BAB II
KAJIAN LITERATUR

2.1 Aplikasi

Aplikasi atau *software* adalah sebuah program yang dirancang dan dikembangkan untuk menyelesaikan tugas-tugas tertentu pada komputer. Aplikasi biasanya terdiri dari satu atau beberapa program yang dibuat dengan tujuan spesifik, seperti aplikasi pengolahan kata, pengolahan data, aplikasi *multimedia*, dan sebagainya. Menurut Ramdan [9] aplikasi dapat terbagi menjadi dua jenis, yaitu aplikasi berbasis *web* dan aplikasi *desktop*. Aplikasi *desktop* merupakan aplikasi yang dijalankan pada komputer yang terpasang pada sistem operasi tertentu. Sedangkan, aplikasi berbasis *web* adalah aplikasi yang diakses melalui *browser* pada perangkat komputer, tablet, atau ponsel.

Dalam penelitian ini, aplikasi yang dikembangkan adalah aplikasi *web*, yang dapat diakses melalui *browser*. Aplikasi *web* dipilih karena lebih fleksibel dalam penggunaannya dan dapat diakses oleh banyak *user* di berbagai perangkat. Aplikasi *web* dapat diakses kapan saja dan di mana saja selama tersedia koneksi internet. Aplikasi *web* juga lebih mudah dalam hal pembaruan dan pemeliharaan dibandingkan dengan aplikasi *desktop*.

2.2 Aplikasi *Web*

![Gambar 2.1 Contoh Aplikasi *Web* shopee.com](image)

Seperti contoh yang ada di atas dari shopee.com [10], aplikasi *web* adalah sebuah aplikasi yang berbasis pada teknologi *web*. Menurut Nurwanto [11], aplikasi *web* merupakan sebuah aplikasi yang dapat diakses melalui *browser* dan memanfaatkan teknologi *web*,

---

© Karya Dilindungi UU Huk Cipta
1. Dilarang menyebarluaskan dokumen tanpa izin.
2. Dilarang melakukan plagiarism.
3. Pelanggaran diberhakkan sanksi sesuai peraturan UU Huk Cipta.

Kelebihan dari aplikasi web adalah fleksibilitas dan kemudahan aksesnya. User dapat mengakses aplikasi web dari berbagai perangkat seperti desktop, laptop, tablet, dan smartphone, serta dari mana saja selama tersedia koneksi internet. Selain itu, aplikasi web juga lebih mudah dan lebih murah dibandingkan dengan aplikasi desktop, karena aplikasi web tidak memerlukan instalasi pada setiap perangkat user, melainkan hanya perlu diakses melalui browser web [2]. Selain itu, pengembangan aplikasi web juga memungkinkan pengembang untuk memperbarui dan mengembangkan aplikasi dengan lebih cepat dan mudah karena tidak perlu melakukan pembaruan secara manual pada setiap perangkat user.


Pada umumnya, aplikasi web dapat digunakan pada berbagai bidang seperti e-commerce, e-learning, dan e-government. Salah satu contoh penggunaan aplikasi web adalah platform e-commerce seperti Tokopedia dan Bukalapak yang memungkinkan user untuk melakukan transaksi jual-beli secara online. Aplikasi web juga banyak digunakan pada industri e-learning, seperti platform Coursera dan Udemy yang memungkinkan user untuk belajar secara online melalui kursus dan video pembelajaran.

2.3 Metodologi Waterfall

Metode pengembangan perangkat lunak Waterfall merupakan salah satu metode pengembangan aplikasi yang paling banyak digunakan pada pengembangan aplikasi web maupun desktop. Metode waterfall terdiri dari beberapa tahap pengembangan seperti requirement gathering, analysis, design, implementation, testing, dan maintenance [3]. Dalam metode waterfall, setiap tahap harus diselesaikan secara berurutan dan harus selesai
sebelum memulai tahap berikutnya. Metode ini cocok untuk pengembangan aplikasi dengan spesifikasi yang jelas dan tidak banyak berubah di tengah jalan [12].

Gambar 2.2 Proses Waterfall

Berikut penjelasan pada gambar di atas terkait tahap metode Waterfall [13]:

1. Tahap analisis kebutuhan

Pada tahap ini, penulis akan melakukan analisis terhadap kebutuhan sistem yang akan diterapkan pada sebuah website dengan cara melakukan wawancara dengan pemilik toko guna memahami kebutuhan dan harapan pemilik toko terhadap sistem yang akan dibuat.

2. Tahap perancangan sistem


3. Tahap coding

Setelah desain sistem selesai dibuat, maka penulis akan melakukan penulisan program menggunakan bahasa pemrograman React JS dan PostgreSQL sebagai database. Hal ini dilakukan untuk menerjemahkan desain sistem menjadi sebuah aplikasi yang dapat dijalankan oleh user.

4. Tahap testing

Setelah kode program selesai dibuat, penulis akan memulai tahap testing dengan menggunakan metode Black Box. Hal ini dilakukan untuk memeriksa segala kemungkinan kesalahan dan menemukan bug atau masalah lain pada sistem. Jika ditemukan kesalahan, maka penulis akan melakukan perbaikan dan pengembangan pada sistem agar dapat berjalan dengan baik dan sesuai dengan hasil yang diinginkan.
5. Tahap implementasi dan pemeliharaan

Tahap terakhir yaitu implementasi dan pemeliharaan aplikasi. Pada tahap ini, aplikasi sudah dapat dijalankan dan digunakan oleh user. Namun, penulis tetap harus melakukan pemeliharaan untuk mengetahui masalah atau kesalahan yang tidak terdeteksi pada tahap sebelumnya. Hal ini bertujuan untuk melakukan perbaikan dan peningkatan sistem agar dapat berjalan dengan lebih baik dan optimal.

2.4 Unified Modeling Language (UML) Diagram

*Unified Modeling Language* (UML) adalah bahasa pemodelan visual yang digunakan untuk mendefinisikan, merancang, dan membangun sistem perangkat lunak. UML digunakan untuk menggambarkan struktur dan perilaku dari sistem perangkat lunak yang akan dibangun. UML terdiri dari berbagai macam diagram yang dapat digunakan untuk menggambarkan berbagai aspek dari sistem, seperti *use case diagram*, *class diagram*, *sequence diagram*, *state diagram*, *activity diagram*, dan lain sebagainya [14].

Pentingnya UML dalam pengembangan perangkat lunak adalah memungkinkan tim pengembang perangkat lunak untuk memvisualisasikan dan memahami sistem perangkat lunak secara menyeluruh sebelum memulai proses pengembangan. Dengan menggunakan UML, tim pengembang perangkat lunak dapat memastikan bahwa mereka memiliki pemahaman yang sama tentang sistem perangkat lunak yang akan dibangun, dan dapat berkomunikasi dengan jelas dan efektif. Hal ini akan membantu meningkatkan efisiensi dan efektivitas dalam proses pengembangan perangkat lunak, dan memastikan bahwa sistem perangkat lunak yang dihasilkan berkualitas tinggi dan memenuhi kebutuhan user.
1. **Use-case Diagram**

Gambar 2.3 Contoh Use-case Diagram

Seperti yang didemonstrasikan oleh Yassine [15] pada gambar diatas. Dalam pengembangan aplikasi Point of Sales (POS) berbasis web untuk Apotek Sehati, tahap analisis menggunakan use-case diagram untuk menggambarkan interaksi antara user dan sistem perangkat lunak. Use-case diagram adalah salah satu jenis diagram UML yang digunakan untuk menggambarkan interaksi antara user dan sistem perangkat lunak, serta mengidentifikasi kebutuhan fungsional dari sistem. Diagram ini terdiri dari actor, use case, dan relasi antara actor dan use case. Actor adalah entitas luar yang berinteraksi dengan sistem, sedangkan use case adalah kumpulan tindakan yang dilakukan oleh sistem dalam merespons permintaan dari actor. Relasi antara actor dan use case dapat berupa relasi include, extend, dan generalization [16].

2. **Database Diagram**

Gambar 2.4 Diagram Database
Selain itu, Diagram Database seperti yang ditampilkan pada gambar diatas dari dbdiagram.io [17] digunakan untuk memodelkan hubungan antar entitas dalam basis data aplikasi. Diagram Database membantu memastikan bahwa hubungan antar entitas terdefinisi dengan jelas dan tidak terjadi redundansi dalam basis data aplikasi [18].

3. Activity Diagram

![Gambar 2.5 Activity Diagram](image)

Seperti yang didemonstrasikan oleh Cvitić et al. [19] pada gambar di atas, Diagram ini digunakan untuk memodelkan alur kerja atau aktivitas dalam proses aplikasi Point of Sales (POS) berbasis web untuk Apotek Sehati. Dalam activity diagram, terdapat node yang digunakan untuk menggambarkan aktivitas atau proses dalam suatu sistem dan edge yang digunakan untuk menghubungkan antara node satu dengan node lainnya dan menunjukkan alur dari suatu proses [20].

Dengan menggunakan UML dalam pengembangan aplikasi Point of Sales (POS) berbasis web untuk Apotek Sehati, tim peneliti dapat memastikan bahwa sistem perangkat lunak yang dibangun memiliki kualitas tinggi, efisien, dan efektif. UML membantu tim pengembang untuk memvisualisasikan dan memahami sistem perangkat lunak secara menyeluruh sebelum memulai proses pengembangan, dan memastikan bahwa sistem perangkat lunak yang dihasilkan memenuhi kebutuhan user.

2.5 Point of Sales

Teknologi informasi adalah salah satu hal yang sangat penting dan banyak dipergunakan dalam kehidupan sehari-hari. Kehadiran teknologi informasi telah membawa perubahan besar dalam kehidupan manusia, terutama dalam bidang komunikasi dan informasi. Dengan teknologi informasi, manusia dapat mengakses informasi dari seluruh dunia dengan mudah dan cepat. Salah satu teknologi informasi yang sangat populer dan banyak digunakan adalah sistem Point of Sales (POS). Sistem POS adalah sebuah sistem...
aplikasi yang diterapkan pada bisnis minimarket atau pertokoan untuk menangani pengolahan data transaksi pembelian, transaksi penjualan eceran, transaksi hutang, transaksi retur pembelian, dan pelaporan transaksi yang secara umum penting dibutuhkan dalam pengambilan keputusan strategik oleh para pebisnis swalayan, organisasi, atau perusahaan yang berskala kecil dan menengah. Perusahaan besar maupun perusahaan menengah yang bergerak di dalam bidang perdagangan pada umumnya akan menggunakan sistem POS dalam menunjang kegiatan usahanya [21].


Ada tiga jenis Arsitektur sistem POS yang umum digunakan, yaitu Arsitektur two-tier, three-tier, dan n-tier:

Sistem Point of Sales (POS) memang dirancang untuk membantu meminimalkan kesalahan manusia dalam pengolahan data transaksi bisnis. Hal ini sangat penting karena kesalahan manusia dapat berdampak sangat buruk bagi bisnis. Kesalahan manusia dalam
pengolahan data transaksi bisnis dapat terjadi pada berbagai tahapan proses, mulai dari input data, penghitungan harga, hingga pengecekan stok barang. Misalnya, kesalahan saat meng-input data bisa saja terjadi ketika kasir atau admin salah memasukkan harga barang atau jumlah barang yang dibeli oleh pelanggan. Hal ini dapat menyebabkan terjadinya ketidakseimbangan antara persediaan barang dan laporan keuangan bisnis [22].

Tidak hanya itu, kesalahan manusia juga dapat terjadi saat pengecekan stok barang. Jika admin tidak melakukan pengecekan stok barang secara teratur, maka bisa saja terjadi ketidakteraturan dalam pengisian stok barang. Akibatnya, pelanggan mungkin kecewa karena barang yang mereka butuhkan tidak tersedia. Namun, dengan adanya sistem POS, semua proses transaksi bisnis dapat tercatat secara otomatis tanpa adanya kesalahan manusia. Sistem ini dapat mengurangi kesalahan manusia dalam pengolahan data transaksi bisnis dan memastikan bahwa semua transaksi tercatat dengan akurat. Dengan sistem ini, kasir atau admin tidak perlu lagi memasukkan data secara manual, yang dapat meminimalkan kesalahan pada saat input data [23].

2.6 Arsitektur Model-View-Controller (MVC)

Model-View-Controller (MVC), merupakan salah satu arsitektur untuk membangun aplikasi berbasis web dengan memisahkan aplikasi menjadi tiga bagian yang terpisah, yaitu model, view, dan controller [24]:

1. Model

Model adalah bagian dari aplikasi yang bertanggung jawab untuk mengatur data dan logika bisnis. Dalam sistem Point of Sales (POS) berbasis web pada Apotek Sehati, model terdiri dari database yang menyimpan data transaksi dan data barang, serta aplikasi web yang mengakses database dan memproses data transaksi dan data barang.

2. View

View adalah bagian dari aplikasi yang bertanggung jawab untuk menampilkan data kepada user. Dalam sistem Point of Sales (POS) berbasis web pada Apotek Sehati, view terdiri dari halaman web yang menampilkan data transaksi dan laporan.

3. Controller

Controller adalah bagian dari aplikasi yang bertanggung jawab untuk mengatur alur program dan interaksi antara model dan view. Dalam sistem Point of Sales (POS) berbasis web pada Apotek Sehati, controller terdiri dari aplikasi web yang mengatur input dan output data, serta memproses data transaksi dan data barang.
Dengan menggunakan pendekatan MVC, sistem Point of Sales (POS) berbasis web pada Apotek Sehati dapat dipecah menjadi tiga bagian yang terpisah, sehingga memudahkan dalam pengembangan, pengujian, dan perawatan. Model dapat diubah tanpa mempengaruhi view dan controller, begitu pula sebaliknya. Selain itu, pendekatan MVC juga memudahkan dalam pengembangan aplikasi berbasis web yang kompleks dan skala besar.

2.7 Profil Apotek Sehati

![Gambar 2.6 Apotek Sehati](image)

Gambar 2.6 Apotek Sehati

Apotek Sehati didirikan pada tahun 1999 dan merupakan sebuah usaha yang bergerak di bidang farmasi yang menyediakan obat-obatan, suplemen, dan jamu-jamu bagi masyarakat di sekitar. Pelanggan dapat membeli obat secara langsung dengan resep dokter atau tanpa resep.

2.7.1 Lokasi

2.7.2 Struktur Organisasi

Berikut adalah struktur organisasi yang sederhana dari Apotek Sehati:

1. Pemilik:
   a. Bertanggung jawab atas pengelolaan keseluruhan Apotek Sehati.
   b. Melakukan pengawasan langsung terhadap staff kasir dan penjualan.
   c. Melakukan evaluasi keuangan secara bulanan untuk memastikan catatan keuangan telah sesuai dan kondisi perusahaan dalam kondisi yang baik.

2. Staff Kasir:
   a. Melakukan transaksi penjualan di kasir.
   b. Menangani pembayaran dari pelanggan.
   c. Melakukan pencatatan transaksi penjualan.

3. Staff Penjualan:
   a. Menjaga ketersediaan stok barang di dalam Apotek Sehati.
   b. Melakukan pemantauan terhadap ketersediaan barang yang ada di dalam sistem.
   c. Menangani permintaan dari pelanggan tentang informasi barang dan stok yang tersedia di Apotek Sehati.

2.7.3 Izin Usaha

Apotek Sehati adalah sebuah usaha perorangan yang diawasi oleh pemilik sarana apotek (PSA) dan dioperasikan oleh satu apoteker dengan bantuan beberapa asisten apoteker.
Usaha ini berbadan hukum dan telah mendapatkan izin dari Dinas Kesehatan dan Balai Pengawasan Obat dan Makanan (BPOM).

### 2.7.4 Proses Kerja

Proses kerja Apotek Sehati sebagian besar dilakukan di dalam toko seperti pada gambar di atas. Proses tersebut dimulai dengan pengadaan barang dari pemasok. Setelah barang tiba, staf penjualan akan memeriksa dan memastikan bahwa barang yang diterima sesuai dengan pesanan dan kondisi barang dalam keadaan baik. Setelah itu, staf penjualan akan memasukkan informasi barang ke dalam sistem dan menambahkan stok ke dalam sistem.

Ketika pelanggan membutuhkan obat atau produk kesehatan, mereka akan datang ke Apotek Sehati dan berkonsultasi dengan staf penjualan mengenai obat atau produk yang dibutuhkan. Staf penjualan akan memberikan informasi yang diperlukan seperti jenis obat, dosis, dan efek samping. Setelah itu, pelanggan dapat memilih produk dan melakukan pembayaran di kasir yang ditangani oleh staf kasir. Setelah pembayaran selesai, staf kasir akan mencetak struk pembelian dan staf penjualan akan mencatat transaksi penjualan ke dalam sistem.

Apetek Sehati menghadapi beberapa masalah yang kompleks, seperti pengelolaan penjualan obat yang masih menggunakan nota manual, dan pengelolaan inventaris obat. Untuk mengatasi masalah tersebut, pemilik Apotek Sehati merasa sangat membutuhkan sistem komputerisasi yang dapat menangani masalah di atas. Sistem baru tersebut diharapkan dapat memudahkan pengolahan dan pengaksesan data serta mengurangi tingkat kesalahan.

![Gambar 2.8 Nota Apotek Sehati](image)

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarkan dokumen tanpa izin.
2. Dilarang melakukan plagiasi.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.
2.8 Model End User Computing Satisfaction (EUCS)

Setelah pembayaran berhasil dilakukan, obat akan diberikan kepada pembeli. Selain itu, apoteker juga harus memberikan informasi dan saran yang tepat tentang penggunaan obat dan efek samping yang mungkin terjadi. Hal ini penting untuk melindungi kesehatan pelanggan dan memastikan bahwa obat yang dikonsumsi aman dan efektif.

Model End User Computing Satisfaction (EUCS) adalah sebuah model yang digunakan untuk mengevaluasi kepuasan user terhadap sistem TIK (Teknologi Informasi dan Komunikasi) dalam lingkungan bisnis. Model ini dikembangkan berdasarkan konsep-konsep dari literatur tentang kepuasan user dan faktor-faktor yang mempengaruhi kepuasan user terhadap sistem TIK. Model EUCS terdiri dari dua faktor utama yang mempengaruhi kepuasan pengguna, yaitu faktor teknis dan faktor non-teknis. Faktor teknis meliputi kualitas sistem, kinerja sistem, dan kemudahan penggunaan. Faktor non-teknis meliputi dukungan teknis, dukungan manajemen, dan kepercayaan user terhadap sistem [25].

Untuk mengukur kepuasan user menggunakan Model EUCS, digunakan Kuesioner yang terdiri dari pertanyaan tentang faktor-faktor teknis dan non-teknis tersebut. Setiap pertanyaan diukur menggunakan skala Likert, yaitu dari 1 sampai 5 atau 1 sampai 7, yang menunjukkan tingkat kepuasan user terhadap setiap faktor. Hasil dari Kuesioner kemudian dihitung untuk menghasilkan nilai keseluruhan kepuasan user terhadap sistem TIK yang digunakan. Selain itu, faktor-faktor yang paling berpengaruh pada kepuasan user juga dapat diidentifikasi dan digunakan untuk meningkatkan kepuasan user dalam penggunaan sistem TIK.

Model EUCS telah digunakan dalam banyak penelitian untuk mengevaluasi kepuasan user terhadap sistem TIK dalam lingkungan bisnis. Model ini dapat membantu organisasi dalam meningkatkan kinerja sistem TIK dan kepuasan pengguna, serta mengidentifikasi faktor-faktor yang perlu ditingkatkan untuk memenuhi kebutuhan pengguna. Oleh karena itu penelitian ini akan memanfaatkan model tersebut untuk mengevaluasi kepuasan pelanggan yakni dari pihak manajemen Apotek Sehati.
Gambar 2.9 Dimensi EUCS

Terdapat beberapa dimensi yang diukur dalam model EUCS, salah satunya adalah sebagai berikut:

1. **Content**: Dimensi ini mengukur sejauh mana user merasa puas dengan konten yang disediakan oleh sistem. Konten yang baik adalah konten yang relevan, lengkap, dan mudah dipahami.

2. **Accuracy**: Dimensi ini mengukur sejauh mana user merasa sistem memberikan informasi yang akurat dan tepat waktu. Informasi yang akurat dan tepat waktu sangat penting dalam mendukung keputusan bisnis yang baik.

3. **Format**: Dimensi ini mengukur sejauh mana user merasa sistem menyediakan format yang mudah dipahami dan mudah digunakan. Format yang baik adalah format yang intuitif, konsisten, dan mudah dinavigasi.

4. **Ease of use**: Dimensi ini mengukur sejauh mana user merasa sistem mudah digunakan. Sistem yang mudah digunakan dapat menghemat waktu dan usaha pengguna, dan meningkatkan produktivitas dan efisiensi kerja.

5. **Timeliness**: Dimensi ini mengukur sejauh mana user merasa sistem memberikan informasi atau layanan tepat waktu. Sistem yang memberikan informasi atau layanan tepat waktu dapat membantu user membuat keputusan yang tepat dan menghindari kerugian.
2.9 Black Box Testing

Pengujian perangkat lunak dapat dilakukan dengan menggunakan beberapa teknik yang berbeda, termasuk teknik All Pair Testing, Boundary Value Analysis, Cause-Effect Graph, Equivalence Partitioning, Fuzzing, Orthogonal Array Testing, dan State Transition. Salah satu teknik yang mudah digunakan adalah Black Box Testing, yang fokus pada pengujian spesifikasi fungsional dari perangkat lunak tanpa menguji desain atau kode programnya. Dalam metode ini, estimasi banyaknya data uji dapat dihitung berdasarkan field data entri, aturan entri, dan kasus batas atas dan batas bawah [21].

Contoh pengujian perangkat lunak menggunakan Black Box Testing dilakukan pada aplikasi Point of Sales untuk memastikan bahwa sistem berfungsi dengan baik. Setiap fitur pada sistem diuji dengan mengatur kondisi tertentu dan membandingkan hasilnya dengan harapan yang diinginkan. Tabel hasil pengujian menunjukkan bahwa semua fitur berjalan dengan sesuai. Teknik-teknik pengujian perangkat lunak ini dapat digunakan dalam penelitian untuk memastikan kualitas perangkat lunak yang dikembangkan. Dengan memilih teknik yang sesuai dan melakukan pengujian dengan benar, dapat meningkatkan kualitas perangkat lunak yang akan digunakan.

Contoh pengujian pada aplikasi Point of Sales menggunakan black box testing untuk memastikan bahwa sistem yang dikembangkan berfungsi dengan baik. Pengujian ini akan menguji setiap fitur pada sistem apakah berjalan dengan sesuai atau belum. Tabel berikut menunjukkan hasil pengujian dengan kondisi tertentu dan harapan yang diinginkan [17].
<table>
<thead>
<tr>
<th>No.</th>
<th>Pengujian</th>
<th>Kondisi</th>
<th>Harapan</th>
<th>Hasil</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Login</td>
<td><em>Username password</em> salah atau <em>input-an</em> salah</td>
<td>Menampilkan halaman <em>login</em> dan menampilkan pemberitahuan</td>
<td>Sesuai</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>Username atau password</em> benar</td>
<td><em>Dashboard</em> masuk ke halaman sesuai hak akses pengguna</td>
<td>Sesuai</td>
</tr>
<tr>
<td>2.</td>
<td>Logout</td>
<td>Keluar dari sistem berhasil <em>logout</em></td>
<td>Menampilkan halaman <em>login</em> dan menampilkan pemberitahuan</td>
<td>Sesuai</td>
</tr>
<tr>
<td>3.</td>
<td>Transaksi</td>
<td><em>User</em> melakukan transaksi</td>
<td>Transaksi berhasil dilakukan dan menampilkan halaman detail transaksi</td>
<td>Sesuai</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>User</em> mencetak detail traksaksi</td>
<td><em>Detail transaksi</em> berhasil mencetak detail transaksi</td>
<td>Sesuai</td>
</tr>
<tr>
<td>4.</td>
<td>Produk</td>
<td><em>User</em> menambah produk</td>
<td>Berhasil melakukan tambah produk dan balik ke halaman produk</td>
<td>Sesuai</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>User</em> mengubah produk</td>
<td><em>Produk</em> berhasil mengubah produk dan balik ke halaman produk</td>
<td>Sesuai</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>User</em> menghapus produk</td>
<td><em>Produk</em> berhasil menghapus produk dan balik ke halaman produk</td>
<td>Sesuai</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>User</em> melihat detail produk</td>
<td>Menampilkan <em>popup</em> detail produk</td>
<td>Sesuai</td>
</tr>
<tr>
<td>5.</td>
<td>Daftar Pegawai</td>
<td><em>User</em> menambah akun</td>
<td>Berhasil mengubah akun dan balik ke halaman daftar pegawai</td>
<td>Sesuai</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>User</em> menghapus akun</td>
<td><em>Akun</em> berhasil menghapus akun dan balik ke halaman daftar pegawai</td>
<td>Sesuai</td>
</tr>
<tr>
<td></td>
<td></td>
<td><em>User</em> melihat detail akun</td>
<td>Menampilkan <em>popup</em> detail akun</td>
<td>Sesuai</td>
</tr>
<tr>
<td>6.</td>
<td>Laporan Penjualan</td>
<td><em>User</em> meng-&lt;br&gt;inputkan tanggal</td>
<td>Menampilkan laporan penjualan berdasarkan <em>input-an</em> pengguna</td>
<td>Sesuai</td>
</tr>
<tr>
<td></td>
<td>Pengaturan Akun</td>
<td>User mengubah data diri</td>
<td>Menampilkan halaman pengaturan akun dan menampilkan pemberitahuan berhasil</td>
<td>Sesuai</td>
</tr>
<tr>
<td>---</td>
<td>----------------</td>
<td>-------------------------</td>
<td>--------------------------------------------------------------------------------</td>
<td>--------</td>
</tr>
</tbody>
</table>

© Karya Dilindungi UU Hak Cipta
1. Dilarang menyebarkannya dokumen tanpa izin.
2. Dilarang melakukan plagiaris.
3. Pelanggaran diberlakukan sanksi sesuai peraturan UU Hak Cipta.