

Perancangan Aplikasi Web Manajemen Data *Invoice* Untuk SHILAU Polibatam

Steven Chris Hadian¹, Arta Uly Siahaan²

^{1,2} Teknologi Rekayasa Multimedia, Politeknik Negeri Batam

Info Artikel

Riwayat Artikel:

Diterima ..., 202x

Direvisi ..., 202x

Diterbitkan ..., 202x

Kata kunci:

Invoice

Aplikasi web

Manajemen data

Personal Scrum

User Acceptance Testing

ABSTRAK

SHILAU (Satuan Hilirisasi Inovasi dan Layanan Usaha) merupakan departemen di Politeknik Negeri Batam yang menangani transaksi penjualan layanan dan produk kepada klien eksternal dengan *invoice* sebagai bukti transaksi. Selama ini, pembuatan dan pencatatan *invoice* masih dilakukan secara manual menggunakan Microsoft Word dan Excel, sehingga rentan terjadi kehilangan dokumen dan kesulitan pencarian dokumen spesifik. Penelitian ini bertujuan merancang dan membangun aplikasi web manajemen data *invoice* untuk staf SHILAU menggunakan metode *Personal Scrum*. Aplikasi dikembangkan menggunakan framework Laravel dan MySQL sebagai *database*. Pengujian sistem dilakukan menggunakan metode *Black Box Testing* dengan skenario uji dan *User Acceptance Testing* (UAT) yang melibatkan staf SHILAU sebagai pengguna akhir. Hasil pengujian *Black Box Testing* menunjukkan tingkat keberhasilan sebesar 100%, yang mengindikasikan seluruh fungsi utama aplikasi berjalan sesuai harapan. Hasil UAT menunjukkan bahwa pengguna akhir menilai aplikasi mudah dipelajari dan digunakan serta tidak ditemukan kendala berupa *bug* atau *error* selama penggunaan. Namun, terdapat catatan perbaikan berupa kebutuhan penambahan fitur tanda tangan dan stempel digital pada dokumen PDF yang dihasilkan. Hasil penelitian menunjukkan bahwa aplikasi ini dapat diterima dan diimplementasikan untuk mendukung proses manajemen data *invoice* di lingkungan kerja SHILAU.

Penulis Korespondensi:

Steven Chris Hadian¹,

Teknologi Rekayasa Multimedia,

Politeknik Negeri Batam,

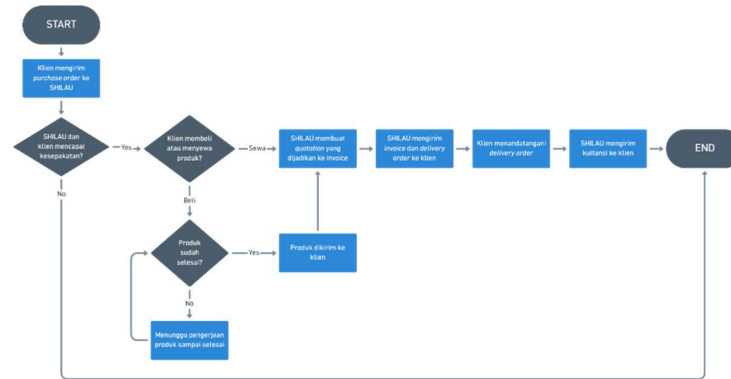
Jl. Ahmad Yani, Tlk.Tering, Kec. Batam Kota, Kota Batam, Kepulauan Riau.

Email: steven.chris03@gmail.com

1. PENDAHULUAN

SHILAU (Satuan Hilirisasi Inovasi dan Layanan Usaha) merupakan salah satu departemen di Politeknik Negeri Batam yang berfokus pada menangani proses hilirisasi inovasi dan layanan usaha jasa yang dihasilkan dari kegiatan pembelajaran yang berada dalam ranah kompetensi Politeknik Negeri Batam. SHILAU menyediakan beragam layanan, mulai dari penyewaan alat dan fasilitas (ruangan dan lahan) untuk berbagai acara, penjualan produk manufaktur dari kampus, penyediaan paket training dan sertifikasi kompetensi serta memfasilitasi kerja sama antara klien eksternal dengan mahasiswa melalui PBL (*Project Based Learning*) [1].

Dalam kegiatan operasional harian, SHILAU memiliki kebutuhan untuk mengelola dokumen transaksi yaitu *invoice*. Saat ini, seluruh proses pembuatan dan pencatatan dokumen-dokumen tersebut masih dilakukan secara manual oleh staf, yaitu dengan membuat masing-masing dokumen menggunakan Microsoft Word, kemudian mengunggahnya ke layanan *cloud* seperti OneDrive. Setelah diunggah, tautan dari setiap dokumen disalin dan ditempel secara manual ke dalam lembar kerja Microsoft Excel, yang juga digunakan untuk mencatat data transaksi seperti tanggal, nomor dokumen, nama customer atau mitra, nama produk atau training, harga, status pembayaran, dan tanggal transaksi. Menurut pengalaman penulis, langkah-langkah tersebut kurang efektif untuk manajemen data *invoice*. Berdasarkan permasalahan yang dinyatakan sebelumnya, pengembangan aplikasi web manajemen data *invoice* adalah solusi yang diharapkan penulis untuk menyelesaikan masalah tersebut.



Gambar 1. Alur pembuatan *invoice* di SHILAU

Seperti yang terlihat pada Gambar 1, proses pembuatan *invoice* di SHILAU dimulai ketika klien mengirimkan *purchase order* kepada SHILAU. Selanjutnya, SHILAU dan klien perlu mencapai kesepakatan terkait produk atau layanan yang akan ditransaksikan; apabila kesepakatan tidak tercapai, proses akan dihentikan. Jika kesepakatan tercapai, ditentukan apakah klien membeli atau menyewa produk. Untuk transaksi sewa, SHILAU langsung membuat *quotation* yang kemudian dijadikan dasar pembuatan *invoice*. Sementara untuk transaksi beli, SHILAU perlu menunggu proses pengerjaan produk hingga selesai sebelum produk dikirimkan kepada klien, barulah *quotation* dan *invoice* dapat dibuat. Setelah *invoice* dan *delivery order* dikirimkan kepada klien, klien menandatangani *delivery order* sebagai bukti penerimaan, dan SHILAU mengirimkan kuitansi kepada klien sebagai bukti pembayaran, yang menandai selesainya proses transaksi.

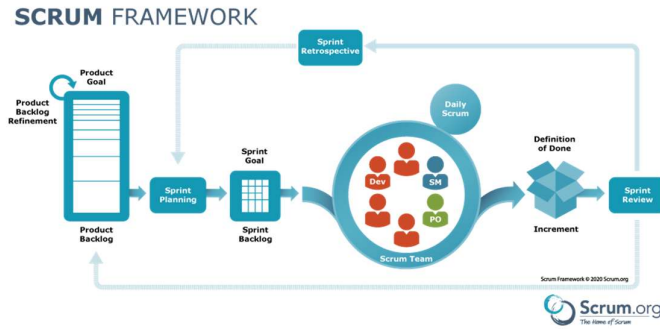
Salah satu keunggulan dari aplikasi web adalah aksesibilitasnya. Para pengguna dapat mengakses aplikasi web melalui *browser* selama terhubung ke internet tanpa melalui proses instalasi ke komputer [2]. Dengan demikian, para staf SHILAU memiliki fleksibilitas lebih dalam hal waktu dan tempat untuk mengakses dan memanajemen data *invoice*. Selain itu, sistem berbasis web juga mendukung kolaborasi secara *real-time* antar pengguna sehingga setiap staf dapat bekerja secara sinkron, mengurangi risiko duplikasi data, serta mempercepat proses manajemen dan pembuatan data *invoice* [3].

Untuk mengembangkan aplikasi web, pemilihan teknologi yang tepat dibutuhkan untuk memastikan sistem web berjalan dengan semestinya [4]. *Framework* Laravel dipilih sebagai basis pengembangan karena memiliki berbagai keunggulan, seperti fleksibilitas, performa yang tinggi, serta dukungan komunitas yang luas. *Framework* ini mempermudah proses pembangunan *user interface* (UI) yang interaktif, responsif, dan ramah pengguna. Selain itu, ekosistem Laravel yang kaya memungkinkan integrasi dengan berbagai *library* dan layanan tambahan yang dapat mendukung kebutuhan sistem, seperti autentikasi, manajemen basis data, hingga pengolahan dokumen [5].

SHILAU sebagai salah satu departemen Politeknik Negeri Batam yang memiliki tanggung jawab dalam penjualan berbagai layanan terus melakukan transaksi dengan klien-klien eksternal. Setiap proses transaksi tersebut didokumentasikan dalam bentuk *invoice* sebagai bukti transaksi yang dilakukan. Dengan menggunakan aplikasi web, diharapkan para staf SHILAU dapat membuat dan memanajemen data *invoice* tanpa menggunakan metode manual seperti pencatatan melalui Microsoft Word dan Microsoft Excel yang selama ini dilakukan. Metode tersebut seringkali menimbulkan kendala seperti sulitnya pencarian data, risiko duplikasi maupun kehilangan dokumen. Aplikasi web akan memberikan solusi berupa kemudahan akses, pengolahan data dengan rapi, serta peningkatan kecepatan dalam pengelolaan *invoice*.

2. METODOLOGI PENELITIAN

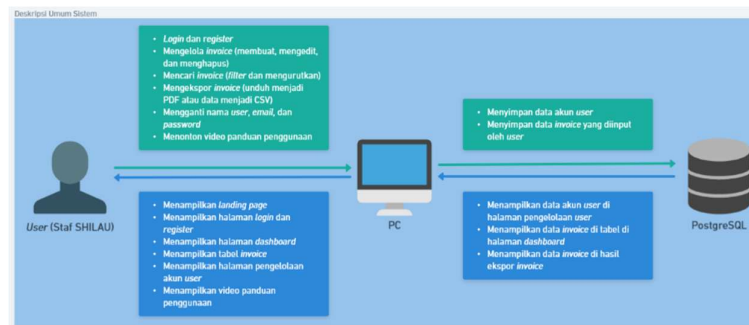
Di penelitian ini, metode pengembangan yang digunakan adalah *Scrum*. Meskipun *Scrum* biasanya diterapkan pada tim, dalam penelitian ini, metode *Scrum* diadaptasi untuk pengembangan individu yang bernama *Personal Scrum*. Penulis berperan sebagai *Product Owner*, *Scrum Master*, dan *Developer* sekaligus. Untuk mencapai penyelesaian proyek yang efektif dan bertahap, *Personal Scrum* berguna bagi individu yang bekerja secara mandiri, dengan kolaborasi yang hanya dibatasi untuk pihak-pihak penting seperti klien dan pengguna akhir [6]. Alur pengerjaan menggunakan *Scrum Framework* terdapat pada Gambar 2.



Gambar 2. Scrum Framework
(Sumber: Scrum.org) [7]

2.1 Deskripsi Umum Sistem

Aplikasi web yang dikembangkan merupakan web yang khusus diakses untuk staf SHILAU. *User* dapat membuat akun dan *login* ke akun tersebut untuk mengakses ke halaman *dashboard* yang merupakan halaman utama untuk manajemen data *invoice*. Jikalau *user* lupa *password*, *user* bisa memulihkan akunnya dengan *link* yang akan *dikirimkan* ke *email user* untuk mengganti *password*. Di halaman tersebut, *user* bisa membuat *invoice* beserta jenis dari *invoice* tersebut, menambah dan mengedit data yang ada di *invoice* tersebut, menghapus *invoice* yang telah dibuat, mencari *invoice* spesifik melalui *search bar*, serta mengeksport *invoice* menjadi dokumen PDF (*Portable Document Format*) atau datanya saja menjadi CSV (*Comma Separated Values*). Di halaman profil, *user* bisa mengganti nama profil, *email*, serta *password*. Gambar 3 menunjukkan diagram deskripsi umum sistem pada aplikasi web tersebut.



Gambar 3. Gambaran Umum Sistem

2.2 Desain Sistem

Untuk mengembangkan suatu aplikasi web, dibutuhkan perancangan *Activity Diagram*, *Entity Relationship Diagram* (ERD), dan tabel dari kode-kode kebutuhan fungsional dan non-fungsional beserta fungsi-fungsi dalam sistem aplikasi web tersebut. Kebutuhan fungsional merupakan fitur apa saja yang akan digunakan oleh *user* di aplikasi web tersebut. Tabel kode fungsional sistem berfungsi untuk mendokumentasikan fitur-fitur aplikasi beserta kode identifikasinya [8]. Tabel kode kebutuhan fungsional dari aplikasi web manajemen data *invoice* terdapat pada Tabel 1.

Tabel 1. Tabel Kode Kebutuhan Fungsional Sistem

NO	Kode Fungsional	Kebutuhan Fungsional
1	FRW01	User bisa login ke web
2	FRW02	User bisa memulihkan akun jika lupa password
3	FRW03	User bisa membuat invoice
4	FRW04	User bisa membuka invoice
5	FRW05	User bisa menghapus invoice
6	FRW06	User bisa mencari invoice yang spesifik

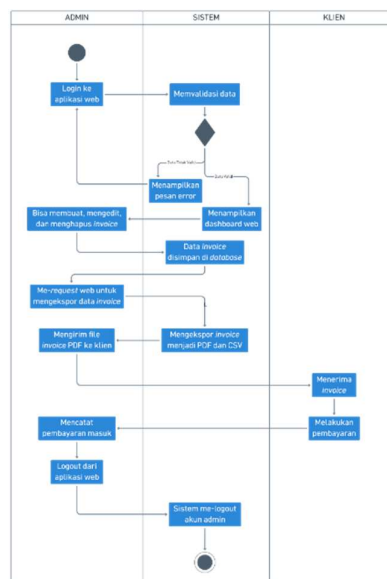
7	FRW07	User bisa menginput data ke <i>invoice</i>
8	FRW08	User bisa mengedit data yang sudah ada di <i>invoice</i>
9	FRW09	User bisa mengekspor <i>invoice</i> menjadi PDF
10	FRW10	User bisa mengekspor data <i>invoice</i> menjadi CSV
11	FRW11	User bisa mengurutkan tabel <i>invoice</i> secara menaik atau menurun
12	FRW12	User bisa mengganti nama profil akun
13	FRW13	User bisa mengganti <i>email</i> akun
14	FRW14	User bisa mengganti <i>password</i> akun

Kebutuhan non-fungsional berfokus pada nilai-nilai seperti performa, keandalan, keamanan, dan kemudahan pemakaian yang menjamin aplikasi web dapat digunakan secara optimal [6]. Tabel kode kebutuhan non-fungsional aplikasi web dapat dilihat pada Tabel 2.

Tabel 2. Tabel Kode Kebutuhan Non-fungsional Sistem

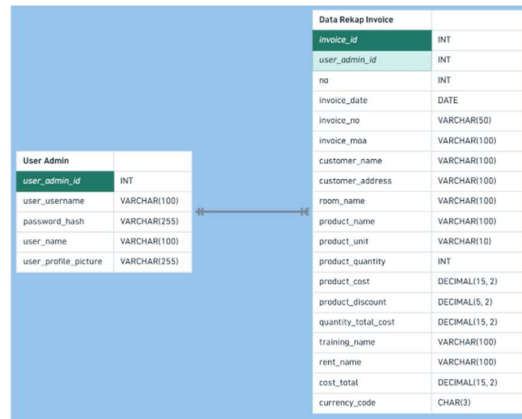
NO	Kode Non-fungsional	Kebutuhan Fungsional
1	NFRW01	User bisa mengakses aplikasi web melalui <i>browser</i> web yang populer, seperti Google, Firefox, dan Edge
2	NFRW02	Aplikasi web hanya bisa diakses oleh <i>user</i> yang sudah terdaftar dan terautentikasi
3	NFRW03	Aplikasi web mengenkripsi <i>password</i> milik <i>user</i> menjadi <i>hash</i> dengan metode <i>hashing</i>
4	NFRW04	User bisa memulihkan akunnya jika lupa <i>password</i> untuk <i>login</i> ke aplikasi web
5	NFRW05	Tampilan dari aplikasi web responsif untuk berbagai ukuran layar, seperti layar laptop, <i>mobile</i> , dan lain-lain
6	NFRW06	Tampilan antarmuka aplikasi web mudah untuk dinavigasi (<i>user friendly</i>)
7	NFRW07	Aplikasi web mem- <i>backup</i> data-data yang ada di <i>database</i> secara berkala

Activity Diagram adalah diagram yang menggambarkan alur kerja dalam suatu proses sistem, termasuk interaksi antara pengguna dan sistem melalui setiap tahapan fungsinya. Diagram ini membantu memvisualisasikan langkah demi langkah penggunaan sistem secara berurutan, termasuk *decision node* dan aliran data antar aktor yang terlibat [9]. *Activity Diagram* dari aplikasi web manajemen data *invoice* ditunjukkan pada Gambar 4.



Gambar 4. Activity Diagram

Entity Relationship Diagram (ERD) adalah diagram yang memodelkan struktur data dengan menunjukkan entitas, atribut, serta hubungan antar entitas dalam suatu sistem basis data disertai dengan kardinalitas antar entitas data [10]. Model ERD ini bernama *Physical Model*, yaitu penggambaran basis data yang siap diimplementasikan di *Database Management System* (DBMS). Model ini menampilkan nama tabel, atribut lengkap beserta tipe datanya, ukuran *field*, serta hubungan antar tabel melalui *primary key* (hijau tua) dan *foreign key* (hijau muda) [11]. *Entity Relationship Diagram* untuk aplikasi web manajemen data *invoice* ditunjukkan pada Gambar 5.



Gambar 5. *Entity Relationship Diagram Physical Model*

2.3 Product Backlog

Product Backlog merupakan daftar prioritas seluruh kebutuhan dan fitur yang harus dikembangkan dalam aplikasi web berdasarkan visi pribadi pengembang sebagai *Product Owner* sekaligus *Developer*. Daftar ini berisi fungsi-fungsi yang akan membentuk sistem secara utuh, disusun berdasarkan tingkat kepentingan dan urutan pengerjaan [6]. Tabel penjabaran deskripsi *Product Backlog* terdapat pada Tabel 3.

Tabel 3. Tabel *Product Backlog*

NO	Deskripsi <i>Backlog</i>	Prioritas
1	Menyiapkan proyek dengan framework Laravel dan koneksi ke database MySQL	High
2	Membuat sistem autentikasi untuk <i>login</i> serta sistem pemulihan akun jika lupa <i>password</i>	High
3	Membuat halaman <i>login</i> , <i>dashboard</i> , dan <i>profil</i>	Medium
4	Membuat sistem CRUD untuk akun <i>user</i> aplikasi web (mengganti nama akun, <i>email</i> , dan <i>password</i>)	High
5	Membuat tabel untuk data <i>invoice</i> dengan sistem <i>pagination</i> dan pengurutan	High
6	Membuat sistem CRUD untuk data <i>invoice</i>	High
7	Membuat sistem pencarian <i>invoice</i> dengan <i>filter</i>	High
8	Membuat fitur mengunduh <i>invoice</i> menjadi PDF dan mengekspor data <i>invoice</i> menjadi format CSV	High
9	Membuat fitur untuk mem- <i>back up</i> data-data <i>invoice</i> secara berkala	High
10	Menguji fungsi-fungsi pada aplikasi web dengan metode <i>black box testing</i>	High
11	Mengoptimalkan tampilan antarmuka aplikasi web	Medium

2.4 Sprint Planning

Sprint Planning merupakan tahap perencanaan awal sebelum *sprint* dimulai, di mana pengembang menentukan *sprint goal*, yaitu memilih item dari *Product Backlog* yang akan dikerjakan serta memperkirakan waktu dan langkah-langkah pengerjaannya. Tujuan dari *Sprint Planning* adalah memastikan pekerjaan selama satu periode *sprint* berjalan terarah dan sesuai prioritas pengembangan aplikasi web [6].

2.5 Sprint Backlog

Sprint Backlog adalah daftar pekerjaan yang dikerjakan dalam satu *sprint* yang memiliki batas maksimal satu bulan [12]. Jangka waktu *sprint* yang ditentukan pada pengembangan ini adalah setiap satu *sprint* dilaksanakan setiap dua minggu. Pada pengembangan ini, *Sprint Backlog* berisi rancangan aktivitas

pengembangan aplikasi web manajemen data *invoice* yang dibagi ke dalam lima *sprint* yang dijabarkan pada Tabel 4.

Tabel 4. *Sprint Backlog*

No	Backlog	Priority	Sprint
1	Setup environment (Laravel, MySQL)	High	1
2	Pembuatan halaman <i>login</i> dan lupa password	High	1
3	Integrasi autentikasi (<i>login/logout</i> , manajemen <i>session</i>)	Medium	2
4	Pembuatan halaman <i>dashboard</i> awal	Medium	2
5	Pembuatan halaman profil	Low	2
6	Fitur mengedit nama, <i>email</i> , dan <i>password</i> profil	Low	2
7	Implementasi CRUD data <i>invoice</i>	High	3
8	Tampilan daftar <i>invoice</i> dalam tabel	High	3
9	Fitur pencarian <i>invoice</i>	High	4
10	Ekspor data <i>invoice</i> ke PDF dan CSV	High	4
11	Fungsi <i>back up</i> data-data <i>invoice</i>	High	5
12	Pengujian keseluruhan aplikasi	High	5

3. HASIL DAN ANALISIS

Berikut adalah hasil perancangan dan analisis dari pengembangan aplikasi web manajemen data *invoice*, dikembangkan menggunakan metode *Personal Scrum* dan dianalisis secara kualitatif menggunakan metode *User Acceptance Testing* (UAT).

3.1 Tahapan-Tahapan Sprint

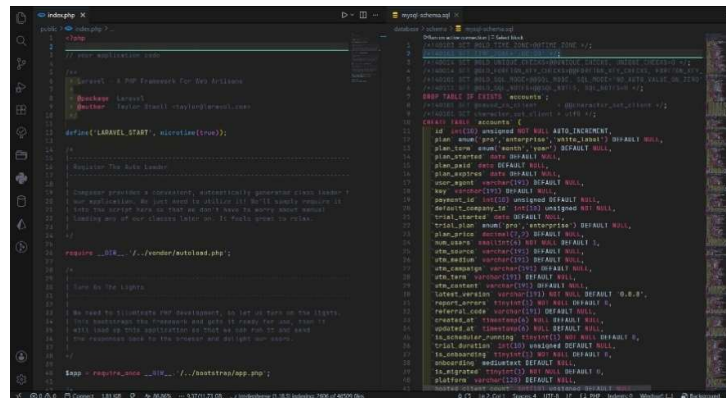
Metode pengembangan *Personal Scrum* menggunakan *sprint* untuk mengembangkan fitur-fitur aplikasi web berdasarkan tingkat prioritasnya sesuai dengan yang sudah direncanakan pada tahapan *Sprint Backlog*. Terdapat enam *sprint* selama proses pengembangan aplikasi web ini.

3.1.1 Sprint 1

Pada *sprint* pertama ini, terdapat dua *backlog* yang dikembangkan, yaitu *men-setup environment* yang terdiri dari *framework* Laravel dan MySQL sebagai *database* untuk memulai pengembangan aplikasi web dan mengembangkan halaman *login* serta lupa *password*.

a) Setup environment (Laravel, MySQL)

Tahap pengembangan dimulai dengan *men-setup environment* dengan menggunakan *framework* Laravel dan MySQL untuk database aplikasi web ini. Sebelum *men-setup* Laravel, bahasa pemrograman PHP harus di-*install* ke komputer karena bahasa pemrograman yang digunakan *framework* Laravel adalah PHP. MySQL juga di-*install* ke komputer untuk menjalankan bahasa pemrograman SQL untuk membuat *database*. Untuk memudahkan *setup*, *Windows Subsystem for Linux* (WSL) digunakan untuk meng-*install* PHP dan MySQL dengan menjalankan *command* tertentu. Tampilan dari *file* Laravel dan MySQL ditampilkan pada Gambar 6.



Gambar 6. File Laravel dan MySQL

b) Pembuatan halaman *login* dan lupa *password*

Halaman *login* dikembangkan sebagai halaman awal untuk proses autentikasi pengguna untuk mengakses fitur-fitur aplikasi web ini. Halaman lupa *password* untuk pengguna yang ingin memulihkan akses akun untuk *login* kembali. *Link* untuk *reset password* akan dikirimkan ke *email* pengguna dan pengguna bisa me-*reset password*-nya melalui *link* tersebut.

3.1.2 Sprint 2

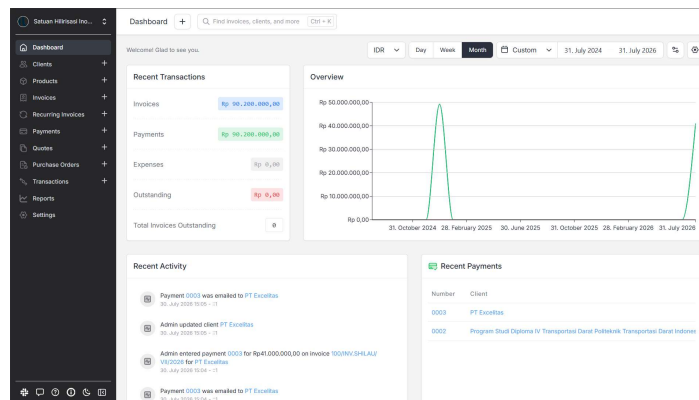
Di *sprint* kedua, *backlog* yang dikembangkan adalah integrasi autentikasi, halaman *dashboard*, *profil*, dan fitur mengedit nama, *email*, dan *password* profil.

a) Integrasi autentikasi

Mengintegrasikan proses autentikasi ke dalam aplikasi web supaya saat pengguna *login* ke dalam web ini, akun tersebut mendapatkan *session* selama masuk ke dalam web dan ketika pengguna *logout*, *session* tersebut akan dihapus dari web sehingga pengguna harus melakukan *login* ulang ke web tersebut.

b) Pembuatan halaman *dashboard* dan *profil*

Halaman *dashboard* merupakan halaman pertama yang ditampilkan ketika pengguna *login* ke web ini. Halaman *dashboard* ini menampilkan informasi-informasi mengenai *invoice* yang telah dibuat dan terdapat *sidebar* untuk menavigasi setiap halaman yang ada di web ini. Halaman profil menampilkan informasi dan pengaturan untuk mengubah berbagai informasi tentang akun pengguna. Tampilan halaman *dashboard* ditampilkan pada Gambar 7.



Gambar 7. Halaman *dashboard*

c) Fitur mengedit nama, *email*, dan *password* akun

Setelah melakukan *login*, pengguna bisa mengakses halaman profil untuk mengedit nama, *email*, dan *password* dari akun pengguna untuk memperbarui informasi sesuai dengan informasi pengguna yang menggunakan akun tersebut.

3.1.3 Sprint 3

Tahap *sprint* ketiga difokuskan pada *database* untuk proses manajemen *invoice*, yaitu dengan mengimplementasikan sistem CRUD ke data *invoice* dan halaman *invoice* bisa menampilkan tabel untuk melihat dan mengatur *invoice-invoice* yang telah dibuat.

a) Implementasi sistem CRUD ke *database invoice*

Sistem CRUD diimplementasikan ke *database invoice* supaya pengguna bisa membuat, mengedit, dan menghapus *invoice*. Di dalam *invoice* tersebut juga terdapat data-data yang bisa ditambah, diedit, dan dihapus juga, seperti data produk, *customer*, dan lain-lain.

b) Tampilan daftar *invoice* dalam tabel

Data-data *invoice* yang telah ditambahkan sebelumnya akan ditampilkan dalam bentuk tabel yang bisa diurutkan sesuai data-data di kolom-kolom tabel. Di tabel tersebut juga terdapat *button actions* untuk menampilkan fitur mengedit dan meng-*export invoice* tersebut menjadi PDF. Tampilan dari tabel tersebut ditampilkan pada Gambar 8.

Invoice ID	Number	Client	Amount	Status	Date	Actions
0005	PT. SHILAU	Rp 8.000.000,00	Rp 0,00	30 July 2024	06 August 2024	Actions
0006	PT. SHILAU	Rp 9.000.000,00	Rp 0,00	30 July 2024	06 August 2024	Actions
0007	PT. SHILAU	Rp 40.000.000,00	Rp 0,00	10 December 2024		Actions

Gambar 8. Tabel data *invoice*

3.1.4 Sprint 4

Di tahap *sprint* keempat, fitur untuk mencari *invoice* melalui *search bar* yang terdapat pada *top navigation bar* serta meng-*export invoice* menjadi PDF dan CSV dikembangkan dan ditambahkan ke dalam web ini.

a) Fitur pencarian *invoice*

Di bagian *top navigation bar* pada web ini, terdapat *search bar* yang bisa digunakan oleh pengguna untuk melakukan pencarian terhadap suatu data *invoice* yang spesifik.

b) Fitur *export invoice* menjadi PDF dan CSV

Pengguna bisa meng-*export* data-data *invoice* yang telah dibuat menjadi *file* berformat PDF yang bisa diunduh ke komputer pengguna dan juga bisa meng-*export* data *invoice* sebagai CSV sehingga bisa dibuka dalam *software* pengolah data, seperti Microsoft Excel.

3.1.5 Sprint 5

Sprint kelima adalah *sprint* terakhir untuk pengembangan fitur *backup* dari data-data *invoice* serta melakukan pengujian awal keseluruhan fitur-fitur dan fungsi-fungsi dari aplikasi web ini.

a) Fungsi *back up* data-data *invoice*

Setiap data *invoice* yang telah dibuat bisa di-*backup* melalui halaman pengaturan. *File backup* tersebut ber-format zip sehingga bisa di-*upload* ke layanan *cloud* atau disimpan di dalam media penyimpanan data lainnya.

b) Pengujian aplikasi web

Sebelum diuji dan digunakan oleh staf di SHILAU, pengembang melakukan pengujian fitur-fitur aplikasi web secara keseluruhan untuk memastikan semua fitur-fitur yang ada pada aplikasi web ini berfungsi dengan semestinya.

3.2 Black Box Testing dan User Acceptance Testing (UAT)

Black Box Testing berfungsi untuk mengetahui fitur-fitur mana saja yang berfungsi dengan sesuai atau tidak sesuai menurut *input* yang diberikan oleh pengguna dan *output* yang diharapkan pengguna. Hasil dari pengujian akan diuraikan dalam bentuk tabel yang akan menampilkan *use case*, hasil yang diharapkan, dan hasil pengujian beserta statusnya apakah *use case* dari fungsi tersebut lolos dari pengujian atau tidak. Dengan menggunakan metode pengujian ini, hasil dari setiap pengujian *use case* dapat diketahui dan jika terdapat fungsi yang masih belum sesuai harapan pengguna, dapat dilakukan perbaikan untuk fungsi tersebut [13].

User Acceptance Testing merupakan metode pengujian yang melibatkan pengguna akhir yang akan secara langsung menggunakan aplikasi web ini untuk mengetahui apakah fitur-fitur aplikasi web tersebut menghasilkan *output* yang sudah sesuai dengan kebutuhan yang diharapkan oleh pengguna akhir yang bertujuan untuk digunakan pada lingkungan kerja pengguna akhir tersebut [14].

Pengguna akhir yang melakukan kedua metode pengujian ini dilakukan oleh Bapak Imam Hidayat, S.Tr.AB. selaku staf SHILAU Polibatam yang bekerja di bagian administrasi yang bertanggung jawab atas pengelolaan berbagai dokumen *invoice* di SHILAU.

3.2.1 Black Box Testing

Skenario uji yang dibuat untuk *Black Box testing* meliputi *use case*, deskripsi dari pengujian, hasil yang diharapkan, hasil pengujian dan status keberhasilan. *Black Box Testing* telah dilakukan bersama dengan pengguna akhir dan hasil dari pengujian ini dapat dilihat pada tabel skenario uji *Black Box Testing* pada Tabel 5.

Tabel 5. Skenario Uji *Black Box Testing*

Use Case	Deskripsi Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status Keberhasilan
Mengakses Aplikasi Web	User mengakses aplikasi web melalui <i>browser</i> web dan PC yang terhubung ke internet	Halaman aplikasi web dapat diakses oleh <i>user</i>	Halaman aplikasi web berhasil diakses dan ditampilkan di <i>browser</i> .	Berhasil
Login User	<i>User login</i> ke aplikasi web menggunakan <i>email</i> dan <i>password</i> yang didaftarkan sebelumnya	Sukses <i>login</i> ke aplikasi web dan menampilkan halaman <i>dashboard</i>	<i>User</i> berhasil <i>login</i> menggunakan <i>email</i> dan <i>password</i> yang benar dan diarahkan ke halaman <i>dashboard</i> .	Berhasil
Login User	<i>User</i> tidak bisa <i>login</i> ke aplikasi web jika memasukkan <i>email</i> dan <i>password</i> yang salah	Menampilkan teks merah yang menyatakan bahwa <i>email</i> atau <i>password</i> yang dimasukkan salah	Sistem menampilkan pesan <i>error</i> bertuliskan data yang di-input tidak termasuk dalam <i>database user</i>	Berhasil
Pemulihan Akun	<i>User</i> memulihkan akunnya jika lupa <i>password</i> akunnya dengan mengklik teks 'Lupa Password' yang akan menampilkan halaman 'Lupa Password'. <i>User</i> memasukkan <i>email</i> dan web akan mengirimkan <i>link</i> untuk mengganti <i>password</i> ke <i>email</i> tersebut	<i>Link</i> penggantian <i>password</i> berhasil terkirim ke <i>email</i> yang dimasukkan <i>user</i> dan berhasil memulihkan akun menggunakan <i>password</i> yang baru diganti	<i>Link</i> penggantian <i>password</i> berhasil terkirim ke <i>email</i> dan akun berhasil dipulihkan dengan <i>password</i> baru.	Berhasil
Register User	<i>User</i> mendaftarkan akun lain untuk <i>login</i> dan mengakses fitur-fitur aplikasi web	Berhasil membuat akun untuk <i>login</i> ke aplikasi web	Akun baru berhasil dibuat dan dapat digunakan untuk <i>login</i> serta mengakses fitur aplikasi.	Berhasil
Sistem CRUD Data Invoice	<i>User</i> menambah data <i>invoice</i> melalui halaman <i>invoice</i> dan menyimpannya	Aplikasi web menyimpan data <i>invoice</i> yang ditambahkan ke <i>database</i> dan menampilkannya di tabel data	Data <i>invoice</i> yang ditambahkan berhasil tersimpan ke <i>database</i> dan muncul di tabel data.	Berhasil
Sistem CRUD Data Invoice	<i>User</i> mengedit data yang sudah ada di <i>invoice</i> lalu menyimpannya	Aplikasi web menyimpan data yang di-update dan menampilkannya di tabel	Data <i>invoice</i> yang diedit berhasil ter-update dan tersimpan sesuai perubahan yang dilakukan.	Berhasil
Sistem CRUD Data Invoice	<i>User</i> menghapus data <i>invoice</i> setelah mengonfirmasi tindakannya	Data <i>invoice</i> akan dihapus dari <i>database</i> dan tidak akan ditampilkan di tabel data	Data <i>invoice</i> berhasil dihapus dari <i>database</i> dan tidak lagi muncul di tabel data.	Berhasil
Mencetak Invoice	<i>User</i> men-download <i>invoice</i> dalam format PDF	Data <i>invoice</i> diubah menjadi format PDF dan tersimpan secara lokal di PC <i>user</i>	<i>Invoice</i> berhasil diunduh dalam format PDF dan tersimpan di perangkat <i>user</i> .	Berhasil
Sistem Pencarian	<i>User</i> mencari data <i>invoice</i> melalui <i>search bar</i>	Data <i>invoice</i> akan ditampilkan di bawah <i>search bar</i>	Sistem pencarian berhasil menampilkan data <i>invoice</i> sesuai kata kunci dan filter yang dimasukkan.	Berhasil
Mengekspor Data	<i>User</i> mengekspor data <i>invoice</i> dalam format CSV	Data <i>invoice</i> akan diekspor menjadi format CSV dan tersimpan secara lokal di PC <i>user</i>	Data <i>invoice</i> berhasil diekspor dalam format CSV dan tersimpan di perangkat <i>user</i> .	Berhasil
Sistem CRUD Akun User	<i>User</i> mengganti nama akun, <i>email</i> , dan <i>password</i> dari akun tersebut di halaman profil dan menyimpannya	Data akun <i>user</i> yang diganti disimpan ke <i>database</i> dan menampilkan datanya di halaman profil serta berhasil <i>login</i> menggunakan <i>email</i> dan <i>password</i> yang sudah diubah	Data akun (nama, <i>email</i> , <i>password</i>) berhasil diperbarui dan tersimpan di <i>database</i> .	Berhasil

<i>Logout User</i>	<i>User keluar dari akun dan mengakhiri sesi login aplikasi web</i>	<i>Menampilkan halaman login dan user harus login ulang jika ingin mengakses fitur-fitur aplikasi web</i>	<i>User berhasil logout dan sistem mengarahkan kembali ke halaman login</i>	Berhasil
--------------------	---	---	---	----------

Hasil dari pengujian *Black Box Testing* dihitung dengan persentase *pass rate* untuk menentukan tingkat keberhasilan fungsional [15]. Rumus persamaan persentase *pass rate* untuk skenario uji ini dinyatakan sebagai berikut:

$$PassRate = \frac{13}{13} \times 100\% = 100\%$$

Hasil perhitungan dengan menggunakan rumus *pass rate* memberikan hasil 100%, yang menyatakan bahwa semua *use case* yang diuji menghasilkan *output* yang sesuai dengan harapan pengguna akhir aplikasi web ini.

3.2.2 User Acceptance Testing (UAT)

Pengujian *User Acceptance Testing* menggunakan metode kualitatif dengan melakukan wawancara secara langsung dengan pengguna akhir setelah pengguna akhir tersebut menggunakan aplikasi web ini. Pertanyaan wawancara ini diukur dengan nilai *learnability*, *ease of use*, *efficiency*, *suitability*, *error handling*, dan *satisfaction* [14]

Tabel 6. Transkrip Wawancara dengan Pengguna Akhir

Pertanyaan Wawancara	Pengukuran Nilai	Transkrip Wawancara Pengguna Akhir
Apakah aplikasi web ini membingungkan atau mudah digunakan?	<i>Learnability</i>	“Secara penggunaan, bisa dibilang cukup mudah dan ada beberapa hal yang perlu disesuaikan.”
Apakah UI dari webnya mudah dipahami dan dinavigasi?	<i>Ease Of Use</i>	“UI dari web ini cukup sederhana ya, jadi ada beberapa menu-menu yang sudah terlihat, sehingga bisa dengan mudah kita gunakan.”
Apakah performa loading dari web ini termasuk cepat?	<i>Efficiency</i>	“Kalau dilihat tadi tergantung jaringannya, jadi kalau misalnya memang jaringannya oke, ya harusnya loadingnya tadi tidak terlalu lama ya.”
Apakah format dari PDF yang di-export sudah sesuai dengan standar SHILAU?	<i>Suitability</i>	“Perlu ada penyesuaian karena ada bagian-bagian yang mungkin perlu ditambahkan karena SHILAU memerlukan tanda tangan langsung walaupun bisa digital, sementara yang sekarang belum ada tanda tangannya karena SHILAU lembaga resmi, jadi perlu ditambahkan nanti terkait dengan tanda tangan dan juga stempel digital.”
Apakah terdapat bug atau error saat penggunaan?	<i>Error Handling</i>	“Setelah menggunakan web ini, bisa dibilang tidak terjadi error. Hanya terjadi masalah pada jaringannya saja ya.”
Secara keseluruhan, apakah Bapak puas dengan aplikasi web ini?	<i>Satisfaction</i>	“Kalau secara keseluruhan ya, barangkali bisa dibilang cukup puas ya, tinggal barangkali ada beberapa penyesuaian. Kalau bisa itu disesuaikan, mungkin saya bisa merasa lebih puas lagi, tinggal nanti bagaimana kita aplikasikannya di SHILAU gitu kan, karena kalau misalnya sudah disesuaikan, ini sangat bagus gitu, bisa memudahkan”

Berdasarkan transkrip wawancara di Tabel 6, dapat disimpulkan bahwa dari aspek *learnability* dan *ease of use*, pengguna akhir menilai aplikasi web ini mudah digunakan meskipun terdapat beberapa penyesuaian pada penggunaan. Kalau dari aspek *efficiency* dan *error handling*, pengguna akhir menilai performa aplikasi web ini berjalan tanpa *error* namun terdapat masalah pada konektivitas dari jaringan yang digunakan.

Terdapat juga temuan penting pada aspek *suitability* dan *satisfaction*, bahwa pengguna akhir memberikan saran bahwa PDF yang di-*export* perlu disesuaikan lagi dengan menambah tanda tangan dan stempel digital ke PDF yang di-*export*. Penilaian ini menjadi rekomendasi perbaikan untuk meningkatkan nilai dari aplikasi web ini.

4. KESIMPULAN

Penelitian ini menghasilkan aplikasi web manajemen data invoice untuk SHILAU Polibatan yang dikembangkan menggunakan framework Laravel dan MySQL dengan menerapkan metode *Personal Scrum* untuk pengembang tunggal yang berperan sekaligus sebagai *Product Owner*, *Scrum Master*, dan *Developer*. Proses pengembangan dilaksanakan melalui lima *sprint* dengan durasi masing-masing dua minggu, mencakup implementasi fitur autentikasi pengguna, sistem CRUD data *invoice*, pencarian dan penyaringan data, ekspor dokumen ke format PDF dan CSV, serta fitur *backup data* secara berkala.

Hasil pengujian menggunakan metode *Black Box Testing* terhadap 13 skenario uji menunjukkan tingkat keberhasilan sebesar 100%, yang mengindikasikan bahwa seluruh fungsi utama aplikasi berjalan sesuai dengan hasil yang diharapkan. Hasil ini didukung oleh pengujian *User Acceptance Testing* yang dilakukan bersama pengguna akhir, yaitu staf SHILAU yang bertanggung jawab dalam pengelolaan dokumen *invoice*. Berdasarkan hasil wawancara, pengguna akhir menilai aplikasi ini mudah dipelajari dan digunakan, memiliki UI yang sederhana dan mudah dinavigasi, serta tidak ditemukan kendala berupa *bug* atau *error* selama penggunaan, kecuali permasalahan yang berkaitan dengan kondisi jaringan internet. Pengguna akhir juga menyatakan kepuasan terhadap aplikasi ini secara keseluruhan, meskipun terdapat catatan perbaikan pada aspek kesesuaian format dokumen, yaitu perlunya penambahan fitur tanda tangan dan stempel digital pada dokumen PDF yang dihasilkan, mengingat SHILAU merupakan bagian dari lembaga resmi yang memerlukan kelengkapan unsur legalitas pada setiap dokumen transaksinya.

Berdasarkan hasil pengujian dan evaluasi tersebut, dapat disimpulkan bahwa aplikasi web manajemen data *invoice* ini telah berhasil memenuhi kebutuhan fungsional dan non-fungsional yang telah dirancang, serta dapat diterima oleh pengguna akhir untuk digunakan dalam lingkungan kerja SHILAU dengan catatan perbaikan pada penambahan fitur tanda tangan dan stempel digital. Untuk pengembangan selanjutnya, disarankan agar aplikasi ini dapat diintegrasikan dengan jenis dokumen transaksi lain seperti *quotation*, *purchase order*, *delivery order*, dan kuitansi, sehingga seluruh proses pengelolaan dokumen SHILAU, mulai dari penawaran hingga bukti pembayaran dapat dilakukan dalam satu sistem yang saling terhubung tanpa perlu berpindah antar aplikasi yang berbeda.

DAFTAR PUSTAKA

- [1] "Tentang SHILAU - SHILAU." Accessed: Sep. 19, 2025. [Online]. Available: <https://shilau.polibatam.ac.id/tentang>
- [2] A. Wijaya and M. Ilmi, "PERANCANGAN APLIKASI INVOICE PADA PERUSAHAAN LESTARI MANDIRI JAYA BERBASIS WEB," *Jutekdiksi: Jurnal Teknologi Digital dan Sistem Informasi*, vol. 1, no. 1, pp. 1–4, Mar. 2024, Accessed: Mar. 23, 2025. [Online]. Available: <https://ojsiibn1.indobaronasional.ac.id/index.php/JUTEKDISI/article/view/786>
- [3] A. Zulfian, P. Putra, and D. Meilani, "Pengembangan Aplikasi Invoice Otomatis Berbasis Website Menggunakan Model Waterfall," *Prosiding Seminar Implementasi Teknologi Informasi dan Komunikasi*, vol. 4, no. 1, pp. 272–279, Jan. 2025, doi: 10.31284/p.semtik.2025-1.7069.
- [4] T. Khan Mohd, J. Thompson, A. Carmine, and G. Reuter, "Comparative Analysis on Various CSS and JavaScript Frameworks," *Journal of Software*, vol. 17, pp. 282–291, Nov. 2022, doi: 10.17706/jsw.17.282-291.
- [5] K. W. Nofa and M. R. N. Ichsan, "RANCANG BANGUN APLIKASI WEBSITE PENJUALAN MAKANAN BEKU MENGGUNAKAN LARAVEL," *JTS*, vol. 1, no. 2, pp. 125–132, Jun. 2022.
- [6] Nor Asma Mohd Zin, Ain Nadia Mazlan, Noorfadzilah Arifin, Noorihan Abdul Rahman, Marina Ahmad, and Maznie Manaf, "Applying Personal Scrum Methodology in System Development for Project-Based Learning Course," *Journal of Mathematics and Computing Science*, vol. 10, no. 2, pp. 64–73, Dec. 2024, doi: 10.24191/jmcs.v10i2.4508.
- [7] "The Scrum Framework Poster | Scrum.org." Accessed: Sep. 29, 2025. [Online]. Available: <https://www.scrum.org/resources/scrum-framework-poster>
- [8] W. Warkim, M. H. Muslim, F. Harvianto, and S. Utama, "Penerapan Metode SCRUM dalam Pengembangan Sistem Informasi Layanan Kawasan," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 6, no. 2, Aug. 2020, doi: 10.28932/jutisi.v6i2.2711.
- [9] A. A. Kadim, I. K. Sutriana, and I. H. Masir, "Perancangan Sistem Aplikasi Layanan Kelurahan Berbasis Web," *Jambura Journal of Informatics*, vol. 4, no. 1, pp. 38–48, Apr. 2022, doi: 10.37905/jji.v4i1.13206.
- [10] Subianto and J. Cristian L, "Perancangan Sistem Pengelolaan Dokumen Invoice Berbasis Web," *Infokam: Informasi Komputer Akuntansi dan Manajemen*, vol. 16, no. 2, pp. 157–167, Sep. 2020, doi: <https://doi.org/10.53845/infokam.v16i2.277>.
- [11] I. R. Mukhlis and R. Santoso, "Perancangan Basis Data Perpustakaan Universitas Menggunakan MySQL dengan Physical Data Model dan Entity Relationship Diagram," *Journal of Technology and Informatics (JoTI)*, vol. 4, no. 2, pp. 81–87, Apr. 2023, doi: 10.37802/joti.v4i2.330.
- [12] V. Hema, S. Thota, S. Naresh Kumar, C. Padmaja, C. B. Rama Krishna, and K. Mahender, "Scrum: An Effective Software Development Agile Tool," in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing Ltd, 2020. doi: 10.1088/1757-899X/981/2/022060.
- [13] M. Shintania, A. Thifal Ananda, A. Perdana, M. S. Fadlurrohman, and A. T. Zy, "Pengujian Aplikasi Invoice Untuk Freelancer Small Business Berbasis Web Dengan Metode Blackbox," *Jurnal Sains Komputer & Informatika (J-SAKTI)*, vol. 8, no. 1, pp. 196–205, Mar. 2024, doi: <http://dx.doi.org/10.30645/j-sakti.v8i1.778>.
- [14] A. P. D. Nugraha and W. L. Y. Saptomo, "Usability Evaluation And User Acceptance Testing (Uat) On Website-Based Digital Psychology Testing Application: A Case Study Of Psikotestyuk.Id," *Jurnal Media Computer Science*, vol. 5, no. 3, pp. 1109–1126, Jul. 2026.
- [15] S. Muhtadi, S. J. Saputra, and C. Anwar, "Penerapan Black Box Testing Berstandar ISO 29119 Pada Aplikasi," *journal.jci.co.id/jisbt*, vol. 1, no. 3, 2025, doi: 10.37034/jsisfotek.v5i2.231.