Contents list avaliable at Directory of Open Access Journals (DOAJ)

JUTIN : Jurnal Teknik Industri Terintegrasi

Volume 8 Issue 3 2025, Page 3078-3089

ISSN: 2620-8962 (Online)

Journal Homepage: https://journal.universitaspahlawan.ac.id/index.php/jutin/index

Pendekatan Metode S-SDLC untuk Perancangan Aplikasi Kasir Pada Toko Aldy Jaya Berbasis *Website*

Aldy Irfansyah[1]✉, Elvi Rahmi[1]

[(1)]Program Studi Keamanan Sistem Informasi, Jurusan Teknik Informatika, Politeknik Negeri Bengkalis, Riau, Indonesia

DOI: 10.31004/jutin.v8i3.47689

✉ Corresponding author:
[cdraldyptra@gmail.com]

Article Info

Kata kunci:
Metode S-SDLC;
Aplikasi Kasir;
Keamanan Sistem;
Pengelolaan Stok

Abstrak

Toko Aldy Jaya merupakan usaha yang menjual berbagai kebutuhan seperti makanan, perlengkapan pancing, perlengkapan ikan, dan perlengkapan burung. Saat ini, toko tersebut belum memiliki sistem pencatatan transaksi maupun pengelolaan stok barang, yang mengakibatkan kesulitan dalam memperoleh laporan penjualan dan seringkali mengecewakan pelanggan karena ketidaktersediaan informasi stok secara real-time. Oleh karena itu, penelitian ini bertujuan untuk merancang aplikasi kasir berbasis web yang dapat mencatat transaksi penjualan dan memantau stok barang secara efektif, guna memudahkan pelaporan dan pengelolaan toko. Pengembangan aplikasi ini menggunakan pendekatan *Secure Software Development Life Cycle* (S-SDLC), yang mencakup identifikasi kebutuhan keamanan, perancangan sistem yang aman, pengembangan kode yang terlindungi, pengujian keamanan, penerapan aplikasi, serta pemeliharaan keamanan. Hasil dari pengembangan menunjukkan bahwa aplikasi kasir ini mampu meningkatkan efisiensi operasional dan menjaga keamanan data. Melalui penerapan klasifikasi data dan kontrol akses berbasis peran, aplikasi ini tidak hanya berfungsi sebagai alat pencatatan, tetapi juga mampu melindungi data dari ancaman, menjaga integritas informasi, serta memastikan ketersediaan sistem secara optimal.

Keywords:
S-SDLC Method;
Cashier Application;
System Security;
Stock Management

Abstract

Aldy Jaya Store is a business that sells various necessities such as food, fishing equipment, fish supplies, and bird supplies. Currently, the store does not have a transaction recording system or stock management, which results in difficulties in obtaining sales reports and often disappoints customers due to the lack of real-time stock information. Therefore, this research aims to design a web-based cash register application that can effectively record sales transactions and monitor stock items, to facilitate reporting and store management. The development of this

Received 13 June 2025; Received in revised form 19 June 2025 year; Accepted 27 June 2025
Available online 5 July 2025 /

application uses the Secure Software Development Life Cycle (S-SDLC) approach, which includes identifying security needs, designing a secure system, developing protected code, testing security, implementing applications, and maintaining security. The results from the development show that this cashier application is capable of improving operational efficiency and maintaining data security. Through the implementation of data classification and role-based access control, this application not only serves as a recording tool but is also able to protect data from threats, maintain information integrity, and ensure optimal system availability.

1. PENDAHULUAN

Kemajuan teknologi informasi telah mendorong banyak pelaku usaha untuk mengadopsi sistem digital guna meningkatkan efisiensi operasional. Salah satu aspek penting dalam dunia usaha adalah pengelolaan transaksi dan stok barang secara terstruktur dan real-time (Pangestu & Astutik, 2024).

Toko Aldy Jaya merupakan toko yang menjual makanan serta berbagai perlengkapan seperti alat pancing, perlengkapan ikan, dan perlengkapan burung. Saat ini, pencatatan transaksi dan pengelolaan stok barang masih dilakukan secara manual, yang mengakibatkan kesulitan dalam memperoleh laporan penjualan serta ketidakpastian ketersediaan barang bagi pelanggan.

Untuk mengatasi masalah tersebut, diperlukan perancangan aplikasi kasir berbasis web yang dapat mencatat transaksi penjualan dan memantau stok barang secara otomatis. Beberapa penelitian sebelumnya, seperti yang dilakukan oleh Hasan, Suhermanto, dan Suharmanto (2021), menunjukkan bahwa penerapan metode Secure Software Development Life Cycle (S-SDLC) dapat meningkatkan keamanan sistem perangkat lunak secara menyeluruh. Selain itu, penelitian oleh Pangestu dan Astutik (2024) juga menunjukkan efektivitas sistem kasir berbasis web dalam mendukung efisiensi operasional toko. Berdasarkan hasil studi-studi tersebut, aplikasi ini dirancang menggunakan metode S-SDLC yang menekankan aspek keamanan pada setiap tahap pengembangan sistem, mulai dari identifikasi kebutuhan hingga pemeliharaan sistem.

Penelitian ini bertujuan untuk merancang dan mengembangkan aplikasi kasir yang dapat digunakan oleh Toko Aldy Jaya guna mendukung kegiatan operasional toko secara lebih efisien dan aman. Dalam proses pengembangannya, digunakan pendekatan *Secure Software Development Life Cycle* (SSDLC) untuk memastikan setiap tahapan pembangunan perangkat lunak memperhatikan aspek keamanan sejak awal. Aplikasi ini diharapkan dapat menyederhanakan proses pencatatan transaksi, meminimalkan kesalahan manusia, dan memberikan laporan penjualan yang jelas serta informasi stok yang akurat (Nuryamin & Risyda, 2021).

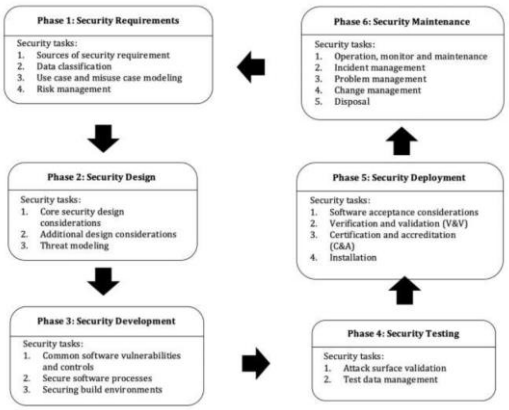
Untuk menjaga fokus dan ruang lingkup penelitian, pembatasan diterapkan pada pendekatan metode S-SDLC, di mana hanya satu tugas keamanan yang diterapkan pada setiap tahap. Pada tahap *Security Requirements* digunakan *Data Classification*, tahap *Security Design* menggunakan *Core Security Design Considerations*, tahap *Security Development* menerapkan *Common Software Vulnerabilities and Controls*, tahap *Security Testing* difokuskan pada *Test Data Management*, tahap *Security Deployment* mencakup *Installation*, dan tahap *Security Maintenance* menggunakan pendekatan *Problem Management*.

Berdasarkan uraian tersebut, penulis bertujuan membuat aplikasi kasir berbasis website dengan mengangkat judul "Pendekatan Metode S-SDLC Untuk Perancangan Aplikasi Kasir Pada Toko Aldy Jaya Berbasis Website"

2. METODE

Untuk menyelesaikan permasalahan di atas, dengan perancangan aplikasi kasir berbasis web untuk Toko Aldy Jaya ini menggunakan pendekatan metode S-SDLC. *Secure Software Development Life Cycle* (S-SDLC) adalah konsep metodologis yang termasuk dalam *Software Development Life Cycle*, yang mencakup analisis, desain, implementasi (pembuatan), pengujian, dan evaluasi (penyebaran dan pemeliharaan). S-SDLC merupakan pendekatan yang menekankan integrasi aspek keamanan ke dalam seluruh tahapan siklus pengembangan perangkat lunak. Membangun perangkat lunak yang aman bukanlah hal yang sederhana, namun dapat dicapai dengan memperbaiki proses pengembangannya guna mengurangi potensi kerentanan yang muncul. Proses S-SDLC mencakup berbagai praktik serta aktivitas yang berorientasi pada keamanan,

dan penerapannya secara tepat dapat secara signifikan meningkatkan tingkat keamanan perangkat lunak yang dikembangkan (Hasan, Suhermanto, & Suharmanto, 2021).



Gambar 1 Metode S-SDLC
Sumber : CyberSecurity Malaysia (2020)

A. *Security Requirements*

Data classification bertujuan untuk mengidentifikasi informasi yang dianggap sebagai aset digital paling bernilai, sehingga dapat diberikan perlindungan secara maksimal dan tepat sasaran (CyberSecurity Malaysia, 2020).

B. *Security Design*

Core security design considerations merupakan pertimbangan utama dalam desain keamanan melibatkan penyusunan sistem perangkat lunak yang mampu menangani aspek-aspek penting seperti kerahasiaan, integritas, ketersediaan, autentikasi, otorisasi, dan pencatatan audit (CyberSecurity Malaysia, 2020).

C. *Security Development*

Common software vulnerabilities and controls dalam perangkat lunak dan penerapan pengendalian bertujuan untuk mencegah kelemahan akibat praktik pengkodean yang tidak aman serta melindungi sistem dari potensi serangan (CyberSecurity Malaysia, 2020).

D. *Security Testing*

Test data management digunakan untuk menetapkan data input serta output yang diharapkan, guna memastikan perangkat lunak berjalan dengan baik sesuai fungsinya (CyberSecurity Malaysia, 2020).

E. *Security Deployment*

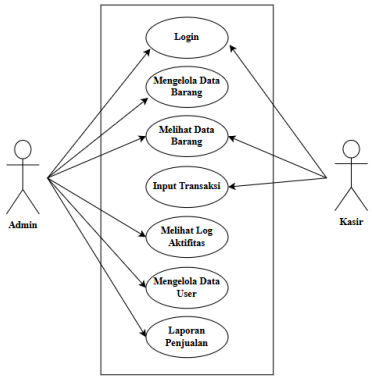
Installation difokuskan pada pengamanan lingkungan produksi serta pengaturan konfigurasi aplikasi secara tepat agar sistem dapat beroperasi dengan aman (CyberSecurity Malaysia, 2020).

F. *Security Maintenance*

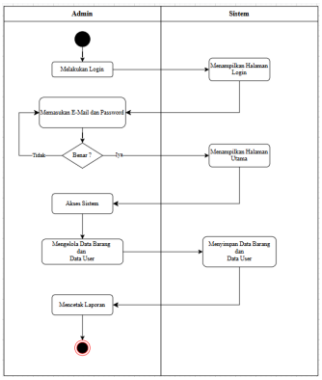
Problem management bertujuan untuk mendeteksi serta menangani akar penyebab dari gangguan yang belum teridentifikasi, sekaligus meningkatkan kualitas layanan perangkat lunak agar masalah yang sama tidak terulang di masa depan (CyberSecurity Malaysia, 2020).

3. HASIL DAN PEMBAHASAN

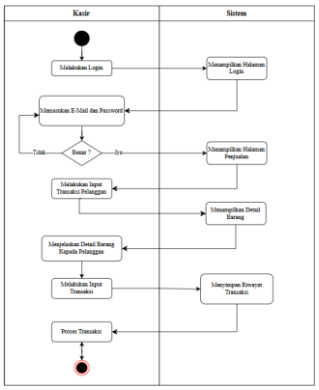
Tahap pertama ini melibatkan pembuatan ide untuk desain atau presentasi situs web yang ingin dirancang. Pembuatan konsep diawali dengan menyusun *usecase*, dan *activity diagram*. Hal ini dilakukan dengan mendeskripsikan solusi yang dibutuhkan untuk mencapai hasil desain yang diinginkan.



Gambar 2. *Usecase Diagram*



Gambar 3. *Activity Diagram* Admin



Gambar 4. *Activity Diagram* Kasir

Pada proses admin, pertama kali admin melakukan login dengan memasukkan email dan password dan akan dicek apakah email dan password benar tidak lupa mengecek level pengguna admin atau kasir. Masuk level sebagai admin akan diarahkan ke halaman utama admin, disini admin dapat mengelola data user dan juga produk baik itu menambah, mengedit maupun menghapus.

Pada proses kasir sama dengan admin melakukan login dan mengecek email dan password apakah benar dan level sebagai kasir. Masuk sebagai kasir maka dapat mengelola transaksi penjualan.

3.1 *Security Requirements*

Tahapan pertama ini menggunakan tugas keamanan *Data Classification* (klasifikasi data) merupakan keamanan yang melindungi data sebagai aset digital yang paling berharga dari potensi ancaman, pelanggaran privasi, atau penyalah gunaan. Melakukan wawancara dan analisa untuk memisahkan mana data yang publik dan rahasia, sehingga memudahkan dalam pemberian label pada data baik itu publik, internal dan rahasia. Pemberian label pada data berdasarkan tingkat sensitivitasnya:

1. Publik

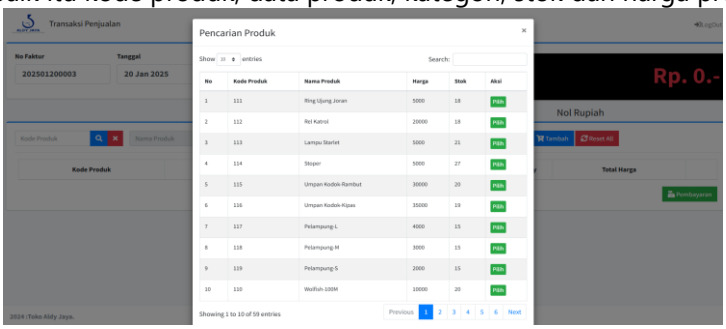
Dalam label publik ini berisikan data yang bisa dilihat siapa saja, baik itu admin, kasir maupun pelanggan. Pada label publik ini juga memberikan informasi terkait nama produk dan harga produk, untuk harga produk sudah di tuliskan pada rak produk tersebut dan dibawah ini tabel semua produk yang dijual.

Tabel 1. Data Produk dan harga

NO	NAMA	KATEGORI	HARGA
1	Ring Ujung Joran	Peralatan Pancing	5.000
2	Rel Katrol	Peralatan Pancing	20.000
3	Lampu/Starlet	Peralatan Pancing	5.000
4	Pelampung L	Peralatan Pancing	4000
5	Pelampung M	Peralatan Pancing	3000
6	Pelampung S	Peralatan Pancing	2000
7	Tempat Minum/Makan M	Peralatan Burung	4.000
8	Tempat Minum/Makan S	Peralatan Burung	3.000
9	Sangkar Besi Panjang	Peralatan Burung	175.000
10	Sangkar Kepala Burung M	Peralatan Burung	75.000
11	Sangkar Kepala Burung S	Peralatan Burung	65.000
12	Serokan L	Peralatan Aquarium	35.000
13	Serokan M	Peralatan Aquarium	30.000
14	Serokan S	Peralatan Aquarium	25.000
15	Penjernih Air	Peralatan Aquarium	8.000
16	Batu/Pasir	Peralatan Aquarium	15.000/kg
17	All Feed 2	Makanan Ikan	12.000
18	All Feed 3	Makanan Ikan	12.000
19	All Feed 4	Makanan Ikan	12.000
20	Pakan Cupang	Makanan Ikan	10.000
21	FF-999	Makanan Ikan	23.000
22	Takari	Makanan Ikan	5.000
23	Ebod Kenari	Makanan Burung	15.000
24	Ebod Lovberd	Makanan Burung	15.000
25	Leopat	Makanan Burung	9.000
26	Juara Kuning	Makanan Burung	13.000

2. Internal

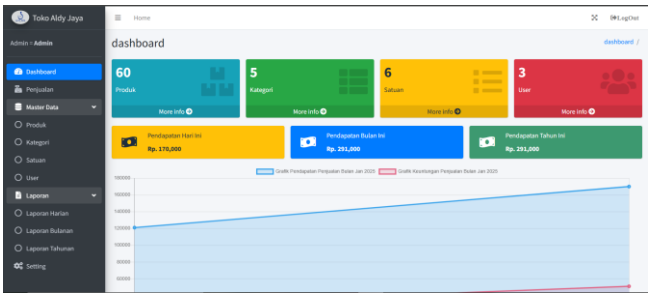
Label internal ini yang dapat mengolah hanya kasir dan selain kasir tidak diperbolehkan, karena berisikan data penjualan baik itu kode produk, data produk, kategori, stok dan harga produk.



Gambar 5. Dashboard penjualan

3. Rahasia

Pada label rahasia ini yang boleh mengakses hanya admin saja, yang didalamnya berisi data produk, data user, dan laporan data penjualan. Sehingga, selain admin tidak dapat mengakses atau melihat informasi tersebut karena sangat rahasia.



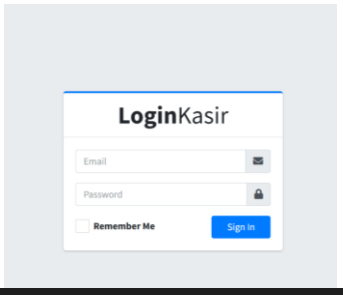
Gambar 6. Dashboard admin

3.2 *Security Design*

Tahapan kedua ini menggunakan tugas keamanan *Core Security Design Considerations* (pertimbangan desain keamanan inti). Dalam hal ini ada bebearapa keamanan yang harus dipenuhi yaitu, Confidentiality, Integrity, Availability, Authentication, Authorization, dan Accountability.

1. *Confidentiality*

Tujuan dari keamanan ini adalah melindungi informasi dari akses yang tidak sah sehingga data hanya dapat diakses oleh pihak yang berwenang. Maka dari itu dibuat halaman login sebagai pintu masuk pada website dan juga sebagai keamanan pada website dan juga memastikan apakah E-mail atau password benar ataupun salah.

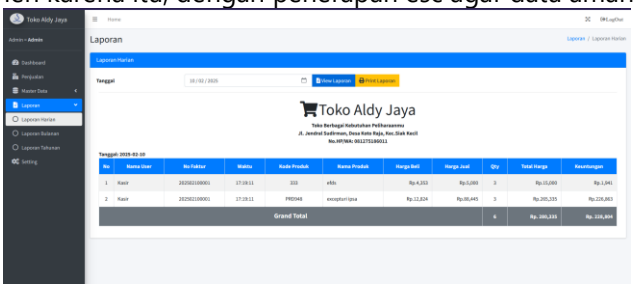



```php
if ($cek_login) {
    session()->set(data: 'id_user', value: $cek_login['id_user']);
    session()->set(data: 'nama_user', value: $cek_login['nama_user']);
    session()->set(data: 'level', value: $cek_login['level']);
```

Gambar 7. Tampilan dan source code cek login

2. *Integrity*

Dalam integritas ini memiliki tujuan untuk memastikan bahwa data tidak dapat diubah tanpa otorisasi, sehingga data tetap akurat dan konsisten. Sehingga aplikasi kasir ini diterapkan pada data penjualan, karena data yang sangat penting. Oleh karena itu, dengan penerapan esc agar data aman dari serangan XSS.




```php
public function CekProduk(): ResponseInterface
{
    $kode_produk = esc(data: $this->request->getPost(index: 'kode_produk'));
    $produk = $this->ModelPenjualan->CekProduk(kode_produk: $kode_produk);

    if (!$produk || $produk['status'] !== 'aktif') {
        return $this->response->setJSON(body: ['error' => 'Produk tidak ditemukan atau sudah non-aktif!']);
```

Gambar 8. Tampilan laporan dan Penerapan esc

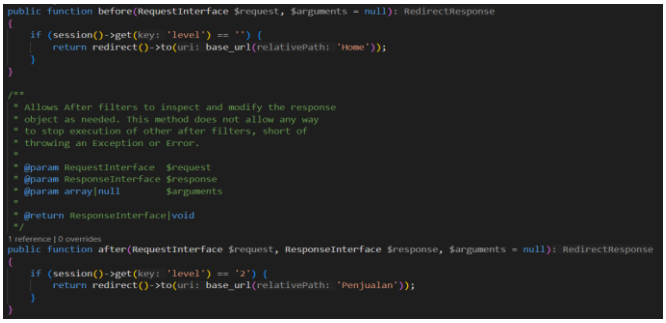
3. *Availability*

Tujuan dari *Availability* adalah memastikan bahwa sistem tetap berfungsi dan memberikan pembatasan akses sesuai dengan peran, dengan melakukan pembagian hak akses saat login dengan filter. FilterAdmin merupakan membuat hak akses pada admin sesuai perannya. Dengan menggunakan fungsi before() untuk memastikan bila ada pengguna yang tidak memiliki level 1 (admin) dalam session akan diarahkan ke halaman home/login kembali, sedangkan fungsi after() memeriksa level dan apakah benar pengguna memiliki level admin makan akan masuk ke halaman admin.

```php
public function before(RequestInterface $request, $arguments = null)
{
    if (session()->get('level') == '') {
        return redirect()->to(base_url('Home'));
    }
}

/**
 * Allows After filters to inspect and modify the response
 * object as needed. This method does not allow any way
 * to stop execution of other after filters, short of
 * throwing an Exception or Error.
 *
 * @param RequestInterface  $request
 * @param ResponseInterface $response
 * @param array|null        $arguments
 *
 * @return ResponseInterface|void
 */
public function after(RequestInterface $request, ResponseInterface $response, $arguments = null)
{
    if (session()->get('level') == '1') {
        return redirect()->to(base_url('Admin'));
    }
}
```

Gambar 9. Source code hak akses admin

FilterKasir sama dengan FilterAdmin yang memiliki fungsi before() untuk mengecek apakah memiliki pengguna memiliki peran level 2 (kasir) dan apabila tidak ada maka akan diarah ke halaman home/login. Sementara itu, juga ada fungsi after() untuk memeriksa apakah pengguna memiliki peran kasir maka akan masuk ke halaman penjualan.



Gambar 10. Source code hak akses kasir

4. *Authentication* (Autentikasi)

Merupakan keamanan pada hak akses didalam sistem untuk membatasi mana hak admin dan kasir. Sehingga perlunya memasukan E-Mail dan password saat login untuk memastikan sistem mendeteksi apakah login sebagai admin atau kasir. Namun, perlu juga untuk membatasi saat ada orang lain login namun E-Mail dan password tidak terdaftar ke database maka sistem akan menolak perintah login tersebut.

Gambar 11. Source code hak akses sesuai peran

5. *Authorization*

Tahapan ini memberikan batasan akses ke sistem sesuai dengan perannya masing-masing.

a. Admin

Untuk admin dapat mengolah data produk, data user dan laporan penjualan, baik itu dalam menambahkan, mengedit, dan mengapus data yang tidak diperlukan. Namun, juga dapat melihat log aktivitas yang terjadi pada produk maupun user.

b. Kasir

Pada kasir hanya dapat mengolah data penjualan, baik itu input transaksi penjualan dan dapat melihat atau mencari data barang. Sehingga, selain itu kasir tidak dapat mengakses data-data lainnya.

Gambar 12. Perintah filter akses

6. *Accountability*

Tahapan ini bertujuan untuk memberikan pencatatan aktivitas agar memberikan kemudahan apabila ada kesalahan atau audit. Sehingga, penerapan log aktifitas pada produk yang sudah ditampilkan yang mana

berguna untuk melihat admin melakukan apa dan kapan pada data produk. Selanjutnya, penerapan log aktivitas pada user yang memberikan informasi terkait log in maupun log out baik admin maupun kasir dan melihat kapan dan menggunakan perangkat apa. Dan terakhir melakukan pelaporan penjualan terkait jumlah transaksi yang dilakaukan selama ini dan memberikan informasi terkait keuntungan.

3.3 *Security Development*

Pada tahap ketiga, diterapkan prinsip *Common Software Vulnerabilities and Controls* (kerentanan dan pengendalian perangkat lunak yang umum) untuk memastikan keamanan pada perangkat lunak yang dikembangkan menggunakan framework CodeIgniter 4 (CI4), diperlukan perhatian lebih terhadap aspek keamanan agar sistem dapat berjalan secara optimal. Oleh karena itu, pengkodean yang diterapkan harus mampu melindungi data penting agar tidak dapat diakses oleh pengguna yang tidak berwenang.

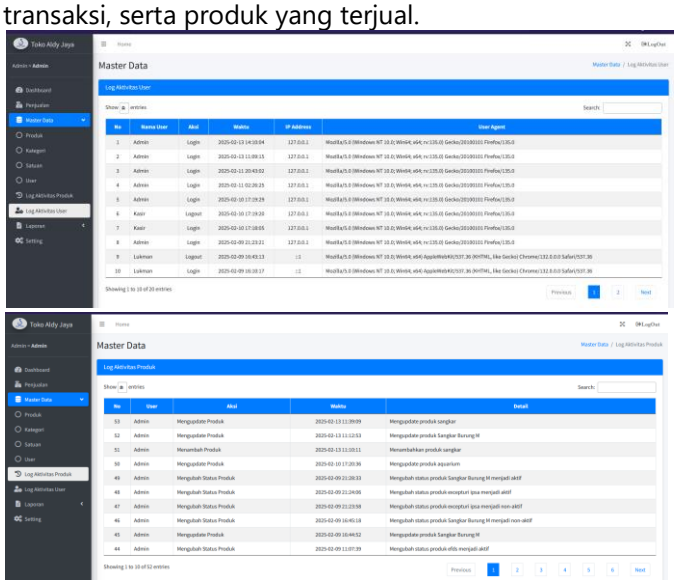
Langkah pertama yang dilakukan adalah menerapkan mekanisme autentikasi saat proses login untuk memisahkan hak akses pengguna sesuai dengan perannya. Proses ini melibatkan validasi kredensial berupa email dan password yang dicocokkan dengan data yang tersimpan dalam basis data. Apabila pengguna terautentikasi sebagai admin, maka sistem akan mengarahkan ke halaman dashboard admin. Sebaliknya, jika pengguna berperan sebagai kasir, maka akan diarahkan ke halaman dashboard penjualan.

Namun, jika pengguna tidak terdaftar dalam basis data atau memasukkan kredensial yang tidak valid, sistem secara otomatis akan mengarahkan kembali ke halaman utama (home).

```
) {
    $email = $this->request->getPost(index: 'email');
    $password = sha1(string: $this->request->getPost(index: 'password'));
    $cek_login = $this->ModelUser->LoginUser(email: $email, password: $password);

    if ($cek_login) {
        session()->set(data: 'id_user', value: $cek_login['id_user']);
        session()->set(data: 'nama_user', value: $cek_login['nama_user']);
        session()->set(data: 'level', value: $cek_login['level']);

        // Simpan log aktivitas login
        $this->ModelLogUser->insertLog(data: [
            'id_user' => $cek_login['id_user'],
            'aksi' => 'login',
            'ip_address' => $this->request->getIPAddress(),
            'user_agent' => $this->request->getUserAgent(),
        ]);

        return redirect()->to(uri: $cek_login['level'] == 1 ? base_url(relativePath: 'Admin') : base_url(relativePath: 'Penjualan'));
    } else {
        session()->setFlashdata(data: 'gagal', value: 'E-Mail Atau Password Salah!!');
        return redirect()->to(uri: base_url(relativePath: 'Home'));
    }
} else {
    session()->setFlashdata(data: 'errors', value: \Config\Services::validation()->getErrors());
    return redirect()->to(uri: base_url(relativePath: 'Home'))->withInput();
}
```

Gambar 13. Perintah aktivitas login

Untuk mendukung keamanan, sistem juga mencatat aktivitas login dan logout agar setiap interaksi pengguna dapat dilacak. Selain itu, aktivitas terkait pengelolaan data produk (tambah, edit, hapus) oleh admin juga dicatat agar pengelolaan data lebih terkontrol. Pada bagian transaksi, digunakan dua tabel terpisah untuk mencatat data penjualan. Ini memungkinkan sistem menghasilkan laporan yang mencantumkan waktu, pelaku transaksi, serta produk yang terjual.



Gambar 14. Menu log aktivitas

3.4 *Security Testing*

Pada tahap keempat, digunakan pendekatan Test Data Management (uji pengelolaan data) untuk memastikan pengolahan data berjalan dengan benar. Proses pengujian dilakukan dengan memanfaatkan data dummy menggunakan library PHP Faker. Penggunaan Faker mempermudah pembuatan data tiruan dalam jumlah besar seperti nama, email, password, dan lainnya, tanpa harus menginput secara manual, sehingga efisien untuk kebutuhan pengujian dan pengembangan.

Melakukan data dummy pada user dengan password dan level random, dalam perintah ini menyisipkan data dummy pada database user, untuk mengecek apakah database sudah berjalan dan tidak mengalami error. Dengan menggunakan data acak baik nama, email, password, dan level. Dengan membuat source code seperti gambar 15 dengan nama file ContactSeeder.php didalam folder seeder, setelah itu jalankan kodingan diterminal menggunakan perintah "php spark db:seed ContactSeeder".

```php
<?php

namespace App\Database\Seeds;

use CodeIgniter\Database\Seeder;

0 references | 0 implementations
class ContactSeeder extends Seeder
{
    1 reference | 0 overrides | prototype
    public function run(): void
    {
        // Inisialisasi Faker
        $faker = \Faker\Factory::create(locale: 'id_ID'); // Lokal Indonesia

        // Array untuk menyimpan data dummy
        $data = [];

        // Membuat 10 data dummy
        for ($i = 0; $i < 10; $i++) {
            // Password acak
            $plainPassword = $faker->password;

            $data[] = [
                'nama_user' => $faker->name,
                'email' => $faker->unique()->email,
                'password' => sha1(string: $plainPassword), // Hash password acak
                'level' => $faker->randomElement(array: [1, 2]), // 1 untuk Admin, 2 untuk Kasir
            ];

            // (Optional) Tampilkan password asli ke console/log untuk referensi
            echo "Generated Password for User {$i}: {$plainPassword}" . PHP_EOL;
        }

        // Insert data ke tabel tbl_user
        $this->db->table(tableName: 'tbl_user')->insertBatch(set: $data);
    }
}
```

Gambar 15. Source code data dummy pada user

Selain itu, dilakukan pengujian black box untuk memastikan fungsi sistem berjalan sebagaimana mestinya tanpa melihat struktur kode. Pengujian ini mencakup proses login, input data produk, dan transaksi penjualan, dengan fokus pada hasil keluaran dari sistem sesuai dengan input yang diberikan.

Tabel 2. Uji black box pada login

No	Skenario Uji	Input	Ekspektasi Hasil	Hasil Aktual	Status
1	Login dengan akun admin yang valid	admin@gmail.com/ password123	Berhasil masuk ke dashboard admin	☑Sesuai	Lulus
2	Login dengan akun kasir yang valid	kasir@gmail.com / kasir123	Berhasil masuk ke dashboard kasir	☑Sesuai	Lulus
3	Login dengan username atau password salah	admin@gmail.com / salah123	Menampilkan pesan "Email atau password salah"	☑Sesuai	Lulus
4	Login tanpa mengisi username & password	-	Menampilkan pesan "Email dan password tidak boleh kosong"	☑Sesuai	Lulus

3.5 *Security Deployment*

Pada tahapan ini, digunakan *Installation* (instalasi) yang mana penting untuk memilih server yang mendukung protokol HTTPS dan SSL demi menjaga keamanan koneksi antara pengguna dan aplikasi. Untuk itu, digunakan layanan dari Niagahoster sebagai penyedia domain dan hosting agar aplikasi dapat diakses secara online kapan pun dibutuhkan.

Langkah instalasi dimulai dengan pembelian hosting sesuai kebutuhan dan anggaran, kemudian dilanjutkan dengan memilih domain yang mencerminkan jenis website, seperti domain .shop untuk situs penjualan. Setelah itu, aplikasi diunggah ke server dengan memisahkan folder public dari folder internal demi melindungi file sensitif dari akses langsung. Instalasi SSL juga dilakukan untuk menjamin koneksi aman.

Setelah aplikasi berhasil dihosting, dilakukan pengujian keamanan menggunakan OWASP ZAP untuk mendeteksi kerentanan sistem. Pengujian lanjutan juga mencakup simulasi serangan SQL Injection guna memastikan sistem tidak rentan terhadap manipulasi data melalui input berbahaya.

Tabel 3. Query SQL Injection

Baypas Query SQL Injection
‘ or”=’
‘or TRUE--
‘or 1=1--
‘or ‘a’=’--
‘or 1=1#
Admin’--
Admin’or’1’=’#
Admin’or ‘1’=’1
Admin’or 1=1 or “=”
Admin’ or ‘1

3.6 *Security Maintenance*

Tahap akhir dalam Pemeliharaan keamanan adalah *Problem Management* (manajemen masalah), yaitu mengidentifikasi dan menangani masalah yang muncul agar tidak terulang kembali di masa depan. Salah satu isu yang ditemukan adalah duplikasi nomor faktur saat dua kasir melakukan transaksi secara bersamaan. Hal ini terjadi karena sistem tidak memperbarui nomor faktur secara otomatis dalam kondisi transaksi paralel.

Tabel 4. Permasalah pada website kasir

No	Masalah	Deskripsi	Status	Catatan Perbaikan
1	Masalah login	Pengguna tidak bisa login meskipun kredensial benar	Aman	Memverifikasi hash password dan pengelolaan sesi login
2	Nomor faktur duplikat	Nomor faktur sama untuk transaksi berbeda	Aman	Menggunakan query FOR UPDATE yang diterapkan pada database
3	Stok produk tidak sinkron	Stok tidak sesuai antara fisik dan database	Aman	Menerapkan Trigger pada database
4	Hak akses tidak berfungsi	Kasir dapat mengakses fitur admin	Aman	Middleware akses diterapkan di semua endpoint
5	Laporan tidak dapat di print	Data laporan tidak terbaca pada sistem	Aman	Perbaikan pada controller laporan
6	Sistem tidak mencatat log aktivitas	Tidak ada log aktifitas saat login	Aman	Sistem login sedang dalam tahap implementasi

Berdasarkan daftar laporan permasalahan yang telah diidentifikasi, salah satu kendala yang ditemukan terkait dengan nomor faktur saat terjadi transaksi bersamaan oleh dua pengguna dengan peran sebagai kasir. Permasalahan ini muncul ketika nomor faktur tidak berubah secara otomatis pada transaksi yang dilakukan secara paralel. Sebagai contoh, jika kasir 1 melakukan transaksi dan mendapatkan nomor faktur 20250121001, maka setelah transaksi selesai, nomor faktur seharusnya berlanjut menjadi 20250121002. Namun, saat kasir 2 melakukan transaksi secara bersamaan, sistem tidak melanjutkan nomor faktur terakhir yang diproses oleh kasir 1, sehingga berpotensi terjadi duplikasi nomor faktur.

Untuk mengatasi permasalahan ini, dilakukan perbaikan pada query nomor faktur dengan menambahkan klausa FOR UPDATE. Penambahan ini bertujuan untuk mengunci baris data yang sedang diproses hingga transaksi selesai, sehingga nomor faktur dapat berlanjut secara berurutan tanpa terjadi duplikasi. Sebagai ilustrasi, pada Gambar 4.36, kasir 1 telah melakukan transaksi sebanyak empat kali, dan ketika kasir 1 logout, lalu kasir 2 login, nomor faktur akan melanjutkan dari nomor terakhir yang digunakan oleh kasir 1. Dengan implementasi ini, integritas data nomor faktur dapat terjaga, dan proses transaksi berjalan lebih aman serta konsisten.

```
public function NoFaktur(): string
{
    $tgl = date(format: 'Ymd');
    $query = $this->db->query(sql: "SELECT MAX(RIGHT(no_faktur,4)) as no_urut from tbl_jual where DATE(tgl_jual)='$tgl' FOR UPDATE");
    $hasil = $query->getRowArray();
    if ($hasil['no_urut'] > 0) {
        $tmp = $hasil['no_urut'] + 1;
        $kd = sprintf(format: '%04s', values: $tmp);
    } else {
        $kd = '0001';
    }
    $no_faktur = date(format: 'Ymd') . $kd;
    return $no_faktur;
}
```

Gambar 16. Source code no faktur pada model

Dari hasil yang diperoleh selama pengembangan sistem, penggunaan perintah FOR UPDATE terbukti berperan penting dalam menjaga agar nomor faktur tetap unik dan tersusun secara berurutan. Perintah ini bekerja dengan cara mengunci baris data tertentu saat proses transaksi sedang berlangsung, sehingga mencegah gangguan dari proses lain yang mencoba mengakses data yang sama secara bersamaan. Dengan begitu, risiko terjadinya duplikasi nomor faktur dapat diminimalkan, terutama dalam kondisi ketika banyak transaksi dilakukan dalam waktu bersamaan. Penerapan mekanisme ini menjadi salah satu langkah penting dalam menjaga keandalan dan konsistensi data pada sistem aplikasi kasir yang dikembangkan.

4. KESIMPULAN

Metode *Secure Software Development Life Cycle* (S-SDLC) yang diterapkan dalam perancangan aplikasi kasir berbasis web di Toko Aldy Jaya menunjukkan hasil yang signifikan dalam meningkatkan keamanan dan efisiensi operasional. Pada tahap *Security Requirements*, klasifikasi data berhasil mengidentifikasi informasi sensitif yang perlu dilindungi, sehingga akses terhadap data penting seperti laporan penjualan dan data pengguna dapat dibatasi hanya untuk pihak berwenang. Dalam tahap *Security Design*, penerapan prinsip-prinsip keamanan inti, seperti kerahasiaan dan integritas, memastikan bahwa data tetap terlindungi dari akses yang tidak sah. Hasil di tahap *Security Development* menunjukkan bahwa kode yang dikembangkan telah menerapkan kontrol keamanan yang efektif untuk meminimalkan risiko kerentanan. Melalui *Security Testing*, aplikasi berhasil diverifikasi dan divalidasi, membuktikan bahwa sistem dapat menangani data dengan benar tanpa kesalahan. Pada tahap *Security Deployment*, aplikasi berhasil diterapkan secara aman di lingkungan produksi, dan dalam tahap *Security Maintenance*, sistem terus dipantau dan diperbaiki untuk mempertahankan kinerjanya. Dengan demikian, penelitian ini membuktikan bahwa penerapan metode S-SDLC tidak hanya meningkatkan keamanan data, tetapi juga memberikan kemudahan dalam pengelolaan transaksi dan pemantauan stok barang secara real-time, memenuhi kebutuhan Toko Aldy Jaya dengan lebih baik.

Beberapa saran yang bisa digunakan untuk pengembangan selanjutnya yaitu disarankan agar penelitian selanjutnya menerapkan lebih dari satu aspek keamanan pada setiap tahapan, atau mengimplementasikan seluruh aspek keamanan dalam setiap tahapan metode S-SDLC, guna meningkatkan ketahanan sistem.

5. REFERENSI

CyberSecurity Malaysia. (2020). *Guidelines for Secure Software Development Life Cycle (SSDLC)* (Edisi 1). Cyberjaya: CyberSecurity Malaysia.

Hasan, M. R., Suhermanto, S., & Suharmanto, S. (2021). Keamanan sistem perangkat lunak dengan Secure Software Development Lifecycle. *Jurnal Ilmu Komputer dan Bisnis (JIKB)*, *12*(1), 88–101.

Naufal, M. D. D., & Nalurita, S. (2023). Pengaruh promosi dan kemudahan penggunaan aplikasi terhadap keputusan pembelian Shopeefood pada mahasiswa Universitas Dirgantara Marsekal Suryadarma Jakarta. *Jurnal Ilmiah M-Progress*, *13*(1), 23–32.

Nuryamin, Y., & Risyda, F. (2021). Perancangan aplikasi kasir pada kedai kopi berbasis web menggunakan model Waterfall. *Jurnal Informatika Universitas Pamulang*, *6*(2), 191–197.

Pangestu, S. D., & Astutik, I. R. I. (2024). Rancangan aplikasi kasir toko kelontong berbasis website menggunakan metode Waterfall. JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika), 9(1), 125–135.

Purnama, W. C., Annas, F., Musril, H. A., & Darmawati, G. (2023). Perancangan media pembelajaran PAI berbasis Android menggunakan Kodular kelas X di SMA N 1 IV Koto. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *7*(2), 1304–1311.

Wibowo, M. H., & Ulum, F. (2023). Sistem informasi koperasi simpan pinjam berbasis website pada PRIMKOPPABRI Bandar Lampung. *Jurnal Teknologi dan Sistem Informasi*, *4*(1), 22–27.