

# BAB 1

## SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK

### 1.1 Pendahuluan

PT. Multidaya Dinamika salah satu perusahaan yang berinovasi dan berorientasi pada teknologi, lingkungan kerja di perusahaan ini menerapkan *open space*. Seperti yang dijelaskan oleh Hedge (1982) bahwa istilah *open space* mengacu pada penataan ruang kantor yang lebih fleksibel serta tidak ada dinding yang memisahkan atau pemisah hanya berupa partisi rendah yang memudahkan aktivitas karyawan. Dengan adanya *open space* yang diterapkan pada perusahaan memiliki beberapa keuntungan dalam hal kolaborasi tim dan komunikasi yang lebih terbuka, tetapi juga terdapat keterbatasan pada penyimpanan barang karyawan. Untuk itu diperlukan kotak penyimpanan pintar yang menggunakan kunci elektronik yang lebih praktis dan dapat mengurangi resiko kehilangan kunci (Agus & Durbin, 2023). Kami mencoba merancang pengembangan kotak penyimpanan pintar sederhana dengan interaksi yang aman dan nyaman untuk karyawan (Pendke, Nadar, Ramteke, Rokade M, Rokade B, 2020).

Kotak penyimpanan pintar berbasis kiosk adalah unit kotak penyimpanan yang mampu menyimpan barang yang terletak di lokasi yang nyaman untuk para karyawan. Pada sistem ini dapat menciptakan teknologi yang efisien, aman, dan mudah digunakan dalam menyimpan barang-barang atau dokumen. Kotak penyimpanan pintar dapat membantu dalam ruang kerja yang terbuka, menjaga keamanan dalam mengakses barang-barang atau dokumen.



**Gambar 1. Kotak Penyimpanan Pintar**

Pengembangan sistem kotak penyimpanan pintar berbasis kiosk merupakan salah satu solusi untuk di lingkungan kerja yang menerapkan *open space*. Solusi ini dapat

memberikan kenyamanan bagi para karyawan dalam menyimpan barang-barang atau dokumen. Karyawan dapat mengambil barang-barang mereka pada saat jam pulang kantor dengan mengakses terminal pintar yang terhubung dengan sistem. Dengan kiosk menggunakan kode verifikasi dan kode pin. Kotak penyimpanan pintar berbasis kiosk menawarkan pendekatan yang terintegrasi dan otomatis dalam menaruh barang-barang pribadi. Dengan menggunakan teknologi kunci elektronik ini dapat memungkinkan karyawan untuk menyimpan barang mereka dengan aman dan nyaman. Penggunaan kotak penyimpanan pintar berbasis kiosk ini juga dapat meminimalkan tempat penyimpanan barang dalam lingkungan kerja yang menerapkan *open space*.

### **1.1.1 Siklus Hidup Pengembangan Aplikasi**

Aplikasi *KeyKeep* merupakan sistem yang dapat membantu karyawan PT. Multidaya Dinamika dalam menyimpan barang-barang di loker. Aplikasi ini memungkinkan karyawan memilih ukuran loker, nomor loker dan membuat kode PIN sebanyak 4 digit. Admin dapat mengelola loker melalui *dashboard*, termasuk mengubah jumlah loker dan status loker. Aplikasi ini dirancang untuk memudahkan dalam menyimpan barang di lingkungan kantor yang menerapkan *open space*. Dalam menghasilkan Aplikasi *KeyKeep*, tim DWP memilih Metodologi *Feature-Driven-Development* (FDD) sebagai landasan utama. Metodologi FDD ini dipilih karena cocok, dimana semua dirancang dengan struktur yang teratur sehingga dapat dirancang dengan baik. Pada FDD ini kamu memfokuskan dalam menemukan fitur yang diinginkan oleh pengguna atau *stakeholder*.

Tahapan-tahapan dalam Metodologi *Feature-Driven-Development* (FDD) mencakup:

1. Pengembangan model:
  - Fokus mengembangkan model secara keseluruhan dari sistem yang akan dikembangkan.
2. Pembuatan daftar fitur:
  - Tim mengidentifikasi dan membuat daftar fitur-fitur yang akan dikembangkan dalam proyek.
3. Pembuatan rencana berdasarkan fitur:
  - Tim merencanakan pengembangan dengan memprioritaskan fitur-fitur yang akan dikembangkan secara iterative untuk setiap fitur yang dipilih.
4. Pembuatan desain berdasarkan fitur:

- Dalam hal ini melibatkan pembuatan desain, diagram dan spesifikasi teknis yang diperlukan.
5. Pembuatan produk berdasarkan fitur:
- Setiap fitur dikerjakan secara terpisah dan diintegrasikan ke dalam sistem secara bertahap.



**Gambar 2. Metodologi *Feature-Driven-Development* (FDD)**

Tahapan-tahapan dalam metodologi *Feature-Driven-Development* (FDD) merupakan pendekatan yang memungkinkan pengembangan perangkat lunak secara terstruktur, fokus terhadap pengembangan fitur-fitur yang jelas, dan mengintegrasikan proses desain, pengujian dan pengembangan secara efisien. Metodologi ini dapat membantu dalam mengurangi risiko, meningkatkan fokus, dan memastikan pengembangan perangkat lunak sesuai dengan kebutuhan dan harapan pengguna (*stakeholder*).

### 1.1.2 Tujuan Penulisan Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi Kebutuhan Perangkat Lunak (SKPL) pada sistem kotak penyimpan pintar berbasis kiosk memiliki tujuan bagi pengembang yaitu mencakup efisiensi, keamanan, mudah diintegrasikan, antarmuka yang intuitif. Bagi pengguna, tujuannya untuk memberikan aksesibilitas yang ramah, cepat, mudah, fungsionalitas yang jelas dan layanan berkualitas tinggi.

### 1.1.3 Cakupan Produk

Aplikasi *KeyKeep* akan dikembangkan untuk digunakan karyawan menyimpan barang-barang mereka karena perusahaan PT. Multiday Dinamika menggunakan konsep *open space* yang dimana penataan ruang kantor dapat menjadi lebih fleksibel, serta tidak ada dinding yang memisahkan (Hedge, 1982). Panel control yang digunakan admin untuk mengelola dan mengkonfigurasi berbagai aspek pada aplikasi dimanakan *dashboard*.

*Dashboard* tidak hanya berfungsi sebagai pusat *monitoring* atau pemantauan tetapi juga sebagai alat untuk melakukan pengaturan, modifikasi, dan pengelolaan pada aplikasi.

Fitur yang dibutuhkan pada pengguna yaitu titip barang dan ambil barang, pilih loker, pilih ukuran loker dan membuat kode pin sendiri dengan 4digit angka. Selain itu fitur yang dibutuhkan untuk untuk admin pada sebuah *dashboard* admin dapat mengubah jumlah, ukuran, status ketersediaan, informasi, ikon, warna, dan deskripsi loker.

Permasalahan yang dijabarkan dalam Aplikasi *KeyKeep* menjadi solusi untuk mempermudah karyawan untuk menyimpan barang dan mengambil barang dan diberikan kesempatan 3 kali untuk membuka loker jika ada barang yang lupa ditaruh di dalam loker. Pengembangan Aplikasi *KeyKeep* harus diselesaikan untuk memenuhi kebutuhan karyawan untuk menggunakan kotak penyimpanan pintar.

Dengan mengembangkan Aplikasi *KeyKeep* sejalan dengan firman Allah dalam Surah An-Nisa Ayat 58, Al-Maidah Ayat 8, dan Hadist Imam Muslim:

An-Nisa Ayat 58:

إِنَّ اللَّهَ يَأْمُرُكُمْ أَنْ تُؤَدُّوا الْأَمْانَاتِ إِلَى أَهْلِهَا وَإِذَا حَكَمْتُمْ بَيْنَ النَّاسِ أَنْ تَحْكُمُوا بِالْعَدْلِ إِنَّ اللَّهَ نِعِمَّا يَعْظُمُ بِهِ إِنَّ اللَّهَ كَانَ سَمِيعًا بَصِيرًا

**Artinya:** “Sesungguhnya Allah memerintahkan kepadamu supaya menyerahkan amanah kepada yang berhak menerimanya dan (juga) apabila kamu menetapkan hukum diantara manusia, supaya kamu menetapkan dengan adil. Sesungguhnya Allah memberikan pelajaran yang baik tentang apa yang patut kamu kerjakan. Sesungguhnya Allah Maha Mendengar lagi Maha Melihat”. (QS. An-Nisa (4): 58)

Tafsir Al Wajiz sesungguhnya Allah memerintahkan kalian untuk memberikan amanat setiap sesuatu yang diamanatkan seseorang yang merupakan hak orang lain, baik amanat yang diperuntukkan untuk Allah atau para hamba. Betapa nikmatnya sesuatu yang diajarkan (diperintahkan) oleh Allah kepada kalian yaitu menunaikan amanah dan menentukan hukum dengan adil (Syaikh Prof. Dr. Wahbah Az-Zuhaili, 2018). Menunaikan amanat berarti memenuhi kewajiban yang dipercayakan kepada kita dengan sebaik-baiknya. Allah menyebut bahwa perintah ini adalah pengajaran yang sangat baik dengan menegakkan amanat dan keadilan akan hidup dalam kedamaian.

Tafsir Ibnu Katsir dalam tafsirnya menjelaskan bahwa mengingatkan umat Islam tentang pentingnya memenuhi amanah dan berlaku adil. Amanah disini mencakup segala yang dipercayakan kepada seseorang mulai dari barang pribadi hingga posisi seperti

pengadilan dan kepemimpinan. Salah satu sifat terpuji yang harus dipegang teguh oleh setiap orang yang memiliki wewenang untuk mengadili adalah keadilan dalam pengadilan. Pentingnya nilai keadilan dalam amanat dalam kehidupan manusia bahwa mengikuti perintah ini akan membawa kebaikan.

#### Al Maidah Ayat 8

يَا أَيُّهَا الَّذِينَ آمَنُوا كُوْنُوا قَوَامِينَ لِلَّهِ شُهَدَاء بِالْقِسْطِ وَلَا يَجْرِمَنَّكُمْ شَنَآنُ قَوْمٍ عَلَى أَلَّا  
تَعْدِلُوا اعْدِلُوا هُوَ أَفْرَبُ لِلتَّقْوَى وَإِنَّ اللَّهَ إِنَّ اللَّهَ خَيْرٌ بِمَا تَعْمَلُونَ

**Artinya:** “Hai orang-orang yang beriman hendaklah kamu jadi orang-orang yang selalu menegakkan (kebenaran) karena Allah, menjadi saksi dengan adil. Dan janganlah sekali-kali kebencianmu terhadap suatu kaum, mendorong kamu untuk berlaku tidak adil. Berlaku adillah, karena adil itu lebih dekat kepada takwa. Dan bertakwalah kepada Allah, sesungguhnya Allah Maha Mengetahui apa yang kamu kerjakan”. (QS. Al-Maidah (5): 8)

Tafsir yang terdapat di surah Al Maidah yaitu Wahai orang-orang yang beriman kepada Allah dan Rasul-Nya Muhammad SAW jadilah kalian para penegak kebenaran demi mencari wajah Allah, sebagai saksi-saksi yang adil. Jangan sampai kebencian kalian terhadap suatu kaum membawa kalian untuk berbuat tidak adil. Berbuat adillah di antara para musuh dan para kawan kalian secara berimbang. Keadilan itu lebih dekat kepada rasa takut kepada Allah dan jangan bertindak zalim. Sesungguhnya Allah Maha Mengenal apa yang kalian lakukan dan Allah akan membalas kalian atasnya (Umar, 2016).

Penjelasan tafsir Al-Jalalain Allah memerintahkan orang-orang yang beriman untuk menjadi saksi yang berdiri teguh demi Allah SWT dan selalu mengutamakan kebenaran dan keadilan, meskipun itu bisa merugikan diri sendiri. Karena orang yang takut kepada Allah tidak akan membiarkan kebencianya terhadap orang lain mendorongnya untuk berlaku tidak adil, keberanian untuk menyatakan kebenaran dan keadilan adalah bukti takwa sejati karena merupakan jalan yang lebih dekat untuk mencapai ketakwaan. Allah Maha Mengetahui segala apa yang dilakukan manusia baik yang tampak maupun yang tersembunyi. Oleh karena itu umat Islam harus selalu adil dalam setiap tindakannya.

#### At-Taubah Ayat 119

يَا أَيُّهَا الَّذِينَ آمَنُوا اتَّقُوا اللَّهَ وَكُوْنُوا مَعَ الصَّادِقِينَ

**Artinya:** “Wahai orang-orang yang beriman, bertakwalah kepada Allah dan tetaplah bersama orang-orang yang benar”. (QS. At-Taubah (9): 119)

Tafsir Ibnu Katsir menjelaskan sebagai orang-orang yang beriman untuk tetap dalam ketakwaan kepada Allah sepanjang waktu dan menghindari segala jenis kebohongan dan kemunafikan. Ketika Allah memerintahkan untuk bersama orang-orang yang jujur. Kejujuran mencakup kejujuran dalam niat, ucapan dan tindakan menjadi orang jujur berarti memiliki integritas dalam sebuah aspek kehidupan dalam ucapan dan tindakan.

Tafsir Al-Muyassar menjelaskan bahwa ketakwaan kepada Allah adalah dasar dari semua Tindakan positif. Mengajak umat muslim untuk bergabung dengan orang-orang yang benar, orang-orang yang jujur adalah mereka yang jujur dalam iman, ucapan dan perbuatan. Kejujuran dalam iman berarti mempercayai Allah dan Rasul-Nya tanpa keraguan. Kejujuran dalam perbuatan berarti selalu berbuat baik dan menahan diri dari perbuatan jahat. Orang-orang yang jujur akan membuat seseorang menjadi lebih baik dan bertakwa.

Dalam membuat Aplikasi *KeyKeep* bagi karyawan, penting untuk memperhatikan kesesuaian dengan prinsip-prinsip Islam sebagaimana diajarkan dalam Islam dan hadis-hadis Rasulullah SAW. Salah satu hadis yang relevan adalah hadis yang diriwayatkan oleh Imam Muslim:

لَا إِيمَانَ لِمَنْ لَا أَمَانَةَ لَهُ، وَلَا دِينَ لِمَنْ لَا عَهْدَ لَهُ

**Artinya:** “Tidaklah beriman orang yang tidak bisa dipercaya dan tidaklah beragama orang yang tidak memegang janji”. (HR. Ahmad dan Al-Baihaqi)

Dalam hadis ini Rasulullah SAW pentingnya menjaga amanah ketika diberi tanggung jawab untuk menyimpan barang milik orang lain dan menjaga kepercayaan yang diberikan. Menjaga amanah adalah bukti dari keimanan seseorang karena itu setiap umat harus berusaha keras untuk menjadi orang yang amanah dan dapat dipercaya.

احْفَظْ عَلَى مَالِكَ فَإِنَّهُ مَتَاعٌ

**Artinya:** “Jagalah harta bendamu, karena sesungguhnya ia adalah bagian dari kehidupanmu”. (HR. Bukhari dan Muslim)

Dalam hadis ini Rasulullah SAW menekankan kewajiban seorang muslim untuk menjaga benda yang dimiliki. Setiap umat muslim bertanggung jawab untuk menjaga benda sendiri dan orang lain di sekitarnya agar tidak disalahgunakan atau dipergunakan untuk hal-hal yang dapat merugikan atau membahayakan orang lain (HR. Bukhari dan Muslim).

عَنْ عَبْدِ اللَّهِ بْنِ مَسْعُودٍ رَضِيَ اللَّهُ عَنْهُ قَالَ، قَالَ رَسُولُ اللَّهِ صَلَّى اللَّهُ عَلَيْهِ وَسَلَّمَ: «عَلَيْكُمْ بِالصِّدْقِ، فَإِنَّ الصِّدْقَ يَهْدِي إِلَى الْبِرِّ، وَالْبِرُّ يَهْدِي إِلَى الْجَنَّةِ، وَمَا يَرَأُ الرَّجُلُ يَصْدُقُ وَيَتَحَرَّى الصِّدْقَ حَتَّى يُكْتَبَ عِنْدَ اللَّهِ صِدِيقًا، وَإِيَّاكُمْ وَالْكَذَبُ، فَإِنَّ الْكَذَبَ يَهْدِي إِلَى الْفُجُورِ، وَالْفُجُورُ يَهْدِي إِلَى النَّارِ، وَمَا يَرَأُ الرَّجُلُ يَكْذِبُ وَيَتَحَرَّى الْكَذَبَ حَتَّى يُكْتَبَ عِنْدَ اللَّهِ كَذَابًا»

**Artinya:** “Kalian harus jujur karena sesungguhnya jujur itu menuntun kepada kebaikan dan kebaikan itu menuntun ke surga. Seseorang yang senantiasa jujur dan berusaha untuk jujur sehingga ditulis di sisi Allah sebagai orang yang jujur dan jauhilah oleh kalian dusta karena sesungguhnya dusta itu menuntun kepada keburukan”. (HR. Muslim, 2014)

Dalam hadis ini Rasulullah SAW menganjurkan untuk senantiasa berlaku jujur. Kejujuran bukan hanya sekedar tidak berbohong tetapi mencakup semua aspek kehidupan termasuk dalam perbuatan, ucapan dan niat. Kejujuran menuntun seseorang kepada kebaikan yaitu segala perbuatan dan kebijakan orang jujur akan selalu berusaha melakukan hal-hal yang benar. Rasulullah SAW menganjurkan untuk menjauhi dusta karena perilaku tersebut tidak hanya merugikan diri sendiri tetapi juga orang lain. Seseorang yang selalu jujur dan berusaha untuk tetap jujur akan dicatat sisi Allah sebagai orang yang sangat mulia dan kepercayaan yang tinggi di mata Allah.

إِنَّ اللَّهَ جَمِيلٌ يُحِبُّ الْجَمَالَ

**Artinya:** “Sesungguhnya Allah itu indah dan mencintai keindahan”. (HR. Muslim)

Dalam hadis ini Rasulullah SAW mengajarkan untuk menyimpan barang dengan rapi dan teratur tidak menciptakan lingkungan yang lebih menyenangkan secara visual tetapi juga mencerminkan penghormatan kita kepada Allah SWT yang mencintai keindahan kebersihan dan kerapian. Menyimpan barang yang rapi dapat menjaga kebersihan tempat penyimpanan. Dengan menyimpan barang dengan rapi dan indah, seorang muslim menerapkan nilai-nilai Islam dalam kehidupan sehari-hari dan menunjukkan tanggung jawab dalam menjaga barang sendiri. (HR. Muslim)

Dengan demikian, pengembangan aplikasi *KeyKeep* tidak hanya memenuhi kebutuhan pengguna tetapi juga sejalan dengan nilai-nilai Islam yang mendorong untuk umatnya belajar.

#### **1.1.4 Definisi, Singkatan, dan Akronim**

Berikut adalah definisi, singkatan dan akronim dari beberapa istilah yang telah disebutkan:

**Tabel 1. Akronim, Singkatan, Deskripsi Bab 1**

Akronim	Singkatan	Deskripsi
Ubuntu	-	Merupakan <i>software</i> yang bersifat <i>free</i> dan <i>open source</i> . Ubuntu merupakan salah satu distro dari Linux yang cukup terkenal karena kemudahan penggunaan serta aplikasi-aplikasi pendukungnya yang sangat lengkap dan sangat cocok untuk mengurangi masalah keamanan dan jaringan komputer (Fauzan, 2011).
Dashboard	-	Tampilan grafis yang menyajikan informasi secara visual, seperti data, metrik atau informasi lainnya.
HTTP	<i>Hypertext Transfer Protocol</i>	Sebuah protokol meminta atau menjawab antara client dan server (Zabar & Novianto, 2015).
SKPL	Spesifikasi Kebutuhan Perangkat Lunak	Merupakan spesifikasi dari perangkat lunak yang akan dikembangkan.
API	<i>Application Programming Interface</i>	Bahasa dan format pesan yang digunakan aplikasi atau program untuk berkomunikasi dengan <i>Operational System</i> (OS) atau program kontrol lainnya seperti manajemen database (Affrianto & Cahyono).
Desktop	-	Tampilan layar utama yang di dalamnya berisi ikon, jendela, <i>toolbar</i> dan <i>wallpaper</i> yang membantu pengguna untuk berinteraksi dengan seluruh sistem operasi dan dapat membantu pengguna untuk menjalankan aplikasi, mengakses file dan menyimpan data.

Akronim	Singkatan	Deskripsi
React JS	-	Menurut Sanchit Aggarwal, ReactJS adalah Pustaka JavaScript yang digunakan untuk mengembangkan komponen antarmuka pengguna ( <i>UI</i> ) yang dapat digunakan kembali.
Laravel	-	Menjadi salah satu <i>framework</i> yang dapat membantu developer untuk memaksimalkan penggunaan PHP di dalam proses pengembangan website (Aipina & Witriyono, 2022).
Web Service	-	Sebuah metode komunikasi yang memungkinkan dua atau lebih sistem komputer melalui jaringan seperti internet.
DB	Database	Database merupakan kumpulan data-data yang tersimpan secara rapi dan terstruktur di dalam suatu komputer dan sangat penting di dalam sistem informasi. Database sebuah sistem perangkat lunak yang menyediakan akses teknologi yang digunakan sangat banyak seperti MySQL, SQL (Chairane et al., 2023).
PHP	<i>Hypertext Preprocessor</i>	Suatu bahasa pemrograman berbasis kode-kode ( <i>script</i> ) yang digunakan untuk mengolah suatu data dan mengirimkannya kembali ke web menjadi kode HTML (Aipina & Witriyono, 2022).
PostgreSQL	-	Merupakan tempat untuk menyimpan dan mengelola data melalui perintah atau query SQL.
GUI	<i>Graphical User Interface</i>	Mekanisme interaksi utama antara perangkat dan pengguna untuk memilih berbagai jenis tindakan. (Utami, 2021).
HTTPS	<i>Hypertext Transfer Protocol Secure</i>	Memiliki kesamaan dengan <i>http</i> hanya saja <i>https</i> memiliki kelebihan fungsi di bidang keamanan ( <i>secure</i> ) (Zabar & Novianto, 2015).

Akronim	Singkatan	Deskripsi
RESTful API	<i>Representational State Transfer Application Programming Interface</i>	Tipe arsitektur dari <i>Application Programming Interface</i> disebut juga atau API. (Kaniya et al., 2022).
UX	<i>User Experience</i>	Persepsi sebuah produk, sistem atau jasa yang menilai seberapa kepuasan dan kenyamanan seseorang terhadap sebuah produk dan sistem. (Ferry, 2020).
FDD	<i>Feature Driven Development</i>	Merupakan pendekatan yang mengacu pada pembuatan sistem menggunakan metode yang mudah dimengerti dan diimplementasikan, teknik <i>problem solving</i> dan pelaporan yang mudah dimengerti dan dikontrol oleh <i>stakeholders</i> . (Palmer & John, 2001).
CSS	<i>Cascading Style Sheet</i>	Merupakan satuan atau gabungan kode yang tersinkron dalam HTML dan bertujuan untuk menggantikan gaya atau tampilan dari sebuah halaman website. (Afisina et al., 2020).

### 1.1.5 Deskripsi Umum Bab

SKPL dalam Aplikasi *KeyKeep* ini menjelaskan sistematika dari perangkat lunak sebagai berikut:

#### 1. Pendahuluan

Bagian ini menjelaskan permasalahan yang ingin diatasi oleh Aplikasi *KeyKeep*. Dalam hal ini mencakup mengenai kebutuhan pengguna, dan lingkungan aplikasi. Adapun informasi tambahan yang mungkin berkaitan dengan tujuan pengembangan aplikasi.

#### 2. Deskripsi umum perangkat lunak

Bagian ini menjelaskan konteks dan asal-usul dari Aplikasi *KeyKeep*. Hal ini mencakup dalam memilih teknologi yang digunakan, platform, dan arsitektur sistem.

### **3. Kebutuhan fungsional**

Bagian ini menjelaskan fungsional pada Aplikasi *KeyKeep*. Mencakup fitur-fitur yang dibutuhkan oleh pengguna dan ditentukan melalui wawancara dan analisis kebutuhan.

### **4. Fitur sistem**

Bagian ini menjelaskan cara mengatur kebutuhan fungsional ke dalam fitur-fitur aplikasi. Hal ini mencakup kebutuhan utama yang disediakan oleh Aplikasi *KeyKeep* untuk memenuhi kebutuhan pengguna.

### **5. Kebutuhan nonfungsional**

Bagian ini menjelaskan Kualitas Aplikasi *KeyKeep* dan batas minimum yang dapat diterima untuk setiap Kualitas. Hal ini mencakup keamanan, kinerja, dan penggunaan.

## **1.2 Deskripsi Umum Perangkat Lunak**

### **1.2.1 Perspektif Produk**

Aplikasi *KeyKeep* merupakan perangkat lunak yang dapat digunakan oleh karyawan PT. Multidaya Dinamika. Aplikasi *KeyKeep* dapat memudahkan karyawan dalam menyimpan barang-barang atau dokumen dalam lingkungan kerja yang menerapkan *open space*.

Diagram komponen utama dari sistem sebagai berikut:



**Gambar 3. Diagram *KeyKeep***

Berikut penjelasan singkat mengenai komponen-komponen dari diagram *KeyKeep*:

#### **1. User Interface**

Tampilan antarmuka yang digunakan oleh pengguna untuk berinteraksi dengan sistem *KeyKeep* dan dapat memudahkan pengguna dalam menggunakan Aplikasi *KeyKeep*.

## **2. Menitipkan Barang**

Fitur yang dapat digunakan pengguna untuk menyimpan barang di dalam loker. Pada fitur ini pengguna diminta untuk membuat kode PIN.

## **3. Mengambil Barang**

Fitur yang dapat digunakan pengguna untuk mengambil barang yang sebelumnya disimpan di loker. Pada fitur ini pengguna diminta memasukkan kode verifikasi yang tertera pada *e-receipt* dan kode PIN yang sudah dibuat sebelumnya.

## **4. Memilih Loker**

Pada fitur ini pengguna dapat memilih loker yang tersedia sesuai dengan kebutuhan mereka, seperti ukuran dan nomor loker.

## **5. Membuat Kode PIN sebanyak 4 digit**

Pengguna setelah memilih ukuran dan nomor loker diminta untuk membuat kode PIN sebanyak 4 digit angka.

## **6. Kode Verifikasi**

Pengguna harus memasukkan kode verifikasi yang tertera pada *e-receipt*. Yang didapatkan setelah melakukan titip barang.

### **1.2.2 Fungsi Produk**

Aplikasi *KeyKeep* dibuat untuk digunakan karyawan menyimpan barang-barang atau dokumen mereka dalam lingkungan kerja *open space*. Fungsi utama dari Aplikasi *KeyKeep* ini sebagai berikut:

1. Karyawan dapat menitipkan barang-barang mereka.
2. Karyawan dapat ambil barang-barang mereka yang sebelum dititipkan.
3. Karyawan dapat memilih nomor loker yang diinginkan.
4. Karyawan dapat memilih ukuran loker yang diinginkan.
5. Karyawan dapat membuat kode pin sendiri dengan 4digit angka.
6. Karyawan dapat membuka kembali loker sebanyak 3 kali kesempatan.
7. Karyawan dapat melihat detail loker yang digunakan.

### **1.2.3 Kelas dan Karakteristik Pengguna**

Pengguna Aplikasi *KeyKeep* terbagi menjadi dua karakteristik yaitu pengguna dan pengelola data.

#### **1. Pengguna**

Hak akses: Menitipkan barang, mengambil barang, memilih ukuran loker, memilih nomor loker, membuat kode pin, mendapatkan detail loker yang digunakan, menginput kode verifikasi.

#### **2. Pengelola Data**

Hak akses: Menambah, mengubah ukuran, jumlah, warna, dan aset pada aplikasi *KeyKeep*.

### **1.2.4 Lingkungan Operasi**

Aplikasi *KeyKeep* ini akan digunakan oleh karyawan PT. Multidaya Dinamika. Aplikasi *KeyKeep* dapat berjalan pada platform Ubuntu dengan spesifikasi minimal menggunakan 20.04.6.

### **1.2.5 Batasan Perancangan dan Implementasi**

Aplikasi *KeyKeep* dapat digunakan setiap saat. Aplikasi *KeyKeep* ini hanya digunakan sebagai alat untuk titip barang dan ambil barang. Terdapat beberapa batasan antara lain:

1. Aplikasi *KeyKeep* hanya dapat digunakan pada platform Ubuntu dan harus menggunakan internet serta komputer yang sudah dikonfigurasi menjadi kiosk.
2. Aplikasi *KeyKeep* dikembangkan dengan fokus pada proses pengguna menitip barang dan mengambil barang yang telah disimpan.

### **1.2.6 Dokumentasi Pengguna**

Dokumentasi pengguna aplikasi *KeyKeep* yang dapat dijabarkan sebagai berikut:

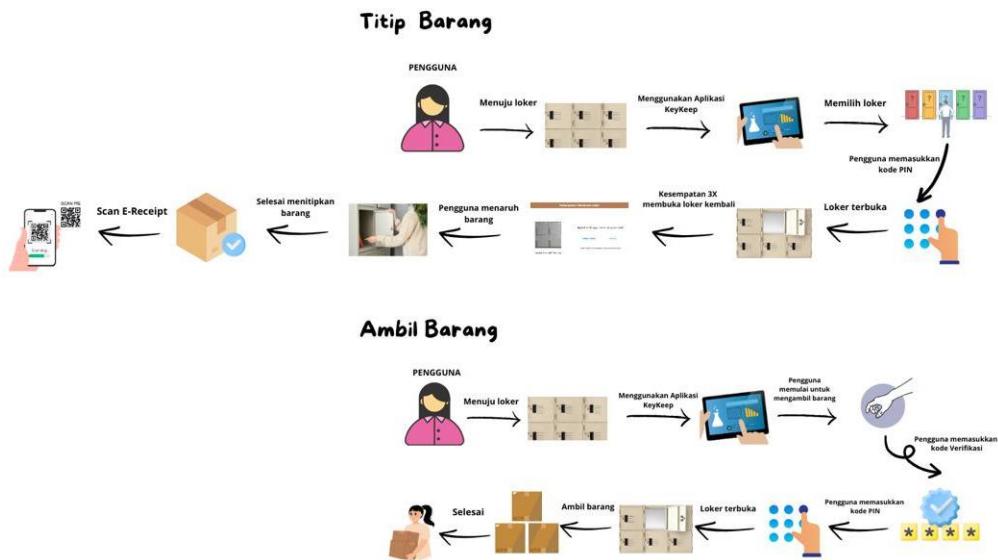
#### **1. Titip Barang**

- Status ketersediaan loker, sebelum pengguna menyimpan barang-barang mereka bisa memeriksa terlebih dahulu loker mana yang masih tersedia atau kosong.

- Nomor loker dan ukuran loker, setelah menemukan loker yang masih tersedia pengguna bisa memilih nomor dan ukuran loker yang diinginkan.
- Membuat kode pin, setelah memilih nomor dan ukuran loker pengguna bisa membuat kode pin dengan 4digit angka.
- Kesempatan membuka kembali loker, jika pengguna ada barang yang tertinggal yang belum dimasukkan, pengguna diberikan kesempatan membuka loker sebanyak 3 kali.
- E-receipt, jika barang sudah ditaruh di dalam loker dan loker sudah tertutup maka pengguna akan mendapatkan kode qr

## 2. Ambil Barang

- Kode verifikasi, untuk mengambil barang pengguna bisa memasukkan kode verifikasi yang tertera di e-receipt.
- Memasukkan kode PIN, setelah pengguna memasukkan kode verifikasi pengguna memasukkan kode PIN yang sudah dibuat sebelumnya.



Gambar 4. Rich Picture

### 1.2.7 Asumsi dan Dependensi

Faktor asumsi dan dependensi pada aplikasi *KeyKeep*, yaitu:

#### 1. Asumsi

- Aplikasi *KeyKeep* dapat digunakan oleh pengguna secara mandiri tanpa bantuan teknis yang signifikan.
- Pengembangan aplikasi akan memerlukan integrasi perangkat keras (*hardware*) dengan sistem perangkat lunak.
- Aplikasi *KeyKeep* dapat diimplementasikan di berbagai lokasi publik atau tempat keramaian

## 2. Depensi

- Aplikasi *KeyKeep* menggunakan teknologi berbasis web untuk antarmuka pengguna (*frontend*) dengan *framework* seperti React.js.
- Aplikasi *KeyKeep* menggunakan teknologi PHP dengan *framework* Laravel sebagai *backend*.
- Integrasi perangkat keras seperti mekanisme pengunci dan layar sentuh menjadi bagian penting dari pengembangan aplikasi ini.
- Aplikasi *KeyKeep* menggunakan PostgreSQL sebagai Database yang digunakan untuk menyimpan data informasi Kotak Penyimpanan Pintar.

### 1.3 Kebutuhan Fungsional

#### 1.3.1 Antarmuka Pengguna

Pengguna berinteraksi dengan aplikasi *KeyKeep* melalui antarmuka grafis atau *Graphical User Interface (GUI)*. Aplikasi *KeyKeep* menampilkan menu utama yang ada pada kotak penyimpanan pintar seperti menitipkan barang dan mengambil barang.

#### 1.3.2 Antarmuka Perangkat Keras

Kebutuhan minimum perangkat keras yang digunakan pemakai untuk berinteraksi dengan aplikasi *KeyKeep* adalah:

1. CPU: Intel® Core™ i5-4278U CPU @ 2.60GHz × 4
2. RAM: 7 GB
3. Grafik: Intel® Iris(R) Graphics 5100 (HSW GT3)
4. OS: Ubuntu 20.04
5. Disk Capacity: 64,0 GB



**Gambar 5. Desain Kotak Penyimpanan Pintar Tampak Luar**



**Gambar 6. Penampakan Kotak Penyimpanan Pintar Tampak Dalam**

### **1.3.3 Antarmuka Perangkat Lunak**

Aplikasi *KeyKeep* dirancang sebagai solusi terintegrasi yang berintegrasi dengan perangkat keras seperti mekanisme pengunci dan layar sentuh untuk menyediakan layanan penyimpanan digital dan efisien. Aplikasi dapat beroperasi pada sistem operasi Ubuntu 20.04, aplikasi ini menggunakan *framework* React.js untuk *frontend*, menggunakan *framework* Tailwind CSS untuk memastikan responsivitas dan kualitas antarmuka pengguna serta dapat memberikan keleluasaan dalam pengembangan. aplikasi

ini melakukan komunikasi dengan server melalui protokol API, dengan implementasi RESTful API yang memfasilitasi pertukaran data yang efisien antara *frontend* dan *backend*. Layanan yang dibutuhkan, seperti memilih loker, menginput kode PIN dan kode verifikasi dilakukan melalui komunikasi HTTP. Aplikasi *KeyKeep* dapat terintegrasi dengan layanan eksternal, memungkinkan aplikasi dengan layanan keuangan, memastikan transaksi keuangan dilakukan secara aman melalui komunikasi HTTPS. *Dashboard* aplikasi berfungsi sebagai alat pemantauan untuk memonitor data loker, termasuk ketersediaan, status penggunaan dan transaksi yang terjadi. Komunikasi data dan permintaan pengguna seperti pencarian barang atau pengelolaan akun dilakukan dengan standar keamanan HTTPS, menjaga kerahasiaan informasi. Meskipun bergantung pada koneksi internet, antarmuka aplikasi *KeyKeep* memberikan pengalaman pengguna yang efisien dan memudahkan akses terhadap informasi serta transaksi yang dilakukan.

#### **1.3.4 Antarmuka Komunikasi**

Aplikasi *KeyKeep* menggunakan standar komunikasi HTTP/HTTPS saat mengirim dan mengambil data dari RESTful API. Penggunaan HTTPS memastikan keamanan data selama proses pertukaran informasi antara *frontend* dan *backend*. Mekanisme sinkronisasi diaktifkan saat mengirim dan mengambil data dari API untuk memperbarui informasi seperti *list* loker, ukuran loker dan detail loker. Upaya ini memungkinkan aplikasi berfungsi efisien, mengoptimalkan komunikasi dengan *server* hanya saat dibutuhkan, memastikan akses pengguna ke informasi terbaru secara aktual.

### **1.4 Fitur Sistem**

#### **1.4.1 Fitur Home pada Aplikasi**

##### **1.4.1.1 Deskripsi**

Fitur *Home* pada aplikasi *KeyKeep* merupakan fitur awal yang akan tampil ketika pengguna ingin menggunakan aplikasi. Fitur ini memberikan akses ke fitur titip barang dan ambil barang.

##### **1.4.1.2 Trigger**

Fitur *Home* diaktifkan secara otomatis ketika device menyala atau menekan “klik untuk mulai”.

#### **1.4.1.3 *Input***

Tidak ada input yang diperlukan dari pengguna untuk melihat tampilan *Home*.

#### **1.4.1.4 *Output***

Keluaran dari fitur ini adalah tampilan *Home* yang mencakup waktu, video dan pilih bahasa.

#### **1.4.1.5 Skenario Utama**

Prakondisi: Aplikasi pada bagian *Home* selalu aktif.

Pascakondisi: Pengguna melihat tampilan berupa video, waktu, dan pilih bahasa.

Langkah-langkah:

1. Pengguna melihat tampilan awal pada fitur *Home*.
2. Aplikasi menampilkan tampilan utama yang berupa video dengan informasi waktu, pilih bahasa dan klik untuk memulai.

#### **1.4.1.6 Skenario Eksepsional 1**

Prakondisi: Pengguna ingin melakukan penitipan barang, terjadi gangguan internet ketika ingin melakukan penitipan barang.

Pascakondisi: Aplikasi *KeyKeep* tidak dapat digunakan secara keseluruhan

Langkah-langkah:

1. Pengguna melakukan penitipan barang.
2. Terjadi gangguan internet pada saat melakukan penitipan barang.
3. Aplikasi *KeyKeep* tidak dapat digunakan secara keseluruhan.

### **1.4.2 Fitur Titip Barang pada Aplikasi**

#### **1.4.2.1 Deskripsi**

Fitur Titip Barang pada Aplikasi *KeyKeep* dibuat untuk membantu karyawan dalam menyimpan barang-barang ke dalam loker dengan aman.

#### **1.4.2.2 Trigger**

Fitur Titip Barang akan diaktifkan ketika pengguna memilih opsi “Titip Barang” di antarmuka pengguna halaman menu pada aplikasi *KeyKeep*

### **1.4.2.3 Input**

Tidak ada masukkan yang perlu diisi oleh pengguna untuk menampilkan fitur Titip Barang.

### **1.4.2.4 Output**

Keluaran dari fitur ini adalah pengguna dapat memilih ukuran dan nomor loker yang sesuai dengan kebutuhan pengguna.

### **1.4.2.5 Skenario Utama**

Prakondisi: Pengguna memilih opsi “Titip Barang” yang terdapat di layar aplikasi.

Pascakondisi: Setelah memilih opsi “Titip Barang” pengguna dapat memilih ukuran dan nomor loker pada aplikasi *KeyKeep*.

Langkah-langkah:

1. Pengguna memilih opsi “Titip Barang”
2. Pengguna memilih ukuran dan nomor loker sesuai dengan kebutuhan
3. Pengguna dapat memilih salah satu loker untuk menitipkan barang-barang.

## **1.4.3 Fitur Ambil Barang pada Aplikasi**

### **1.4.3.1 Deskripsi**

Fitur Ambil Barang pada aplikasi *KeyKeep* digunakan ketika pengguna ingin mengambil barang yang telah disimpan di dalam loker.

### **1.4.3.2 Trigger**

Fitur Ambil Barang akan diaktifkan ketika pengguna memilih opsi “Ambil Barang” di antarmuka pengguna halaman menu pada aplikasi *KeyKeep*.

### **1.4.3.3 Input**

Tidak ada input yang perlu diisi oleh pengguna untuk menampilkan fitur Ambil Barang.

### **1.4.3.4 Output**

Keluaran dari fitur ini adalah halaman untuk memasukkan kode verifikasi yang telah diperoleh sebelumnya yang tercantum di *e-Receipt*.

### **1.4.3.5 Skenario Utama**

Prakondisi: Pengguna ingin mengambil barang-barang di dalam loker dengan menginput kode verifikasi yang tertera di *e-Receipt*

Pascakondisi: Pengguna menginput kode verifikasi dan kode PIN dengan benar maka loker akan terbuka.

Langkah-langkah:

1. Pengguna ingin mengambil barang-barang di loker.
2. Pengguna menginput kode verifikasi dan kode PIN
3. Loker terbuka.

#### **1.4.3.6 Skenario Eksepsional 1**

Prakondisi: Pengguna ingin mengambil barang di dalam loker dan salah menginput kode verifikasi

Pascakondisi: Pengguna tidak dapat mengambil barang karena kode verifikasi yang di input tidak sesuai dengan yang tertera pada *e-Receipt*.

Langkah-langkah:

1. Pengguna ingin mengambil barang.
2. Pengguna salah menginput kode verifikasi.
3. Loker tidak terbuka karena kode verifikasi tidak sesuai.

#### **1.4.4 Fitur Pilih Loker pada Aplikasi**

##### **1.4.4.1 Deskripsi**

Fitur Pilih Loker pada aplikasi *KeyKeep* diberikan untuk pengguna untuk memilih loker dengan ukuran dan nomor loker yang sesuai dengan kebutuhan penyimpanan pengguna.

##### **1.4.4.2 Trigger**

Fitur ini diaktifkan ketika pengguna memilih opsi “Titip Barang” di antarmuka pengguna halaman menu pada aplikasi *KeyKeep*, lalu pengguna dapat memilih ukuran maupun nomor yang tersedia.

##### **1.4.4.3 Input**

Data yang di dapat untuk fitur ini sesuai dengan pilihan pengguna.

##### **1.4.4.4 Output**

Keluaran dari fitur ini adalah konfirmasi tentang pemilihan loker yang dipilih, lalu menyimpan data terkait detail loker yang dipilih.

#### **1.4.4.5 Skenario Utama**

Prakondisi: Pengguna ingin menitipkan barang di dalam loker.

Pascakondisi: Pengguna bisa memilih ukuran dan nomor loker yang diinginkan sesuai dengan kebutuhan penyimpanan.

Langkah-langkah:

1. Pengguna ingin menitipkan barang di dalam loker.
2. Pengguna memilih ukuran dan nomor loker.
3. Pengguna bisa mengetahui status ketersediaan loker.

#### **1.4.5 Fitur *Input PIN* pada Aplikasi**

##### **1.4.5.1 Deskripsi**

Fitur Input PIN pada aplikasi *KeyKeep* digunakan pengguna setelah menitipkan barang dengan menginput kode sebanyak 4digit angka.

##### **1.4.5.2 Trigger**

Fitur ini akan diaktifkan pada saat, setelah memilih loker dan juga dibutuhkan pada saat pengambilan loker sebagai validasi.

##### **1.4.5.3 Input**

Data yang diperlukan untuk fitur ini adalah menginput 4digit angka dari pengguna.

##### **1.4.5.4 Output**

Keluaran dari fitur ini ketika setelah memilih loker, input akan disimpan yang akan digunakan pada saat pengambilan barang.

#### **1.4.5.5 Skenario Utama**

Prakondisi: Pengguna memilih opsi “Titip Barang”

Pascakondisi: Pengguna diminta untuk membuat kode PIN dengan 4digit angka.

Langkah-langkah:

1. Pengguna ingin menitipkan barang.
2. Pengguna diminta untuk membuat kode PIN dengan 4digit angka.
3. Setelah pengguna membuat kode PIN, loker akan terbuka.

#### **1.4.5.6 Skenario Eksepsional 1**

Prakondisi: Pengguna tidak memasukkan kode PIN sebanyak 4 digit.

Pascakondisi: PIN tidak bisa disimpan.

Langkah-langkah:

1. Pengguna memasukkan kode PIN kurang dari 4digit angka.
2. Aplikasi mendeteksi bahwa kode PIN tidak lengkap atau kurang dari 4digit angka.
3. Kode PIN tidak bisa disimpan atau tombol simpan tidak aktif.

#### **1.4.6 Fitur Fitur Kesempatan Membuka Kembali pada Aplikasi**

##### **1.4.6.1 Deskripsi**

Fitur Kesempatan Membuka Kembali memungkinkan pengguna untuk membuka kembali loker. Pengguna memiliki 3 kali kesempatan, jika sudah memakai kesempatan sebanyak 3 kali, maka pengguna tidak dapat membuka loker lagi kecuali dengan memilih fitur “ambil barang”.

##### **1.4.6.2 Trigger**

Fitur ini dapat dipicu oleh keadaan loker, ketika pengguna telah selesai memasukkan barang dan pintu loker sudah tertutup dengan benar.

##### **1.4.6.3 Input**

Fitur ini dapat diaktifkan ketika pengguna menekan tombol “Buka Kembali” dengan kesempatan membuka sebanyak 3 kali.

##### **1.4.6.4 Output**

Keluaran dari fitur ini yaitu loker dapat terbuka kembali.

##### **1.4.6.5 Skenario Utama**

Prakondisi: Pengguna menekan “Buka Kembali”

Pascakondisi: Pintu loker terbuka kembali.

Langkah-langkah:

1. Pengguna menekan tombol “Buka Kembali”.
2. Pintu loker terbuka kembali.
3. Pengguna dapat meletakan barang yang lupa disimpan.

#### **1.4.6.6 Skenario Eksepsional 1**

Prakondisi: Pengguna menekan tombol “buka kembali” dengan kondisi loker tidak tertutup dengan benar.

Pascakondisi: Aplikasi menampilkan pesan bahwa loker belum tertutup dengan benar.

Langkah-langkah:

1. Loker belum tertutup dengan benar.
2. Pengguna menekan tombol “Buka Kembali”
3. Aplikasi menampilkan pesan bahwa loker belum tertutup dengan benar.

#### **1.4.6.7 Skenario Eksepsional 2**

Prakondisi: Pengguna menekan tombol “selesai” dengan kondisi loker tidak tertutup dengan benar.

Pasckondisi: Aplikasi menampilkan pesan bahwa loker belum tertutup dengan benar.

Langkah-langkah:

1. Loker belum tertutup dengan benar.
2. Pengguna menekan tombol “selesai”.
3. Aplikasi menampilkan pesan bahwa loker belum tertutup dengan benar.

#### **1.4.7 Fitur Detail Loker pada Aplikasi**

##### **1.4.7.1 Deskripsi**

Fitur Detail Loker memungkinkan pengguna akan mendapatkan detail loker yang digunakan, yang tertera pada *e-Receipt*.

##### **1.4.7.2 Trigger**

Pengguna diminta untuk menekan tombol “selesai” dengan kondisi loker tertutup dengan benar.

##### **1.4.7.3 Input**

Pengguna diminta untuk memindai kode qr yang telah tertampil di layar dan membuka link yang tertera pada hasil pindaian.

#### **1.4.7.4 Output**

Pengguna mendapatkan detail loker seperti ukuran, nomor loker, waktu transaksi dan kode verifikasi.

#### **1.4.7.5 Skenario Utama**

Prakondisi: Pengguna melakukan pemindaian pada kode qr yang tertampil di layar setelah melakukan titip barang.

Pascakondisi: Pengguna membuka tautan hasil dari pemindaian kode qr dan mendapatkan detail loker berupa *e-Receipt*.

Langkah-langkah:

1. Pengguna melakukan pemindaian kode qr yang tertampil pada layar.
2. Pengguna membuka tautan yang didapat dari hasil pemindaian kode qr.
3. Pengguna mendapatkan detail loker seperti ukuran, nomor loker, waktu transaksi dan kode verifikasi berupa *e-Receipt*.

#### **1.4.7.6 Skenario Eksepsional 1**

Prakondisi: Sebelum mendapatkan *e-Receipt* internet mati.

Pascakondisi: *e-Receipt* tidak tampil di layar pengguna.

Langkah-langkah:

1. Pengguna melakukan penitipan barang.
2. Sebelum pengguna mendapatkan *e-Receipt* internet mati.
3. *e-Receipt* tidak tampil di layer pengguna.

#### **1.4.8 Fitur Validasi Kode Verifikasi pada Aplikasi**

##### **1.4.8.1 Deskripsi**

Fitur Validasi Kode Verifikasi dibuat untuk mekanisme keamanan yang dibuat untuk memastikan hanya pengguna yang dapat mengakses dan mengambil barang.

##### **1.4.8.2 Trigger**

Fitur ini diaktifkan ketika pengguna memilih opsi “Ambil Barang”.

#### **1.4.8.3 Input**

Data yang dibutuhkan dalam fitur ini adalah kode verifikasi yang tercantum di *e-Receipt* dan kode PIN yang telah dibuat oleh pengguna pada saat menitip barang.

#### **1.4.8.4 Output**

Keluaran dari fitur ini adalah apabila pengguna memasukkan kode verifikasi dengan benar, maka pengguna akan diminta untuk memasukkan kode PIN.

#### **1.4.8.5 Skenario Utama**

Prakondisi: Pengguna memilih opsi “Ambil Barang”.

Pascakondisi: Pengguna menginput kode verifikasi yang tertera di *e-Receipt*

Langkah-langkah:

1. Pengguna memilih opsi “Ambil Barang”.
2. Pengguna menginput kode verifikasi yang tertera di *e-Receipt*.

#### **1.4.8.6 Skenario Eksepsional 1**

Prakondisi: Pengguna tidak memasukkan kode verifikasi yang terdiri dari 6 digit.

Pascakondisi: Pengguna tidak dapat melanjutkan ke halaman berikutnya.

Langkah-langkah:

1. Pengguna melakukan validasi kode verifikasi.
2. Pengguna tidak memasukkan kode verifikasi yang terdiri dari 6 digit.
3. Pengguna tidak dapat melanjutkan ke halaman berikutnya.

#### **1.4.8.7 Skenario Eksepsional 2**

Prakondisi: Pengguna salah memasukkan kode verifikasi

Pascakondisi: Pengguna tidak dapat melanjutkan ke halaman berikutnya

Langkah-langkah:

1. Pengguna melakukan validasi kode verifikasi yang tidak terdaftar.
2. Server mendeteksi bahwa kode verifikasi tidak sesuai dan menampilkan pesan kesalahan.
3. Pengguna tidak dapat melanjutkan ke halaman berikutnya.

#### **1.4.9 Fitur Validasi Kode PIN pada Aplikasi**

##### **1.4.9.1 Deskripsi**

Fitur Validasi Kode PIN dilakukan setelah pengguna berhasil memasukkan validasi kode verifikasi. Fitur ini dibuat untuk Meningkatkan keamanan dala melakukan pengambilan barang.

##### **1.4.9.2 Trigger**

Fitur ini diaktifkan setelah pengguna memasukkan kode verifikasi dengan benar.

##### **1.4.9.3 Input**

Masukkan yang dibutuhkan fitur ini yaitu kode PIN yang sudah dibuat oleh pengguna.

##### **1.4.9.4 Output**

Loker terbuka dan pengguna dapat mengambil barang yang dititipkan sebelumnya.

##### **1.4.9.5 Skenario Utama**

Prakondisi: Pengguna telah berhasil memasukkan kode verifikasi dan kode PIN yang sebelumnya dibuat dengan benar.

Pascakondisi: Loker terbuka dan pengguna dapat mengambil barang yang dititipkan sebelumnya.

Langkah-langkah:

1. Pengguna telah berhasil memasukkan kode verifikasi.
2. Pengguna telah memasukkan kode PIN sebanyak 4digit angka yang sebelumnya dibuat dengan benar.
3. Loker berhasil terbuka.

##### **1.4.9.6 Skenario Eksepsional 1**

Prakondisi: Pengguna memasukkan validasi kode PIN kurang dari 4 digit.

Pascakondisi: Loker tidak terbuka.

Langkah-langkah:

1. Pengguna melakukan validasi kode PIN.
2. Pengguna memasukkan validasi kode PIN kurang dari 4 digit.
3. Loker tidak terbuka.

#### **1.4.9.7 Skenario Eksepsional 2**

Prakondisi: Pengguna salah memasukkan kode PIN.

Pascakondisi: Kode PIN tidak bisa melanjutkan kode halaman berikutnya.

Langkah-langkah:

1. Pengguna memasukkan validasi kode PIN yang tidak terdaftar.
2. Server mendeteksi bahwa kode PIN tidak sesuai dan menampilkan pesan kesalahan.
3. Pengguna tidak bisa melanjutkan ke halaman berikutnya.

#### **1.4.10 Fitur Modul Loker pada Web Service**

##### **1.4.10.1 Deskripsi**

Fitur Modul Loker admin dapat mengelola loker seperti menambah, mengurangi, mengatur status ketersediaan dan mengkategorikan loker sesuai dengan ukuran yang tersedia.

##### **1.4.10.2 Trigger**

Fitur ini diaktifkan pada saat admin masuk ke dalam Modul Loker pada *web service*.

##### **1.4.10.3 Input**

Masukkan yang dibutuhkan dalam fitur ini adalah menambah, mengurangi, mengatur status ketersediaan dan mengkategorikan loker sesuai dengan ukuran yang tersedia.

##### **1.4.10.4 Output**

Keluaran dari fitur ini adalah pengkonfirmasi yang telah diubah sesuai dengan kebutuhan.

##### **1.4.10.5 Skenario Utama**

Prakondisi: Admin masuk ke dalam Modul Loker dan memiliki akses untuk menambah, mengurangi, mengatur status ketersediaan dan mengkategorikan loker.

Pascakondisi: Admin berhasil mengelola loker dengan menambah, mengurangi, mengatur ketersediaan dan mengkategorikan loker sesuai dengan kebutuhan.

Langkah-langkah:

1. Admin login ke dalam *web service* dengan menggunakan akun admin.

2. Admin masuk ke dalam modul terminal.
3. Admin memilih terminal mana yang ingin diubah.
4. Admin masuk ke dalam Modul Loker.
5. Admin memilih opsi untuk menambah loker.
6. Admin memilih opsi untuk mengurangi loker.
7. Admin memilih opsi untuk mengatur ketersediaan.
8. Admin memilih opsi untuk mengkategorikan loker.
9. Admin berhasil menyimpan perubahan dari opsi yang telah diubah.

#### **1.4.10.6 Skenario Eksepsional**

Prakondisi: Admin menambah loker dengan id loker yang sama dengan yang sudah ada.

Pascakondisi: Admin tidak berhasil menambah loker, *web service* menampilkan pesan “*The locker id has already been taken*”.

Langkah-langkah:

1. Admin login ke dalam *web service* dengan menggunakan akun admin.
2. Admin masuk ke dalam modul terminal.
3. Admin memilih terminal mana yang ingin diubah.
4. Admin masuk ke dalam modul loker.
5. Admin memilih opsi untuk menambah loker.
6. Admin menggunakan id loker yang sama dengan yang sudah ada.
7. *Web service* menampilkan pesan “*The locker id has already been taken*”.
8. Admin tidak berhasil menambahkan loker.

#### **1.4.11 Fitur Modul Loker Size pada Web Service**

##### **1.4.11.1 Deskripsi**

Fitur Modul Loker Size admin dapat mengelola informasi mengenai ukuran dan deskripsi loker seperti menambah, mengurangi, mengubah ukuran loker, mengubah informasi dimensi loker dan mengubah informasi deskripsi loker.

##### **1.4.11.2 Trigger**

Fitur ini diaktifkan pada saat admin masuk ke dalam Modul Loker *Size*.

#### **1.4.11.3 Input**

Masukkan yang dibutuhkan dalam fitur ini adalah menambah, mengurang, mengubah informasi deskripsi loker.

#### **1.4.11.4 Output**

Keluaran dari fitur ini adalah mengkonfirmasi yang telah diubah sesuai dengan kebutuhan.

#### **1.4.11.5 Skenario Utama**

Prakondisi: Admin masuk ke dalam Modul Loker *Size* dan memiliki akses untuk menambah, mengurangi, mengubah ukuran loker, mengubah informasi dimensi loker dan mengubah informasi deskripsi loker.

Pascakondisi: Admin berhasil mengelola loker *size* dengan menambah, mengurangi, mengubah ukuran loker, mengubah informasi dimensi loker dan mengubah informasi deskripsi loker.

Langkah-langkah:

1. Admin login ke dalam *web service* dengan menggunakan akun admin.
2. Admin masuk ke dalam modul terminal.
3. Admin memilih terminal mana yang ingin diubah.
4. Admin masuk ke dalam Modul Loker *Size*
5. Admin memilih opsi untuk menambah ukuran loker.
6. Admin memilih opsi untuk mengurangi ukuran loker.
7. Admin memilih opsi untuk mengubah ukuran loker.
8. Admin memilih opsi untuk mengubah informasi dimensi loker.
9. Admin memilih opsi untuk mengubah informasi deskripsi loker.
10. Admin berhasil menyimpan perubahan dari opsi yang telah diubah.

#### **1.4.11.6 Skenario Eksepsional**

Prakondisi: Admin menambah ukuran loker dengan id loker *size* yang sama dengan yang sudah ada.

Pascakondisi: Admin tidak berhasil menambah loker *size*, *web service* menampilkan pesan “*The size locker id has already been taken*”.

Langkah-langkah:

1. Admin login ke dalam *web service* dengan menggunakan akun admin.
2. Admin masuk ke dalam halaman Terminal.
3. Admin memilih terminal mana yang ingin diubah.
4. Admin masuk ke dalam Modul Loker *Size*.
5. Admin memilih opsi untuk menambah ukuran loker.
6. Admin menggunakan id loker *size* yang sama dengan yang sudah ada.
7. *Web service* menampilkan pesan “*The size locker id has already been taken*”.
8. Admin tidak berhasil menambahkan ukuran loker.

#### **1.4.12 Fitur Dashboard pada Web Service**

##### **1.4.12.1 Deskripsi**

Fitur *Dashboard* admin ini untuk melihat informasi status loker seperti loker yang tersedia, tidak tersedia, loker yang sedang dalam perbaikan, grace period (pengguna yang meninggalkan barangnya tanpa sengaja) dan jumlah transaksi.

##### **1.4.12.2 Trigger**

Fitur ini diaktifkan pada saat admin masuk ke dalam *dashboard* yang ada pada *web service*.

##### **1.4.12.3 Input**

Masukkan dari fitur ini, diperlukan login ke dalam *web service* sebagai admin.

##### **1.4.12.4 Output**

Keluaran dari fitur ini adalah tampilan *dashboard* yang memiliki tampilan untuk melihat loker yang tersedia, tidak tersedia, sedang dalam perbaikan, grace period (pengguna yang meninggalkan barangnya tanpa sengaja) dan jumlah transaksi.

##### **1.4.12.5 Skenario Utama**

Prakondisi: Admin telah masuk ke halaman *dashboard*, data terkait status loker sudah tersedia dalam sistem.

Pascakondisi: Admin dapat melihat informasi terkait status loker dengan jelas dan akurat melalui *dashboard* admin.

Langkah-langkah:

1. Admin login ke dalam *web service* dengan menggunakan akun admin.
2. *Dashboard* admin menampilkan informasi mengenai status loker.

#### **1.4.12.6 Skenario Eksepsional**

Prakondisi: Admin login tidak menggunakan akun admin saat ingin mengakses.

Pascakondisi: Admin tidak dapat mengakses *web service* dan *dashboard*.

Langkah-langkah:

1. Admin mengakses *Web Service* dan *Dashboard* tanpa login menggunakan akun admin.
2. Sistem mendeteksi bahwa akun tidak memiliki hak akses sebagai admin.
3. *Web Service* dan *Dashboard* tidak dapat diakses.

#### **1.4.13 Fitur Transaksi pada *Web Service***

##### **1.4.13.1 Deskripsi**

Fitur Transaksi pada admin dapat melihat detail setiap transaksi loker, termasuk informasi mengenai kode verifikasi dan kode PIN yang digunakan.

##### **1.4.13.2 Trigger**

Fitur ini diaktifkan pada saat admin masuk ke modul transaksi pada sistem.

##### **1.4.13.3 Input**

Tidak ada masukkan yang perlu diisi oleh admin untuk menampilkan modul transaksi.

##### **1.4.13.4 Output**

Keluaran dari fitur ini adalah admin dapat melihat tampilan transaksi penggunaan loker termasuk kode verifikasi dan kode PIN.

##### **1.4.13.5 Skenario Utama**

Prakondisi: Admin masuk ke modul transaksi

Pascakondisi: Admin dapat melihat transaksi yang terjadi pada loker dan dapat mengetahui detail setiap transaksi penggunaan loker seperti kode verifikasi dan kode PIN.

Langkah-langkah:

1. Admin masuk ke dalam modul transaksi.
2. Admin dapat melihat transaksi yang terjadi pada loker dan mengetahui detail setiap transaksi penggunaan loker.

#### **1.4.14 Fitur Terminal pada *Web Service***

##### **1.4.14.1 Deskripsi**

Fitur Terminal ini, admin dapat melihat terminal mana yang sedang aktif dalam sistem. Fitur ini membantu admin dalam mengelola penggunaan terminal.

##### **1.4.14.2 Trigger**

Fitur ini diaktifkan pada saat admin masuk ke halaman terminal pada sistem.

##### **1.4.14.3 Input**

Tidak ada masukkan yang perlu diisi oleh admin untuk menampilkan halaman terminal.

##### **1.4.14.4 Output**

Keluaran dari fitur ini adalah menampilkan terminal mana yang sedang aktif dalam sistem.

#### **1.4.14.5 Skenario Utama**

Prakondisi: Admin telah masuk ke halaman terminal.

Pascakondisi: Admin dapat melihat dan mengetahui terminal mana yang sedang aktif dalam sistem melalui halaman terminal.

Langkah-langkah:

1. Admi telah masuk ke halaman terminal
2. Admin dapat melihat dan mengetahui terminal mana yang sedang aktif dalam sistem melalui halaman terminal.

#### **1.4.15 Fitur Modul Config pada *Web Service***

##### **1.4.15.1 Deskripsi**

Fitur Modul Config di *dashboard* aplikasi *KeyKeep* digunakan admin untuk mengelola dan mengkonfigurasikan elemen-elemen tampilan aplikasi termasuk warna dasar, logo, video pada tampilan awal, serta serangkaian ikon yang berkaitan dengan operasional loker seperti titip barang, ambil barang, loker terbuka, loker tertutup dan konfirmasi selesai.

##### **1.4.15.2 Trigger**

Admin memutuskan untuk memperbarui tampilan aplikasi.

### **1.4.15.3 Input**

Data yang digunakan untuk fitur ini adalah:

1. Pilihan warna primary baru.
2. File logo baru.
3. File video baru untuk background pada tampilan awal.
4. File-file ikon baru.

### **1.4.15.4 Output**

Tampilan diperbarui sesuai dengan konfigurasi yang telah diubah oleh admin.

### **1.4.15.5 Skenario Utama**

Prakondisi: Admin sudah masuk ke *dashboard* aplikasi *KeyKeep* dan memilih opsi “Modul Config”.

Pascakondisi: Aplikasi menampilkan tampilan yang telah diperbarui sesuai dengan pengaturan admin.

Langkah-langkah:

1. Admin masuk ke *dashboard* aplikasi *KeyKeep* dan memilih opsi “Modul Config”.
2. Admin memilih warna primary baru.
3. Admin memilih dan mengunggah file logo baru
4. Admin memilih dan mengunggah video *background* baru untuk tampilan awal
5. Admin mengunggah ikon baru seperti titip barang, ambil barang, loker terbuka, loker tertutup, ikon selesai.
6. Admin menekan tombol “simpan” untuk menerapkan perubahan.

### **1.4.15.6 Skenario Eksepsional**

Prakondisi: Admin telah masuk ke *dashboard* Modul Config.

Pascakondisi: Perubahan tidak berhasil disimpan karena file tidak valid atau terjadi kesalahan sistem.

Langkah-langkah:

1. Admin membuka halaman pengaturan tampilan pada *dashboard* dan mengunggah file yang didukung atau dengan format salah

2. Sistem memeriksa file yang diunggah dan mendeteksi ketidaksesuaian format atau kesalahan lain.
3. Sistem menampilkan pesan error kepada admin seperti “format file tidak didukung” atau “kesalahan saat mengunggah file”.
4. Admin diminta untuk mengunggah file yang sesuai dengan format yang benar.

## 1.5 Kebutuhan Nonfungsional

### 1.5.1 Atribut Kualitas

Berikut ini adalah tiga indikator atribut kualitas:

1. *User Experience (UX)* yang Baik

Deskripsi: Atribut ini menekankan pengalaman pengguna yang intuitif, efisien dan memuaskan saat menggunakan aplikasi *KeyKeep*.

Batas Minimum: Aplikasi harus dirancang dengan desain antarmuka yang responsive, navigasi yang jelas dan interaksi yang intuitif.

Pengaruh pada Bagian Sistem: *User experience* yang baik Meningkatkan penerimaan dan kepuasan pengguna, serta mengurangi kesalahan pengguna dalam penggunaan aplikasi.

2. Dukungan Bahasa (*Language Support*)

Deskripsi: Atribut ini mencakup kemampuan aplikasi untuk mendukung pengguna dengan preferensi bahasa yang lain.

Batas Minimum: Sistem harus dapat menyediakan dukungan multibahasa atau kemampuan untuk mengkonfigurasi antarmuka dalam berbagai bahasa.

Pengaruh pada Bagian Sistem: Dukungan bahasa meningkatkan aksesibilitas aplikasi dan memungkinkan pengguna dengan latar belakang bahasa yang berbeda untuk menggunakan sistem dengan lebih nyaman.

### **1.5.2 Kebutuhan Legal**

Rancang Bangun Kotak Penyimpanan Pintar berbasis KIOSK sudah disepakati melalui Surat Perjanjian Implementasi antara Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas YARSI dengan Klien.

dengan:

No. Program Studi Teknik Informatika: 049/Prodi TI/HK.20/X/2024

Yang berisikan aturan yang harus dipenuhi dalam aplikasi. Data yang didapat untuk diolah dan ditampilkan dalam aplikasi *KeyKeep* sesuai dengan keadaan loker, seperti jumlah dan ukuran.