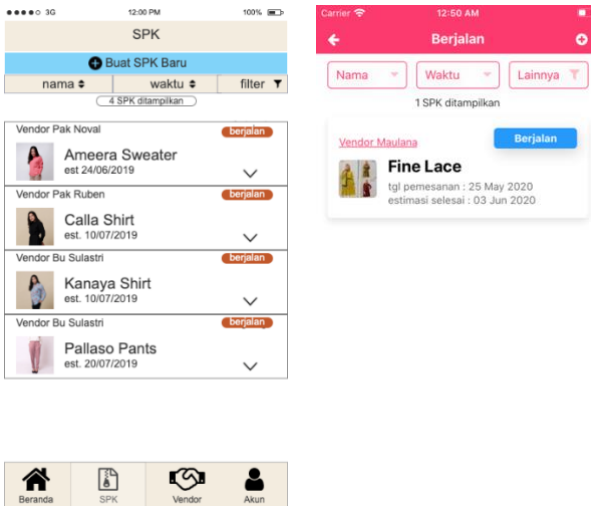
Gambar 8.9 SPK Progress



Gambar 8.10 SPK Received

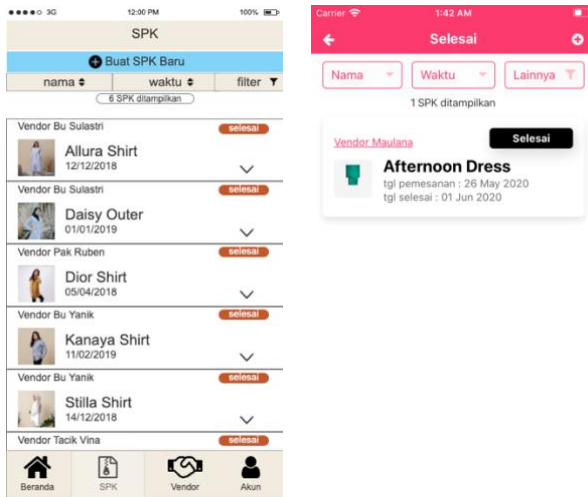


Gambar 8.11 SPK Evaluation

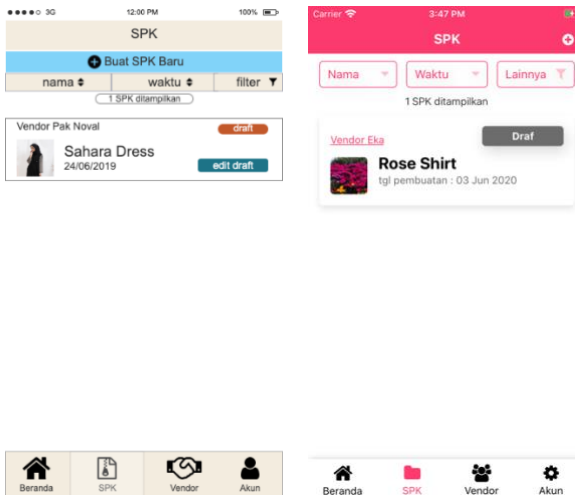


Gambar 8.12 Ongoing SPK

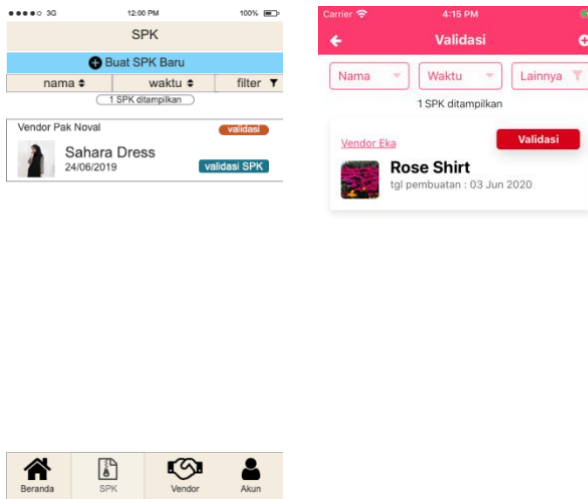




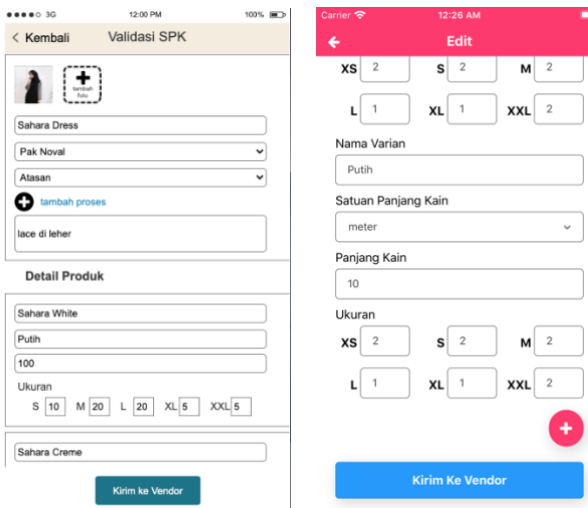
Gambar 8.13 Finished SPK



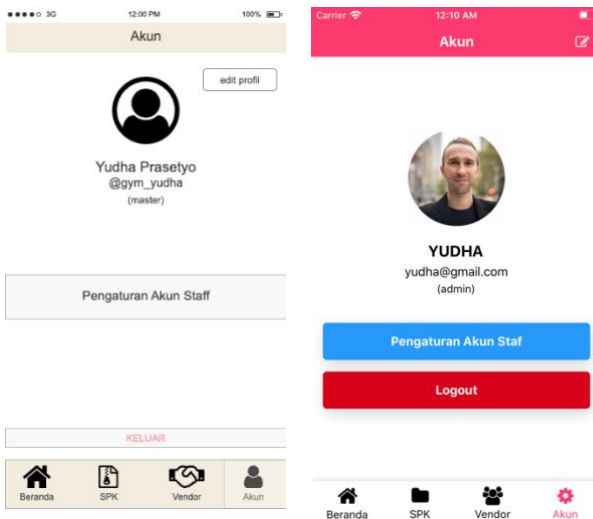
Gambar 8.14 SPK Draft



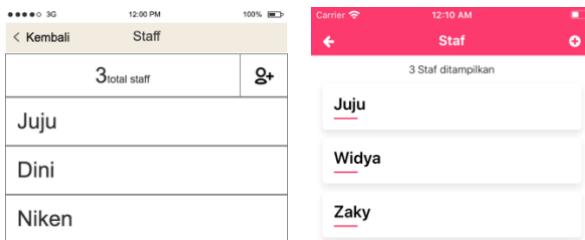
Gambar 8.15 SPK Validation



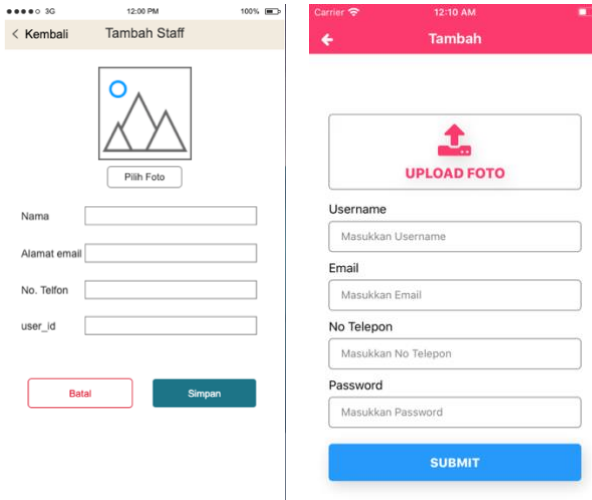
Gambar 8.16 Edit SPK



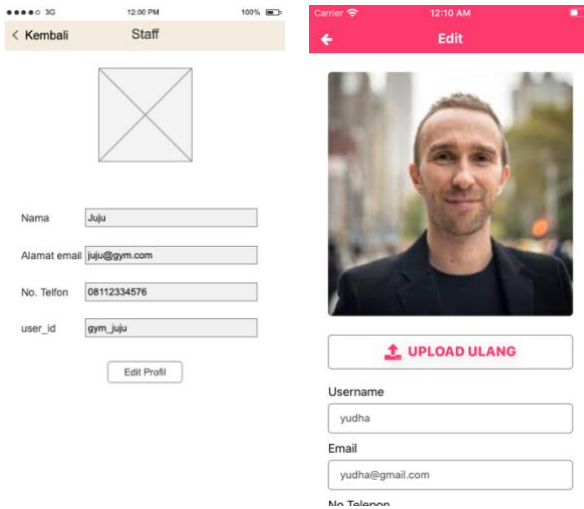
Gambar 8.17 Account



Gambar 8.18 Account List



Gambar 8.19 Add Account



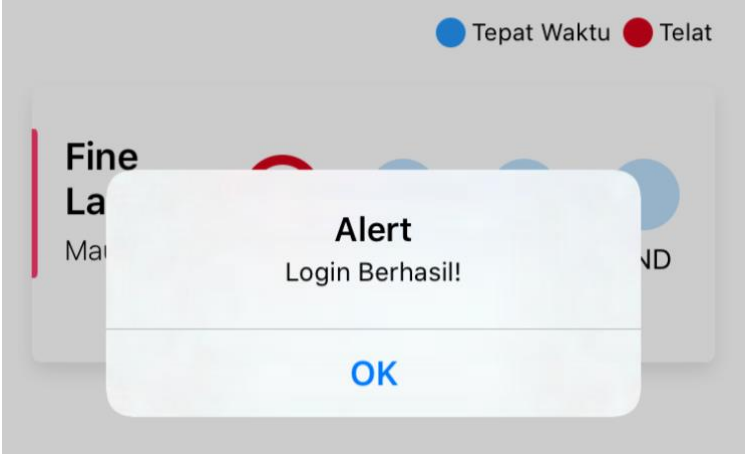
Gambar 8.20 Edit Account

## LAMPIRAN B. BLACKBOX TESTING


Pada tahap ini, dokumentasi Blackbox testing dibuat untuk menguji dan memastikan apakah fungsi-fungsi dari aktivitas yang dilakukan pada aplikasi *frontend* sistem manajemen vendor sudah sesuai dengan hasil yang diharapkan.

### AKUN


Tabel 8.1 Login

<b>ID Test Case: A1</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : Login		Tujuan: Pengguna masuk ke halaman utama untuk mengakses fitur-fitur yang tersedia	
Tester: Alvian Eka Maulana		Route: <b>Login</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Memasukkan username	Kolom username sudah terisi	Sukses
2.	Memasukkan password	Kolom password sudah terisi	Sukses
3.	Menekan tombol <b>Masuk</b>	Pengguna masuk ke halaman utama	Sukses
Contoh Luaran:			
			

Tabel 8.2 Menampilkan Akun

<b>ID Test Case: A2</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : Menampilkan Akun		Tujuan: Pengguna masuk ke halaman akun untuk melihat foto, nama, email, dan role pengguna	
Tester: Alvian Eka Maulana		Route: <b>Account</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Menekan tombol <b>Akun</b>	Menampilkan foto, nama, email, dan role pengguna	Sukses
<p>Contoh Luaran:</p> <div style="text-align: center;">  <p><b>YUDHA</b> yudha@gmail.com (admin)</p> <p><b>Pengaturan Akun Staf</b></p> <p><b>Logout</b></p> </div>			

**Tabel 8.3 Menampilkan Halaman Edit Akun**

<b>ID Test Case: A3</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : Menampilkan Halaman Edit Akun		Tujuan: Pengguna masuk ke halaman edit akun untuk melihat foto, nama, email, no. telepon, dan password	
Tester: Alvian Eka Maulana		Route: <b>Account</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Menekan logo 	Menampilkan foto, nama, email, no. telepon, dan password	Sukses

Contoh Luaran:



 **UPLOAD ULANG**

Username

yudha

Email

yudha@gmail.com

No Telepon

081231684641

Password

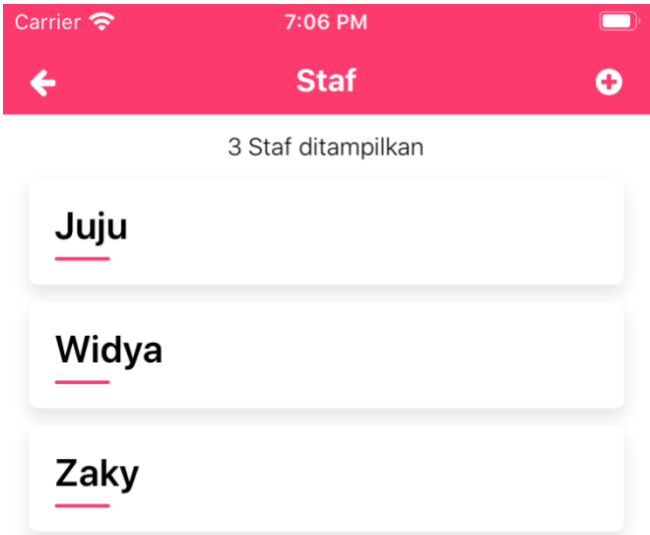
••••••••

Tabel 8.4 Merubah Informasi Akun


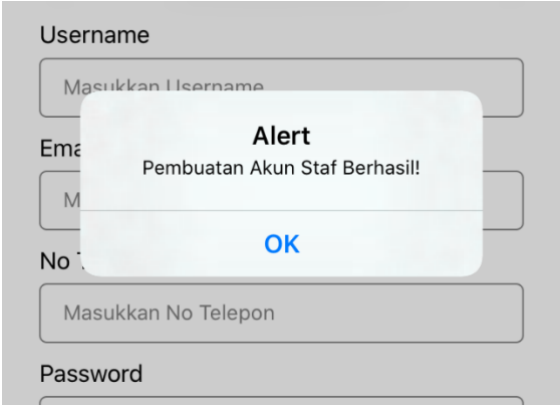
<b>ID Test Case: A4</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : Merubah Informasi Akun		Tujuan: Pengguna merubah informasi akun seperti foto, nama, no.telepon, dan password	
Tester: Alvian Eka Maulana		Route: <b>Account -&gt; Edit Account</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Menekan tombol <b>Upload Ulang</b>	Menampilkan pilihan untuk membuka Kamera Foto atau memilih gambar dari Galeri	Sukses
2.	Memilih <b>Pilih Dari Galeri</b>	Galeri <i>Smartphone</i>	Sukses
3.	Memilih gambar di galeri	Gambar ditampilkan di dalam aplikasi	Sukses
4.	Merubah informasi username	Kolom username terisi dengan informasi yang baru	Sukses
5.	Merubah informasi email	Kolom email terisi dengan informasi yang baru	Sukses
6.	Merubah informasi nomor telepon	Kolom nomor telepon terisi dengan informasi yang baru	Sukses
7.	Merubah informasi password	Kolom password terisi dengan informasi yang baru	Sukses
8.	Menekan tombol <b>Update</b>	Perubahan informasi akun disimpan	Sukses
Contoh Luaran:			
			



Tabel 8.5 Menampilkan Daftar Akun Staf

<b>ID Test Case: A5</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : Menampilkan Daftar Akun Staf		Tujuan: Pengguna menampilkan daftar akun para staf	
Tester: Alvian Eka Maulana		Route: <b>Account</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Menekan tombol <b>Pengaturan Akun Staf</b>	Menampilkan daftar akun para staf	Sukses
Contoh Luaran:			
 <p>The screenshot shows a mobile application interface. At the top, there is a pink header bar containing a back arrow on the left, the word "Staf" in the center, and a plus sign on the right. Below the header, the text "3 Staf ditampilkan" is centered. Underneath, there is a list of three staff members, each represented by a white rounded rectangle with a red underline: "Juju", "Widya", and "Zaky".</p>			

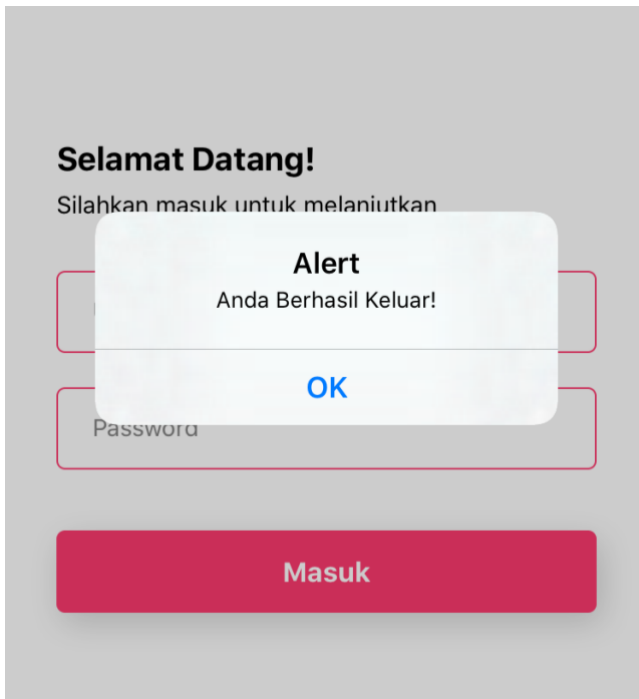
Tabel 8.6 Menambah Akun Staf Baru

<b>ID Test Case: A6</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : Menambah Akun Staf Baru		Tujuan: Pengguna merubah informasi akun seperti foto, nama, no.telepon, dan password	
Tester: Alvian Eka Maulana		Route: <b>Account -&gt; List Account</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Menekan logo 	Menampilkan formulir isian untuk menambah akun staf	Sukses
2.	Mengisi username	Kolom username sudah terisi	Sukses
3.	Mengisi email	Kolom email sudah terisi	Sukses
4.	Mengisi nomor telepon	Kolom nomor telepon sudah terisi	Sukses
5.	Mengisi password	Kolom password sudah terisi	Sukses
6.	Menekan tombol <b>Submit</b>	Akun staf baru dibuat	Sukses
Contoh Luaran:			
			

Tabel 8.7 Logout

<b>ID Test Case: A7</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : Logout		Tujuan: Pengguna keluar menuju halaman Login	
Tester: Alvian Eka Maulana		Route: <b>Account</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Menekan tombol <b>Logout</b>	Menampilkan <i>pop-up</i> untuk mengkonfirmasi apakah akan keluar aplikasi	Sukses
2.	Menekan tombol <b>Ya</b>	Keluar menuju halaman Login	Sukses

Contoh Luaran:



## VENDOR

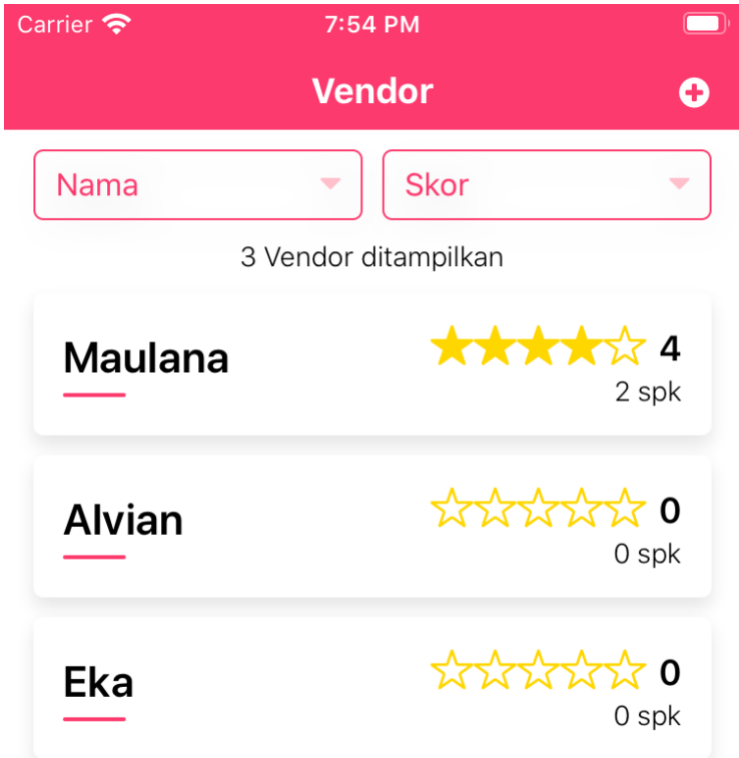
**Tabel 8.8 Menampilkan Daftar Vendor**

<b>ID Test Case: V1</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : Menampilkan Daftar Vendor		Tujuan: Pengguna menampilkan daftar vendor yang berisi nama, jumlah spk dan skor	
Tester: Alvian Eka Maulana		Route: <b>Home</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Menekan tombol <b>Vendor</b>	Pengguna masuk ke halaman vendor	Sukses
Contoh Luaran:			
<p>Carrier 7:47 PM</p> <p><b>Vendor</b> +</p> <p>Nama Skor</p> <p>3 Vendor ditampilkan</p> <p><b>Alvian</b> ★★★★★ 0 0 spk</p> <p><b>Eka</b> ★★★★★ 0 0 spk</p> <p><b>Maulana</b> ★★★★★ 4 2 spk</p>			


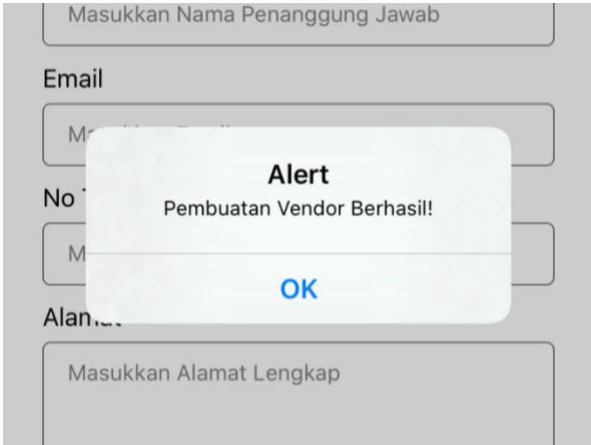
Tabel 8.9 Sortir Daftar Vendor Berdasarkan Skor

ID Test Case: V2		Tanggal dibuat: 17 Mei 2020	
Nama Test Case: Sortir Daftar Vendor Berdasarkan Skor		Tujuan: Pengguna menampilkan daftar vendor berdasarkan urutan skor	
Tester: Alvian Eka Maulana		Route: <b>Vendor</b>	
No	Fungsi Test Case	Hasil yang diharapkan	Status
1.	Menekan tombol <b>Skor</b>	Menampilkan urutan daftar vendor sesuai skor	Sukses





Contoh Luaran:



Tabel 8.10 Menambah Vendor Baru

<b>ID Test Case: V3</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : Menambah Vendor Baru		Tujuan: Pengguna menambahkan vendor baru	
Tester: Alvian Eka Maulana		Route: <b>Vendor</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Menekan logo 	Menampilkan formulir isian untuk menambah vendor	Sukses
2.	Mengisi nama	Kolom nama sudah terisi	Sukses
3.	Mengisi email	Kolom email sudah terisi	Sukses
4.	Mengisi nomor telepon	Kolom nomor telepon sudah terisi	Sukses
5.	Mengisi alamat	Kolom alamat sudah terisi	Sukses
6.	Menekan tombol <b>Submit</b>	Akun staf baru dibuat	Sukses
Contoh Luaran:			
			

Tabel 8.11 Menampilkan Detail Vendor

<b>ID Test Case: V4</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : <b>Menampilkan Detail Vendor</b>		Tujuan: Pengguna menampilkan informasi detail tentang vendor	
Tester: Alvian Eka Maulana		Route: <b>Vendor</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Menekan salah satu Vendor	Menampilkan identitas, spk, dan skor vendor	Sukses
Contoh Luaran:			
<div style="border: 1px solid #ccc; padding: 10px; background-color: #f9f9f9;"> <div style="display: flex; justify-content: space-between; align-items: flex-start;"> <div style="width: 70%;"> <p><b>Vendor Maulana</b></p> <hr style="width: 20%; margin-left: 0;"/> <p>Perum Kodam V No. 2</p> <p> <a href="mailto:maulana@gmail.com">maulana@gmail.com</a></p> <p> <a href="tel:081236482472">081236482472</a></p> <p><b>Total SPK ada 2</b></p> <p>SPK Selesai: 1            SPK Berjalan: 1            SPK Tepat Waktu: 0            SPK Telat: 1</p> </div> <div style="width: 25%; text-align: center;">  <p>4</p> </div> </div> <div style="display: flex; justify-content: space-around; margin-top: 10px;"> <div style="border: 1px solid #f44336; border-radius: 10px; padding: 5px 15px; display: flex; align-items: center;"> <span>Nama</span> <span style="font-size: 0.8em; margin-left: 5px;">▼</span> </div> <div style="border: 1px solid #f44336; border-radius: 10px; padding: 5px 15px; display: flex; align-items: center;"> <span>Waktu</span> <span style="font-size: 0.8em; margin-left: 5px;">▼</span> </div> <div style="border: 1px solid #f44336; border-radius: 10px; padding: 5px 15px; display: flex; align-items: center;"> <span>Skor</span> <span style="font-size: 0.8em; margin-left: 5px;">▼</span> </div> </div> <div style="margin-top: 10px; padding: 10px 0 0 0;"> <p style="color: #f44336; margin: 0;"><u>20 pcs</u></p> <div style="display: flex; justify-content: space-between; align-items: center;">  <div style="text-align: left;"> <p><b>Afternoon Dress</b></p> <p>tgl pemesanan : 26 May 2020</p> <p>tgl selesai : 01 Jun 2020</p> <p><span style="color: #f44336;">★★★★★</span> 4</p> </div> <div style="background-color: #212121; color: white; padding: 5px 15px; border-radius: 10px; font-weight: bold;">Selesai</div> </div> </div> </div>			

Tabel 8.12 Menghubungi Nomor Telepon Vendor

<b>ID Test Case: V5</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : Menghubungi Nomor Telepon Vendor		Tujuan: Pengguna menghubungi vendor via nomor telepon	
Tester: Alvian Eka Maulana		Route: <b>Vendor -&gt; Vendor Detail</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Menekan nomor telepon	Masuk ke fitur telepon di <i>smartphone</i>	Sukses
<p>Contoh Luaran:</p> 			



## SPK

Tabel 8.13 Menampilkan Daftar SPK

<b>ID Test Case: S1</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : Menampilkan Daftar SPK		Tujuan: Pengguna menampilkan daftar spk yang berisi nama vendor, koleksi, tanggal pesan, tanggal selesai, foto, dan status	
Tester: Alvian Eka Maulana		Route: <b>Home</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Menekan tombol <b>SPK</b>	Pengguna masuk ke halaman spk	Sukses

Contoh Luaran:

**SPK** +

Nama ▼    Waktu ▼    Lainnya ▼

2 SPK ditampilkan

**Selesai**

Vendor Maulana

**Afternoon Dress**

tgl pemesanan : 26 May 2020

tgl selesai : 01 Jun 2020

**Berjalan**

Vendor Maulana

**Fine Lace**

tgl pemesanan : 25 May 2020

estimasi selesai : 03 Jun 2020

Tabel 8.14 Sortir Daftar SPK Berdasarkan Vendor

<b>ID Test Case: S2</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : Sortir Daftar SPK Berdasarkan Vendor		Tujuan: Pengguna menampilkan daftar spk berdasarkan vendor	
Tester: Alvian Eka Maulana		Route: <b>SPK</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Menekan tombol <b>Lainnya</b>	Menampilkan pop-up yang berisi status-status spk dan nama vendor	Sukses
2.	Menekan tombol salah satu vendor	Menampilkan urutan daftar spk sesuai vendor pilihan	Sukses

Contoh Luaran:

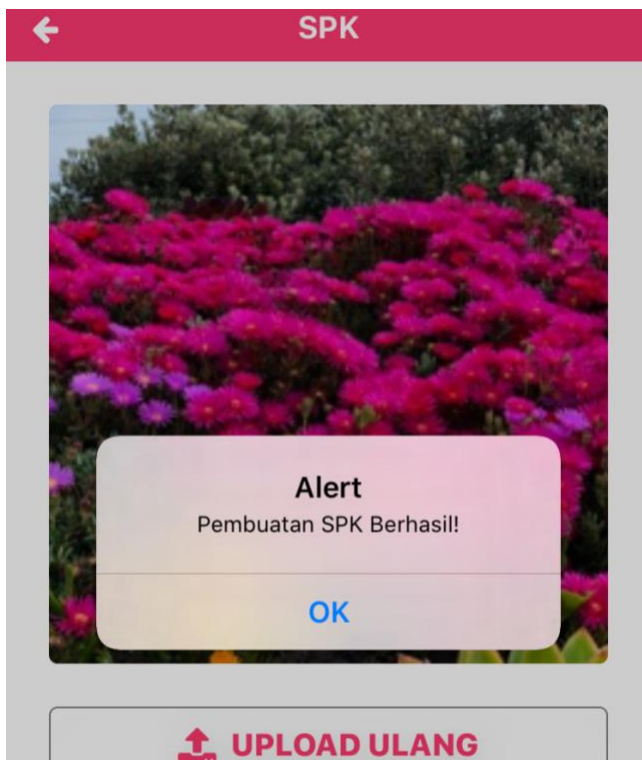


Tabel 8.15 Menambah Draf SPK Baru

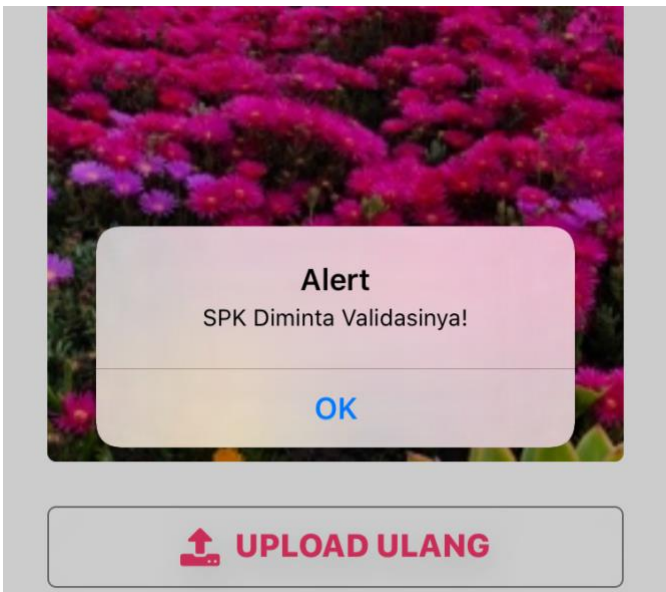
<b>ID Test Case: S3</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : Menambah Draf SPK Baru		Tujuan: Pengguna menambahkan Draf SPK baru	
Tester: Alvian Eka Maulana		Route: <b>SPK</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Menekan logo 	Menampilkan formulir isian untuk menambah SPK	Sukses
2.	Mengisi nama koleksi	Kolom nama koleksi sudah terisi	Sukses
3.	Memilih vendor	Kolom vendor sudah terpilih	Sukses
4.	Memilih jenis koleksi	Kolom jenis koleksi sudah terpilih	Sukses
5.	Mengisi durasi cutting	Kolom durasi cutting sudah terisi	Sukses
6.	Mengisi durasi jahit	Kolom durasi jahit sudah terisi	Sukses
7.	Mengisi durasi finishing	Kolom durasi finishing sudah terisi	Sukses
8.	Mengisi detail aksesoris	Kolom detail aksesoris sudah terisi	Sukses
9.	Mengisi nama varian	Kolom nama varian sudah terisi	Sukses
10.	Memilih satuan panjang kain	Kolom satuan Panjang kain sudah terpilih	Sukses
11.	Mengisi panjang kain	Kolom Panjang kain sudah terisi	Sukses
12.	Mengisi ukuran XS	Kolom ukuran XS sudah terisi	Sukses

13.	Mengisi ukuran S	Kolom ukuran S sudah terisi	Sukses
14.	Mengisi ukuran M	Kolom ukuran M sudah terisi	Sukses
15.	Mengisi ukuran L	Kolom ukuran L sudah terisi	Sukses
16.	Mengisi ukuran XL	Kolom ukuran XL sudah terisi	Sukses
17.	Mengisi ukuran XXL	Kolom ukuran XXL sudah terisi	Sukses
18.	Menekan tombol <b>Simpan Draf</b>	Draf SPK baru dibuat	Sukses

Contoh Luaran:



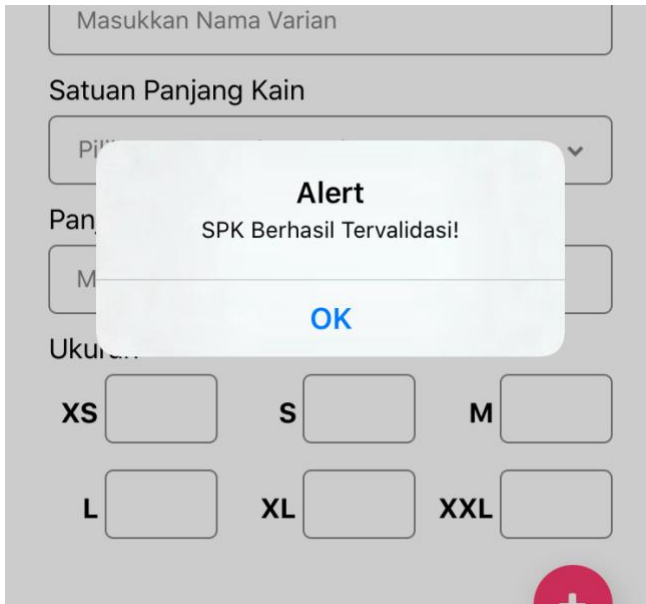
Tabel 8.16 Meminta Validasi SPK

<b>ID Test Case: S4</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : Meminta Validasi SPK		Tujuan: Pengguna meminta validasi SPK	
Tester: Alvian Eka Maulana		Route: <b>Home</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Menekan tombol <b>Draf SPK</b>	Menampilkan halaman daftar SPK yang berstatus Draf	Sukses
2.	Menekan salah satu SPK	Menampilkan detail SPK	Sukses
3.	Menekan tombol <b>Minta Validasi</b>	Status Draf SPK berubah menjadi Validasi	Sukses
Contoh Luaran:			
			

Tabel 8.17 Melakukan Validasi SPK

<b>ID Test Case: S5</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : Melakukan Validasi SPK		Tujuan: Pengguna melakukan validasi SPK	
Tester: Alvian Eka Maulana		Route: <b>Home</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Menekan tombol <b>Validasi SPK</b>	Menampilkan halaman daftar SPK yang berstatus Validasi	Sukses
2.	Menekan salah satu SPK	Menampilkan detail SPK	Sukses
3.	Menekan tombol <b>Kirim Ke Vendor</b>	Status Validasi SPK berubah menjadi Berjalan	Sukses

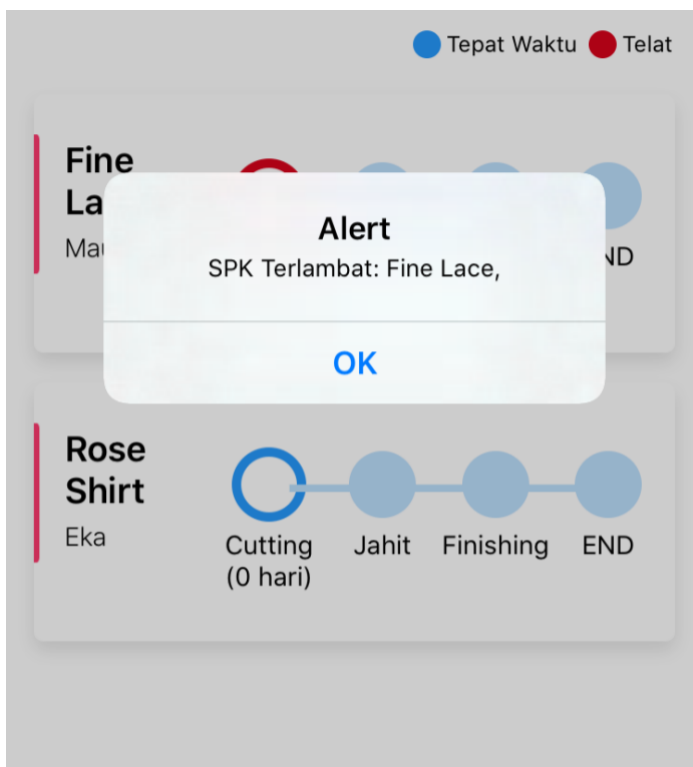
Contoh Luaran:



Tabel 8.18 Melihat SPK Yang Terlambat

<b>ID Test Case: S6</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : Melihat SPK Yang Terlambat		Tujuan: Pengguna melihat SPK mana saja yang terlambat pengerjaannya	
Tester: Alvian Eka Maulana		Route: <b>Home</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Menekan tombol <b>Beranda</b>	Menampilkan halaman beranda dan <i>pop-up</i> SPK yang terlambat	Sukses

Contoh Luaran:



Tabel 8.19 Melihat Detail Koleksi SPK Berjalan

<b>ID Test Case: S7</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : Melihat Detail Koleksi SPK Berjalan		Tujuan: Pengguna melihat Detail Koleksi SPK Berjalan yaitu jumlah barang, detail aksesoris, foto, dan jumlah varian	
Tester: Alvian Eka Maulana		Route: <b>Home</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Menekan salah satu <b>SPK</b>	Menampilkan halaman detail koleksi SPK	Sukses

Contoh Luaran:

### Fine Lace - Maulana

tgl pemesanan : 25 May 2020  
 estimasi selesai : 03 Jun 2020

Detail  
Koleksi

Progress  
SPK

Penerimaan  
Barang

Evaluasi  
Vendor

### Fine Lace (Putih, Hitam)

16 pcs



#### Detail Aksesoris

Percikan berlian samudera

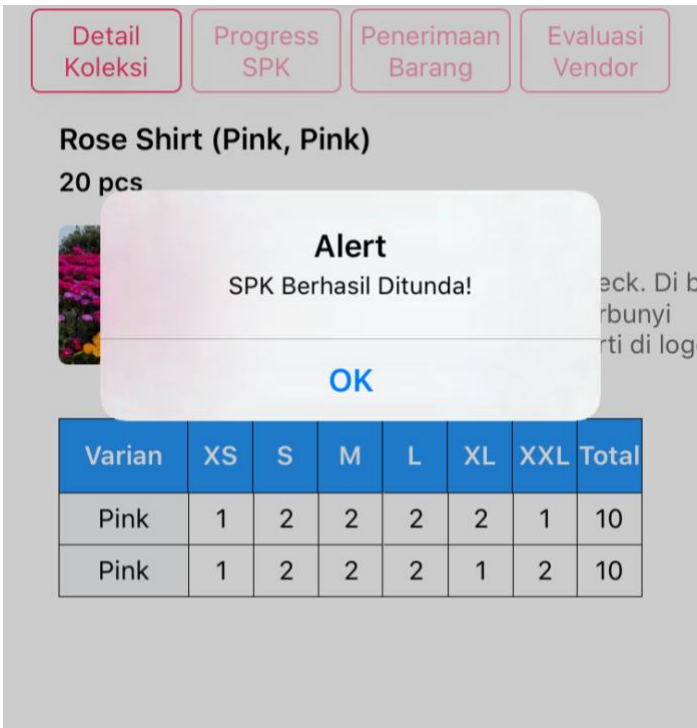
Varian	XS	S	M	L	XL	XXL	Total
Putih	2	2	2	1	2	1	10
Hitam	1	1	1	1	1	1	6



Tabel 8.20 Menunda SPK Berjalan

<b>ID Test Case: S8</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : Menunda SPK Berjalan		Tujuan: Pengguna melakukan penundaan SPK yang sedang berjalan	
Tester: Alvian Eka Maulana		Route: <b>Home</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Menekan salah satu <b>SPK</b>	Menampilkan halaman detail koleksi SPK	Sukses
2.	Menekan tombol <b>Tunda SPK</b>	SPK ditunda	Sukses

Contoh Luaran:



The screenshot shows a mobile application interface for a collection of 'Rose Shirt (Pink, Pink)'. At the top, there are four buttons: 'Detail Koleksi', 'Progress SPK', 'Penerimaan Barang', and 'Evaluasi Vendor'. Below the buttons, the product name 'Rose Shirt (Pink, Pink)' and quantity '20 pcs' are displayed. A size selection table is visible at the bottom, with columns for 'Varian', 'XS', 'S', 'M', 'L', 'XL', 'XXL', and 'Total'. An alert dialog box is overlaid on the screen, displaying the message 'Alert' and 'SPK Berhasil Ditunda!' with an 'OK' button.

Varian	XS	S	M	L	XL	XXL	Total
Pink	1	2	2	2	2	1	10
Pink	1	2	2	2	1	2	10

Tabel 8.21 Melanjutkan SPK Yang Ditunda

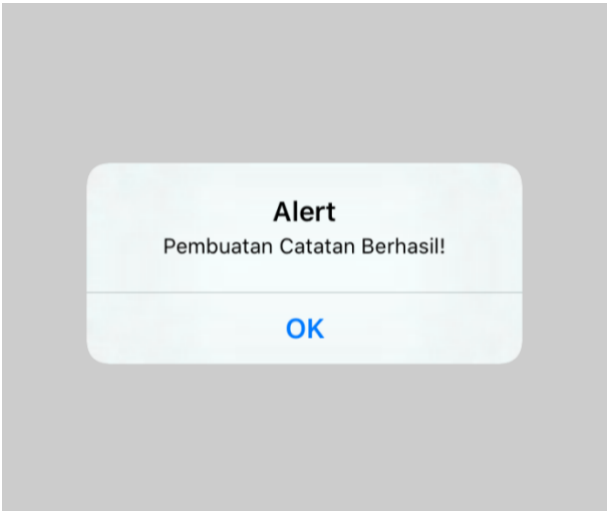
<b>ID Test Case: S9</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : Melanjutkan SPK Yang Ditunda		Tujuan: Pengguna melanjutkan kembali SPK yang ditunda	
Tester: Alvian Eka Maulana		Route: <b>SPK</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Menekan salah satu <b>SPK dengan status Pending</b>	Menampilkan halaman detail koleksi SPK	Sukses
2.	Menekan tombol <b>Tunda SPK</b>	SPK ditunda	Sukses

Contoh Luaran:

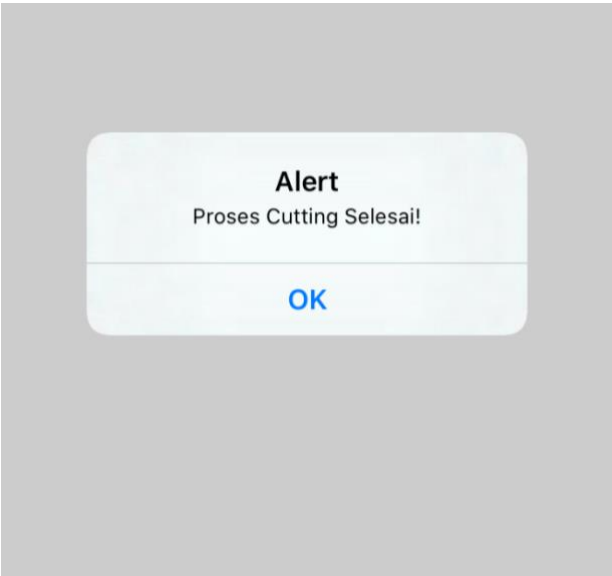
The screenshot displays a mobile application interface for managing a collection of items. At the top, there are four navigation buttons: 'Detail Koleksi', 'Progress SPK', 'Penerimaan Barang', and 'Evaluasi Vendor'. Below these, the main content area shows a collection titled 'Rose Shirt (Pink, Pink)' with a quantity of '20 pcs'. An alert dialog box is overlaid on the screen, with the text 'Alert' and 'SPK Dilanjutkan Kembali!' (SPK Resumed). Below the alert is an 'OK' button. In the background, a table is visible with the following structure:

Variasi	XS	S	M	L	XL	XXL	Total
Pink	1	2	2	2	2	1	10
Pink	1	2	2	2	2	1	10

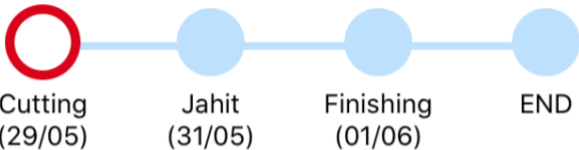
Tabel 8.22 Menambahkan Catatan Proses SPK

<b>ID Test Case: S10</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : Menambahkan Catatan Proses SPK		Tujuan: Pengguna menambahkan catatan di salah satu proses SPK	
Tester: Alvian Eka Maulana		Route: <b>Home</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Menekan salah satu <b>SPK</b>	Menampilkan halaman detail koleksi SPK	Sukses
2.	Menekan tab <b>Progress SPK</b>	Menampilkan halaman Progress SPK	Sukses
3.	Mengisi catatan	Kolom catatan terisi	Sukses
4.	Menekan tombol <b>Submit</b>	Catatan tersimpan	Sukses
Contoh Luaran:			
 <p>The image shows a screenshot of a mobile application's alert dialog. The dialog is a light blue rounded rectangle centered on a grey background. It contains the text "Alert" in bold black font, followed by "Pembuatan Catatan Berhasil!" in a smaller black font. At the bottom of the dialog is a blue "OK" button.</p>			

Tabel 8.23 Menyelesaikan Proses SPK

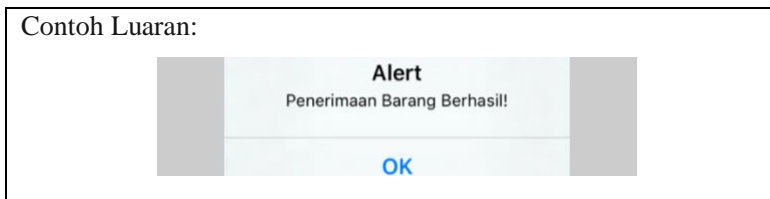
<b>ID Test Case: S11</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : Menyelesaikan Proses SPK		Tujuan: Pengguna menyelesaikan salah satu proses SPK	
Tester: Alvian Eka Maulana		Route: <b>Home</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Menekan salah satu <b>SPK</b>	Menampilkan halaman detail koleksi SPK	Sukses
2.	Menekan tab <b>Progress SPK</b>	Menampilkan halaman Progress SPK	Sukses
3.	Menekan tombol <b>Selesaikan Cutting</b>	Proses Cutting selesai	Sukses
Contoh Luaran:			
 <p>The image shows a light blue alert dialog box centered on a grey background. The dialog box has rounded corners and contains the text "Alert" in bold black font, followed by "Proses Cutting Selesai!" in a smaller black font. At the bottom of the dialog box, there is a blue "OK" button.</p>			

Tabel 8.24 Melihat Proses SPK Yang Telah Selesai

<b>ID Test Case: S12</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : Melihat Proses SPK Yang Telah Selesai		Tujuan: Pengguna melihat kembali proses SPK yang sudah diselesaikan	
Tester: Alvian Eka Maulana		Route: <b>Home</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Menekan salah satu <b>SPK</b>	Menampilkan halaman detail koleksi SPK	Sukses
2.	Menekan tab <b>Progress SPK</b>	Menampilkan halaman Progress SPK	Sukses
3.	Menekan tombol <b>Lihat Cutting</b>	Menampilkan detail proses Cutting	Sukses
<p>Contoh Luaran:</p>  <p><b>Catatan Step Cutting:</b></p> <p>yudha: "Ada kain yg robek" 27/05 3:06</p> <p>yudha: "tes lagi" 27/05 3:20</p> <div style="border: 1px solid #ccc; padding: 5px; margin-top: 10px; width: fit-content;"> <p>Tambahkan Catatan</p> </div>			

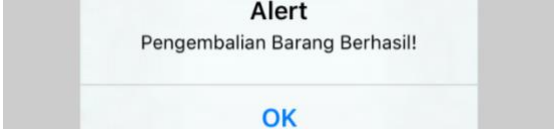
Tabel 8.25 Menerima Barang SPK

<b>ID Test Case: S13</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : Menerima Barang SPK		Tujuan: Pengguna menerima Barang SPK	
Tester: Alvian Eka Maulana		Route: <b>Home</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Menekan salah satu <b>SPK</b>	Menampilkan halaman detail koleksi SPK	Sukses
2.	Menekan tab <b>Penerimaan Barang</b>	Menampilkan halaman Penerimaan Barang	Sukses
3.	Menekan tombol <b>Terima</b> di salah satu varian	Menampilkan <i>pop-up</i> kolom jumlah barang yang akan diterima di setiap jenis ukuran	Sukses
4.	Mengisi ukuran XS	Kolom ukuran XS terisi	Sukses
5.	Mengisi ukuran S	Kolom ukuran S terisi	Sukses
6.	Mengisi ukuran M	Kolom ukuran M terisi	Sukses
7.	Mengisi ukuran L	Kolom ukuran L terisi	Sukses
8.	Mengisi ukuran XL	Kolom ukuran XL terisi	Sukses
9.	Mengisi ukuran XXL	Kolom ukuran XXL terisi	Sukses
10.	Menekan tombol <b>Submit</b>	Jumlah barang yang diterima sudah tersimpan	Sukses



**Tabel 8.26 Mengembalikan Barang SPK**

<b>ID Test Case: S14</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : Mengembalikan Barang SPK		Tujuan: Pengguna mengembalikan Barang SPK	
Tester: Alvian Eka Maulana		Route: <b>Home</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Menekan salah satu <b>SPK</b>	Menampilkan halaman detail koleksi SPK	Sukses
2.	Menekan tab <b>Penerimaan Barang</b>	Menampilkan halaman Penerimaan Barang	Sukses
3.	Menekan tombol <b>Terima</b> di salah satu varian	Menampilkan <i>pop-up</i> kolom jumlah barang yang akan dikembalikan di setiap jenis ukuran	Sukses
4.	Mengisi ukuran XS	Kolom ukuran XS terisi	Sukses
5.	Mengisi ukuran S	Kolom ukuran S terisi	Sukses
6.	Mengisi ukuran M	Kolom ukuran M terisi	Sukses
7.	Mengisi ukuran L	Kolom ukuran L terisi	Sukses
8.	Mengisi ukuran XL	Kolom ukuran XL terisi	Sukses

9.	Mengisi ukuran XXL	Kolom ukuran XXL terisi	Sukses
10.	Menekan tombol <b>Submit</b>	Jumlah barang yang dikembalikan sudah tersimpan	Sukses
Contoh Luaran:			
			

Tabel 8.27 Menutup SPK

<b>ID Test Case: S15</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : Menutup SPK		Tujuan: Pengguna menutup SPK	
Tester: Alvian Eka Maulana		Route: <b>Home</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Menekan salah satu <b>SPK</b>	Menampilkan halaman detail koleksi SPK	Sukses
2.	Menekan tab <b>Penerimaan Barang</b>	Menampilkan halaman Penerimaan Barang	Sukses
3.	Menekan tombol <b>Tutup SPK</b>	SPK ditutup	Sukses
Contoh Luaran:			
			



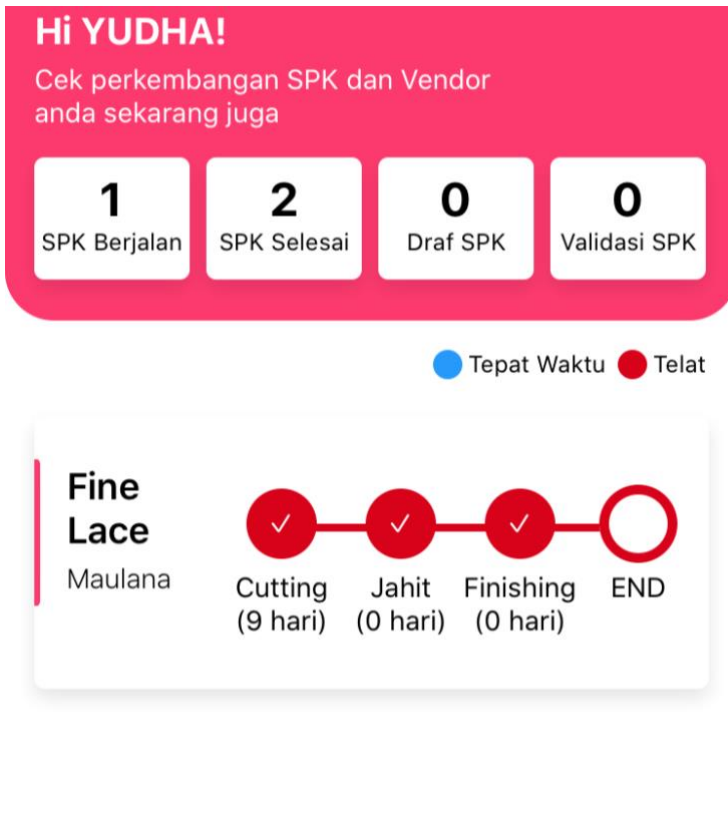
Tabel 8.28 Mengevaluasi Vendor

<b>ID Test Case: S16</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : Mengevaluasi Vendor		Tujuan: Pengguna melakukan evaluasi Vendor	
Tester: Alvian Eka Maulana		Route: <b>Home</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Menekan salah satu <b>SPK</b>	Menampilkan halaman detail koleksi SPK	Sukses
2.	Menekan tab <b>Evaluasi Vendor</b>	Menampilkan halaman Evaluasi Vendor	Sukses
3.	Memilih bintang penilaian di baris Kerapihan	Kerapihan dinilai	Sukses
4.	Memilih bintang penilaian di baris Kualitas Pengerjaan	Kualitas Pengerjaan dinilai	Sukses
5.	Memilih bintang penilaian di baris Ketepatan Waktu	Ketepatan Waktu dinilai	Sukses
6.	Memilih bintang penilaian di baris Kesesuaian Jumlah	Kesesuaian Jumlah dinilai	Sukses
7.	Mengisi komentar	Kolom komentar terisi	Sukses
8.	Menekan tombol <b>Submit</b>	Vendor dinilai	Sukses
Contoh Luaran:			
			

Tabel 8.29 Melihat Progress SPK

<b>ID Test Case: S17</b>		Tanggal dibuat: 17 Mei 2020	
Nama <i>Test Case</i> : Melihat Progress SPK		Tujuan: Pengguna melihat Progress SPK saat ini	
Tester: Alvian Eka Maulana		Route: <b>Home</b>	
No	Fungsi <i>Test Case</i>	Hasil yang diharapkan	Status
1.	Menekan tombol <b>Beranda</b>	Menampilkan halaman beranda	Sukses

Contoh Luaran:



## LAMPIRAN C. PROTOKOL PENGUJIAN

Tabel 8.30 Protokol Pengujian

Protokol	Deskripsi
Penulis	Alvian Eka Maulana
Kontak	alvian.em169@gmail.com
Produk yang diuji	Aplikasi <i>frontend</i> sistem manajemen vendor Giyomi.id
Tujuan pengujian	Menguji keefektifan aplikasi yang dibangun
Peserta	Owner Giyomi.id yang berpengalaman dalam manajemen vendor di UMKM
Peralatan	Smartphone (Android atau IOS) Catatan Dokumen <i>Test Case</i> Aplikasi <i>Whatsapp</i> (Khusus Iterasi Kedua)
Tugas	Alvian Eka Maulana (moderator dan mengontak peserta)
Prosedur Pengujian	Terima kasih telah bergabung dalam uji coba aplikasi ini. Tujuan dari kegiatan ini adalah untuk menguji efektifitas tampilan aplikasi Manajemen Vendor Giyomi. Proses produksi dari Giyomi dikerjakan oleh vendor, sehingga aplikasi ini akan membantu Giyomi dalam proses produksi di vendor tersebut. Diharapkan dari pengiriman SPK (surat perintah kerja) kepada vendor, kemudian vendor akan melakukan produksi sesuai dengan detail permintaan yang tersampaikan di SPK tersebut. Ada 3 tahap dalam proses SPK yaitu tahap pemotongan, menjahit dan finishing. Jika produk sudah jadi, vendor akan mengirimkan barang ke gudang Giyomi untuk dilakukan Quality Control. Anda akan diberikan 12 test case yang berisi beberapa poin kegiatan yang harus Anda lakukan di dalam aplikasi ini secara urut.

	<p>Anda tidak dapat meminta bantuan yang disediakan untuk moderator karena tugas moderator adalah mencatat semua kesulitan yang Anda alami selama menjalankan test case tersebut.</p> <p>Jika Anda merasa bingung dan yakin tidak berhasil menyelesaikan kasus, maka Anda akan diizinkan untuk menyelesaikan dan membuktikan kasus tersebut.</p> <p>Diakhir sesi, moderator akan meminta evaluasi tentang aplikasi yang telah Anda coba. Hasil dari pengujian ini akan digunakan sebagai masukan untuk perbaikan selanjutnya dari desain dan fungsi aplikasi.</p> <p>Sebelum kami memulai pengerjaan tugas, moderator akan meminta Anda untuk membaca <i>test case</i>. Anda dapat mengajukan pertanyaan jika ada <i>test case</i> yang kurang jelas.</p>
--	---

## LAMPIRAN D. FEEDBACK PENGGUNA

### ITERASI PERTAMA

Pengguna : Yudha Prasetyo

Posisi : Owner Giyomi.id

**Tabel 8.31 Feedback Pengguna Iterasi Pertama**

UI ID	UI Name	Feedback
UI1	Login	<ul style="list-style-type: none"> <li>Sudah bagus sekali tampilannya, pemilihan warna yang cerah dan pemberian logo Giyomi yang <i>eye catching</i></li> </ul>
UI2	Home	<ul style="list-style-type: none"> <li><i>Pop Up</i> pemberitahuan dibutuhkan jika ada SPK yang telat, jadi tidak perlu menunggu melihat detail SPK terlebih dahulu</li> <li>Sebaiknya tampilan beranda ini dibuat seperti aplikasi lain yang memberikan <i>greetings</i> di awal, sehingga tidak terlalu kaku</li> </ul>
UI4	Add Vendor	<ul style="list-style-type: none"> <li>Ketika menyelesaikan proses penambahan sebaiknya diberikan <i>Pop Up</i> untuk memberitahu apakah proses berjalan sukses atau gagal</li> </ul>
UI6	SPK	<ul style="list-style-type: none"> <li>Pilihan sortir SPK hanya tersedia nama dan waktu, harusnya ada yang berdasarkan status dan nama vendor</li> </ul>
UI7	Add SPK	<ul style="list-style-type: none"> <li>Ukurannya dibuat standar saja, yaitu XS, S, M, L, XL, XXL</li> <li>Setiap SPK harusnya bisa mempunyai berbagai macam</li> </ul>

		<p>varian. Disini masih belum bisa menambahkan varian sesuai selera</p> <ul style="list-style-type: none"> <li>• Untuk pilihan satuan panjang kain dibuat meter dan yard saja. Jadi ketika pengguna membuat SPK tinggal memilih antara dua itu</li> <li>• Setiap SPK sudah pasti ada proses Cutting, Jahit, dan Finishing</li> <li>• Upload gambar hanya 1 saja di setiap SPK</li> <li>• Saat memilih vendor harusnya ada skor disamping nama vendor sehingga bisa memilih vendor dengan performa paling bagus</li> </ul>
UI9	SPK Progress	<ul style="list-style-type: none"> <li>• Belum bisa menambahkan catatan lebih dari satu</li> </ul>
UI10	SPK Received	<ul style="list-style-type: none"> <li>• Ketika mengambil barang, harusnya ada <i>Pop Up</i> terlebih dahulu untuk memasukkan ukuran <i>real</i> yang diambil, jadi tidak langsung semua</li> </ul>
UI17	Account	<ul style="list-style-type: none"> <li>• Sudah bagus karena ada fotonya juga</li> </ul>

## ITERASI KEDUA

Pengguna : Yudha Prasetyo  
 Posisi : Owner Giyomi.id

**Tabel 8.32 Feedback Pengguna Iterasi Kedua**

UI ID	UI Name	Feedback
UI2	Home	<ul style="list-style-type: none"> <li>• Sudah sesuai dengan kebutuhan Giyomi, tetapi loading nya</li> </ul>

		<p>lama sekali dibandingkan halaman lainnya. Ini terjadi ketika SPK yang sedang berjalan sudah lebih dari 10. Proses loading untuk menampilkan setiap SPK bisa mencapai 5 detik. Padahal sebelumnya hanya 1 detik.</p>
UI3	Vendor	<ul style="list-style-type: none"> <li>• Bagus ketika tampilan skor di bagian bintang sudah terisi. Jadi lebih mudah ketika ingin mengetahui kualitas suatu vendor. Cukup lihat dari bintangnya</li> </ul>
UI5	Vendor Detail	<ul style="list-style-type: none"> <li>• Sudah ada sortir daftar SPK di halaman Detail Vendor berdasarkan skor, sebelumnya belum ada</li> </ul>
UI6	SPK	<ul style="list-style-type: none"> <li>• Proses sortir daftar SPK jadi lebih mudah ketika sudah ada pilihan sortir lainnya, yaitu berdasarkan status dan nama vendor</li> </ul>
UI7	Add SPK	<ul style="list-style-type: none"> <li>• Ketika menekan tombol Simpan Draf dan Minta Validasi, ada peringatan terlebih dahulu sehingga tidak takut lagi kalau tidak sengaja tertekan.</li> </ul>
UI9	SPK Progress	<ul style="list-style-type: none"> <li>• Bisa menambahkan banyak catatan, ini memudahkan karena biasanya sewaktu-waktu bisa menambahkan catatan</li> </ul>
UI10	SPK Received	<ul style="list-style-type: none"> <li>• Paling lama menghabiskan waktu di halaman ini. Karena memasukkan setiap jumlah</li> </ul>

		barang yang diambil atau dikembalikan
--	--	---------------------------------------



## BIODATA PENULIS



Penulis dengan nama lengkap Alvian Eka Maulana ini bertempat tinggal di Kabupaten Sidoarjo. Anak pertama dari dua bersaudara ini lahir pada tanggal 16 September 1998. Selama menjalani masa perkuliahan di Departemen Sistem Informasi ITS, penulis aktif sebagai anggota UKM Robotika di tahun pertama.

Anggota aktif di Departemen RISTEK HMSI, Departemen MEDFO BEM FTIF, Divisi IT Development ISE, Divisi IT Development FTIF Festival, Divisi Web Design & Documentation YES Summit, Divisi Web Development ITS Expo pada tahun kedua. Pada tahun ketiga, penulis aktif di Tim Robot Barunastra ITS, Kementerian Kominfo BEM ITS, dan Divisi Publikasi INI LHO ITS. Sedangkan di tahun keempatnya, penulis sangat aktif di software house seperti PT. Solusi Master dan PT. Poros Kolektif Nusantara, aktif juga di startup seperti Olimpiade.id, Diakad, dan juga Alinamed Indonesia. Selain itu penulis juga aktif mengerjakan tugas akhir agar bisa segera lulus dari Departemen Sistem Informasi ITS.

Tugas Akhir ini tak lepas dari kritik dan saran. Jika ada yang ingin didiskusikan ataupun ditanyakan, jangan ragu untuk menghubungi penulis via email: [alvian.em169@gmail.com](mailto:alvian.em169@gmail.com). Terimakasih dan semoga Tugas Akhir ini bermanfaat.