



Penerapan *Payment Gateway* pada Aplikasi *Wedding Organizer* Berbasis *Web* dengan Metode *Agile* menggunakan *Framework Scrum*

Ridwan Setiawan¹, Rickard Elsen², Via Nur Afifah³

Jurnal Algoritma
Institut Teknologi Garut
Jl. Mayor Syamsu No. 1 Jayaraga Garut 44151 Indonesia
Email: jurnal@itg.ac.id

¹ridwan.setiawan@itg.ac.id

²rickardelsen@itg.ac.id

³1906166@itg.ac.id

Abstrak – *Wedding Organizer* (WO) adalah penyedia layanan *event organization* yang bekerja langsung untuk membantu calon pengantin dalam perencanaan acara pernikahan dengan waktu dan biaya yang telah ditentukan. Permasalahan yang sering terjadi diantaranya pada saat klien atau calon pengantin menentukan paket pernikahan. Proses pemilihan paket seringkali memerlukan waktu yang cukup lama karena membutuhkan diskusi yang mendalam. Selain itu, proses pembayaran juga menjadi perhatian karena saat ini klien hanya dapat membayar melalui transfer dengan mengirimkan bukti pembayaran, yang memakan waktu untuk diverifikasi oleh admin. Tujuan dari penelitian ini yaitu menerapkan teknologi *payment gateway* pada aplikasi *wedding organizer* berbasis *web* yang dapat menyajikan informasi mengenai layanan paket pernikahan dan dapat melakukan pemesanan dengan mudah dan cepat. Metodologi yang digunakan dalam penelitian ini yaitu *Agile* dengan *Framework Scrum* melalui tahapan *Requirements*, *Product Backlog*, *Sprint Planning*, *Sprint*, *Sprint Review*, dan *Sprint Retrospective*. Untuk pemodelan sistem pada penelitian ini menggunakan *Unified Modeling Language* (UML) dengan 2 diagram yaitu *Use Case Diagram* dan *Class Diagram*. Hasil dari penelitian ini yaitu aplikasi *wedding organizer* berbasis *web* yang dapat melakukan pembayaran secara *real time* dengan diimplementasikannya salah satu *payment gateway* yaitu *midtrans* sehingga memudahkan admin dalam proses verifikasi pembayaran dan juga agar klien bisa bertransaksi dengan mudah dan cepat.

Kata Kunci – *Agile*; *Payment Gateway*; *Scrum*; *Web*; *Wedding Organizer*.

I. PENDAHULUAN

Perkembangan teknologi informasi telah memberikan dampak yang signifikan dalam berbagai sektor, termasuk dalam bidang bisnis [1]. Salah satu contohnya adalah bisnis *Wedding Organizer* (WO), di mana masyarakat kini memanfaatkan teknologi untuk mengembangkan bisnis pernikahan secara *online*. Selain itu, kemudahan aplikasi berbelanja *online* semakin ditingkatkan dengan adanya metode pembayaran yang memungkinkan proses pembelian atau pemesanan menjadi lebih cepat dan praktis [2]. *Wedding Organizer* (WO) adalah penyedia layanan *event organization* yang bekerja langsung untuk membantu calon pengantin dalam perencanaan acara pernikahan dengan waktu dan biaya yang telah ditentukan [3]. Namun, banyak *Wedding Organizer* yang masih mengandalkan sistem konvensional dengan pertemuan langsung antara WO dan klien. Salah satunya, Mantumanten WO yang merupakan jasa *Wedding Organizer* di Kabupaten Garut masih menghadapi masalah dalam menentukan paket pernikahan. Diskusi antara klien dan WO mengenai paket pernikahan seringkali memakan waktu yang lama. Selain itu, proses pembayaran melalui transfer juga

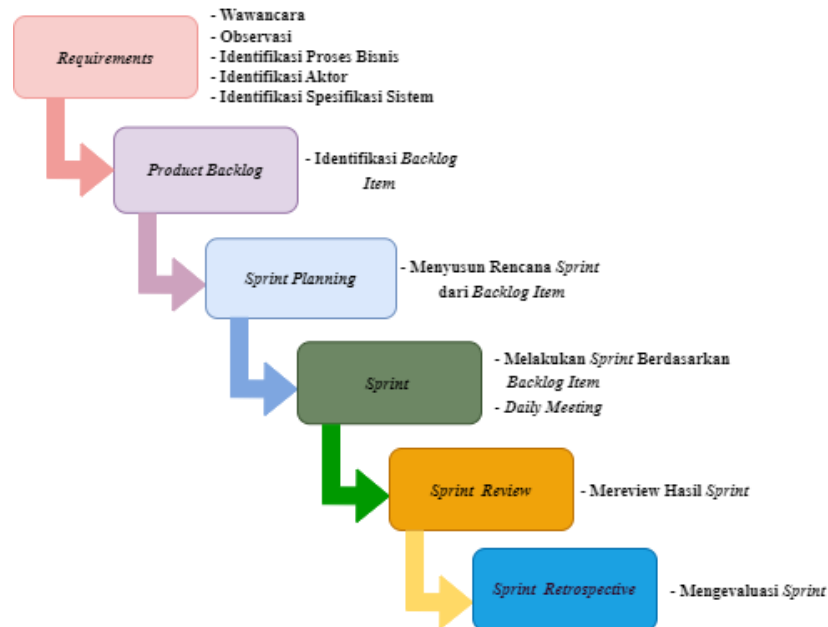
kurang efisien karena memerlukan waktu untuk verifikasi oleh admin. Untuk mengatasi hal ini, maka *Wedding Organizer* akan mengadopsi teknologi *payment gateway*. *Payment gateway* merupakan metode pembayaran *online* yang memfasilitasi verifikasi otomatis dan pencatatan transaksi dalam sistem, sehingga klien tidak perlu lagi mengirimkan bukti pembayaran secara terpisah [4]. *Payment gateway* memberikan berbagai keuntungan dan kemudahan bagi pelaku *e-commerce* untuk melakukan transaksi keuangan secara digital, menggunakan internet. Melalui *Payment Gateway*, proses pembayaran dalam bisnis *online* menjadi lebih efisien dan aman, memfasilitasi transaksi secara elektronik dengan lebih mudah dan terpercaya [5]. Salah satu penyedia layanan *payment gateway* terbesar yang ada di Indonesia adalah *Midtrans* dengan ribuan pelanggan bisnis di berbagai sektor dan industri. *Midtrans* adalah salah satu *platform* untuk menggunakan *payment gateway*. Dengan menggunakan *Midtrans*, penjual dan *customers* bisa dengan mudah melakukan transaksi melalui sistem pembayaran ini [6].

Penelitian ini didasari dengan beberapa penelitian sebelumnya. Penelitian pertama oleh Rosita Wulandari, Ridwan Setiawan, dan Asri Mulyani (2020) tentang merancang sistem informasi mengenai *Wedding Organizer* berbasis *web* dengan metode *Scrum* untuk memudahkan *user* dalam mencari informasi, harga, dan melakukan transaksi jasa *Wedding Organizer* secara *online* [7]. Penelitian kedua oleh Haris Ginanjar dan Ridwan Setiawan (2020) tentang merancang sebuah aplikasi *Wedding Organizer* berbasis *Android* yang dapat membantu calon pengantin dalam memesan layanan paket pernikahan melalui perangkat *Android* [8]. Penelitian ketiga oleh Dennis Gunadi dan Riki (2022) menganalisis dan merancang aplikasi *Wedding Planner* dengan metode *prototype* dengan tujuan untuk membantu klien dalam merencanakan acara pernikahan sesuai dengan kebutuhan mereka [9]. Penelitian keempat oleh Lutfi Andrea Ridardo (2022) membangun aplikasi *Wedding Organizer* berbasis *web* menggunakan metode *waterfall* untuk mengelola data pemesanan dengan lebih efisien [3]. Penelitian kelima oleh Tri May Mega Puspasari dan Dina Maulina (2019) mengimplementasikan *payment gateway* pada marketplace menggunakan *Midtrans*, mempermudah pembayaran *online* bagi pengguna dan memberikan kemudahan bagi admin dalam melihat status pemesanan [10].

Berdasarkan pemaparan masalah diatas maka tujuan dari penelitian ini yaitu dapat membantu klien (calon pengantin) dalam menentukan paket pernikahan yang diinginkan tanpa perlu datang ketempat WO secara langsung dan untuk sistem pembayaran secara *online* menggunakan *payment gateway* dilakukan untuk membantu admin dalam mengelola proses pembayaran tanpa harus verifikasi pembayaran *customer* dan juga agar *customer* bisa bertransaksi dengan mudah dan cepat.

II. METODOLOGI PENELITIAN

Metode pengembangan *software* yang digunakan untuk mendukung penelitian yaitu *Scrum*. *Scrum* adalah sebuah *framework* atau kerangka kerja dalam pengembangan perangkat lunak yang digunakan untuk mengembangkan produk dengan pendekatan yang *iteratif* dan *incremental* yang didasarkan pada prinsip-prinsip *Agile* [11]. Adapun tahapan yang dilakukan dalam penerapan *framework scrum* diantaranya *requirement*, *product backlog*, *sprint planning*, *sprint*, *sprint review*, dan *sprint retrospective* [11]. Tahapan pengembangan tersebut dapat dilihat pada Gambar 1.



Gambar 1: Tahapan Pengembangan Aplikasi [11]

Untuk uraian penjelasan dari Gambar 1. Tahapan Pengembangan Aplikasi adalah sebagai berikut:

1. *Requirements*, tahap ini diawali dengan aktivitas untuk pengumpulan data dengan observasi dan wawancara. Pada tahap ini juga menentukan proses bisnis yang disepakati dengan *product owner*, penentuan aktor yang akan terlibat dan mengidentifikasi spesifikasi sistem. Adapun hasilnya berupa proses bisnis yang sedang berjalan, aktor yang terlibat dan spesifikasi sistem kemudian akan diidentifikasi ditahapan *product backlog* [12].
2. *Product Backlog*, aktivitas yang dilakukan adalah mengidentifikasi kebutuhan sistem seperti komponen apa yang akan dibutuhkan pada sistem yang akan dibuat dengan disajikan dalam bentuk tabel. Sehingga hasil dari tahap ini yaitu tabel *backlog item* [13].
3. *Sprint Planning*, aktivitas yang dilakukan adalah menyusun perencanaan kebutuhan sistem yang telah dibuat dalam bentuk tabel *backlog item* disesuaikan dengan tingkat kesulitan dan menentukan estimasi waktu dalam pembuatan yang akan dilakukan sehingga diperoleh menu atau fitur yang akan dibutuhkan oleh sistem. Adapun hasil dari tahapan ini yaitu tabel *sprint backlog* [13].
4. *Sprint*, aktivitas yang dilakukan adalah pengimplementasian mengenai menu atau fitur yang telah ditentukan kedalam bahasa pemrograman. Selama kegiatan *sprint* berlangsung dilakukan *meeting* harian yang bertujuan untuk memantau kinerja, pertemuan dilakukan untuk melaporkan fitur yang sedang dibangun, serta dilakukan sinkronisasi untuk menyesuaikan hasil dari pekerjaan yang dilakukan [14].
5. *Sprint Review*, dilakukan aktivitas persentasi hasil dari sinkronisasi *software product* yang telah dibuat pada tahap *sprint* kepada *product owner* dengan menilai kesesuaian terhadap perencanaan pada setiap *backlog* yang berjalan [13].
6. *Sprint Retrospective*, Pada tahap *sprint retrospective* dilakukan evaluasi terhadap *sprint* yang telah berjalan. Salah satu evaluasi yang dilakukan yaitu mengenai estimasi waktu pada *product backlog* dengan realisasi pengerjaan yang digambarkan menggunakan *burndown chart* untuk mengetahui *sprint* sebelumnya berjalan [13].

III. HASIL DAN PEMBAHASAN

A. Hasil Penelitian

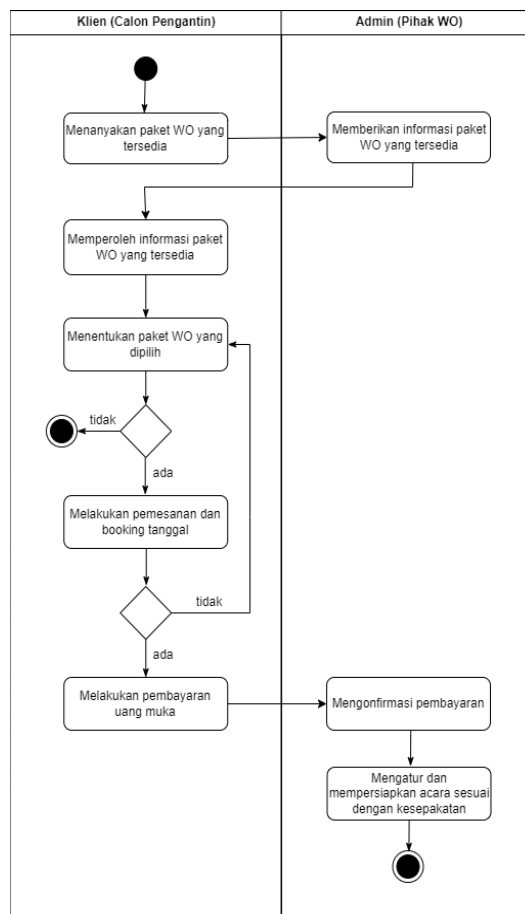
1. *Requirements*
 - a. Identifikasi Proses Bisnis

Untuk penggambaran dari proses bisnis manual dapat dilihat pada Gambar 2. Gambar ini merupakan proses bisnis yang sedang berjalan di Mantumanten WO ketika *customer* melakukan pemesanan dimana *customer* setelah menentukan paket yaitu melakukan pemesanan dan *booking* tanggal. Jika layanan tersedia *customer* langsung melakukan pembayaran uang muka, dan admin akan mengonfirmasi pembayaran dan mengatur persiapan acara sesuai dengan kesepakatan.

b. Identifikasi Spesifikasi Sistem

Adapun untuk spesifikasi sistem yang dibuat diantaranya sebagai berikut:

- 1) Sistem dapat mengakomodir perbedaan level akses *user*;
- 2) Sistem dapat menampilkan produk kategori paket;
- 3) Sistem dapat menambahkan, menghapus, dan mengedit informasi mengenai paket WO;
- 4) Sistem dapat melakukan pembatalan otomatis jika pembayaran dp tidak dilakukan lebih dari 1x24 jam;
- 5) Sistem dapat melakukan verifikasi pembayaran melalui *payment gateway*;
- 6) Sistem ini menyediakan berbagai fitur diantaranya, admin dapat mengelola kategori paket dari mulai menambah, mengedit dan menghapus, admin dapat mengelola galeri, admin juga dapat mengelola data seperti data customer dan pemesanan. Selain itu *customer* dapat melihat informasi mengenai layanan paket yang tersedia dan melakukan *booking* paket.



Gambar 2: Proses Bisnis Manual

Berdasarkan proses bisnis yang berjalan maka ditentukanlah aktor sebagai berikut yaitu Admin dan *Customer*.

2. Product Backlog

Pada tahap ini dilakukan aktivitas mengidentifikasi kebutuhan sistem seperti komponen apa yang akan

dibutuhkan pada sistem yang disajikan pada Tabel 1 berikut.

Tabel 1: *Backlog Item*

No.	Aktivitas <i>Product Backlog</i>	Keterangan
1.	<i>Use Case Diagram</i>	Diagram untuk menggambarkan interaksi antara Admin/ <i>Customer</i> dengan sistem <i>wedding organizer</i> .
2.	<i>Class Diagram</i>	Diagram ini menjelaskan mengenai struktur sistem berdasarkan kelas-kelas.
3.	Struktur Sistem	Struktur untuk memandu pengguna dalam mengakses dan menggunakan fitur-fitur atau fungsi-fungsi pada sistem <i>wedding organizer</i> .
4.	Rancangan <i>Interface</i>	Proses merancang tata letak, elemen visual, dan interaksi antarmuka pengguna.
5.	Halaman <i>Landing Page</i>	Halaman utama saat mengakses sistem.
6.	Halaman <i>Login</i>	Halaman untuk <i>user</i> masuk ke sistem
7.	Halaman <i>Register</i>	Halaman untuk daftar jika belum punya akun.
8.	Halaman <i>Dashboard Admin</i>	Halaman utama ketika masuk sebagai admin.
9.	Halaman <i>Dashboard Customer</i>	Halaman utama ketika masuk sebagai <i>customer</i> .
10.	Halaman Profil <i>Admin</i>	Halaman mengenai informasi admin.
11.	Halaman Master Kategori	Halaman mengenai kategori paket yang terdiri dari WO Only dan All In.
12.	Halaman Master Data	Halaman mengenai informasi data termasuk data <i>customer</i> , data galeri dan data pemesanan.
13.	Halaman Kalender Pemesanan	Halaman mengenai informasi penjadwalan mengenai pemesanan paket WO.
14.	Halaman Kategori	Halaman berisi informasi mengenai kategori paket.
15.	Halaman <i>Cart</i>	Halaman mengenai informasi keranjang pemesanan <i>customer</i> . dan halaman pemesanan (<i>checkout</i>) <i>customer</i> .
16.	Halaman Riwayat Pemesanan	Halaman yang berisi informasi mengenai riwayat pemesanan <i>customer</i> .

3. *Sprint Planning*

Pada tahap *sprint planning*, aktivitas yang dilakukan adalah menyusun perencanaan kebutuhan sistem yang telah dibuat pada tabel *backlog item* disesuaikan dengan tingkat kesulitan dan menentukan estimasi waktu yang disajikan pada Tabel 2 berikut.

Tabel 2: *Sprint Backlog*

No.	Aktivitas <i>Product Backlog</i>	Estimasi (Jam)	Tingkat Kesulitan
1.	<i>Use Case Diagram</i>	12	<i>Height</i>
2.	<i>Class Diagram</i>	12	<i>Height</i>
3.	Struktur Sistem	13	<i>Medium</i>
4.	Rancangan <i>Interface</i>	20	<i>Height</i>
5.	<i>Landing Page</i>	14	<i>Medium</i>
6.	<i>Login</i>	11	<i>Medium</i>
7.	<i>Register</i>	11	<i>Medium</i>
8.	<i>Dashboard Admin</i>	13	<i>Medium</i>
9.	<i>Dashboard Customer</i>	12	<i>Medium</i>
10.	Profil Admin	10	<i>Medium</i>
11.	Master Kategori	18	<i>Height</i>
12.	Master Data	15	<i>Height</i>
13.	Kalender Pemesanan	16	<i>Height</i>
14.	Kategori	12	<i>Medium</i>
15.	<i>Cart</i>	15	<i>Height</i>
16.	Riwayat Pemesanan	14	<i>Height</i>

4. *Sprint*

Pada tahap ini dilakukan aktivitas merancang *sprint* berdasarkan elemen *backlog* dengan tujuan untuk membuat kebutuhan pengguna menjadi prioritas dengan waktu kerja yang ditentukan yang disajikan pada Tabel dibawah ini:

Tabel 3: *Sprint 1 Use Case Diagram*

Sprint 1	Pengerjaan	Estimasi (Waktu/ Jam)				
		1	2	3	4	5
Use Case Diagram	Merancang Use Case Diagram	6	6			
	Total	12 Jam				

Pada Tabel 3 yaitu *Sprint 1* Merancang *Use Case Diagram* dengan total estimasi waktu sebanyak 11 jam yang dilaksanakan 2 hari di minggu ke-1, pada hari pertama dilaksanakan selama 5 jam dan hari kedua selama 5 jam.

Tabel 4: *Sprint 2 Login*

Sprint 2	Pengerjaan	Estimasi (Waktu/ Jam)				
		1	2	3	4	5
Login	Membuat <i>database</i>		1			
	Pengkodean				=	4
	Testing	2				
	Total	11 Jam				

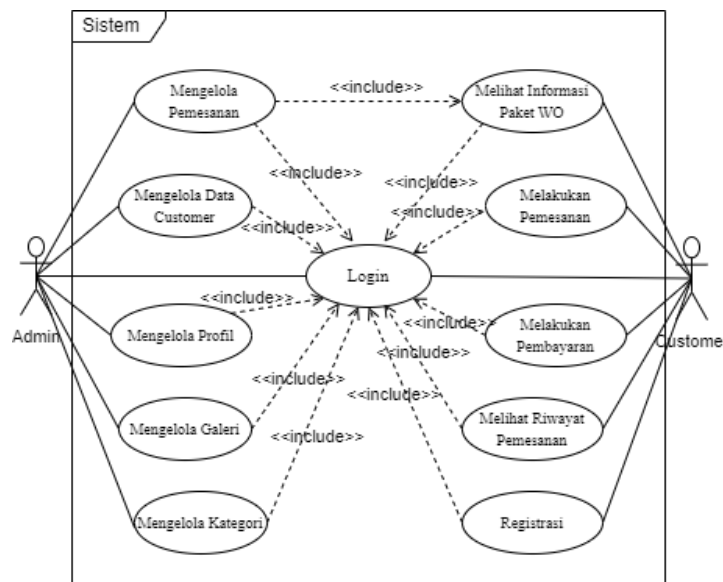
Pada Tabel 4 yaitu *Sprint 2* Pengkodean *form login* dengan total estimasi waktu sebanyak 12 jam yang dilaksanakan 4 hari, pada hari ketiga dilaksanakan selama 1 jam untuk membuat *database*, hari keempat dan kelima selama 9 jam untuk pengerjaan *form login* dan 2 jam pada hari pertama minggu ke-4 testing dilakukan.

5. *Sprint Review*

Pada tahap *sprint review* dilakukan aktivitas mereview *sprint* yang telah berjalan. Adapun hasil dari *sprint review* adalah sebagai berikut:

a. *Use Case Diagram*

Untuk penggambaran *use case* dapat dilihat pada Gambar 3 berikut.



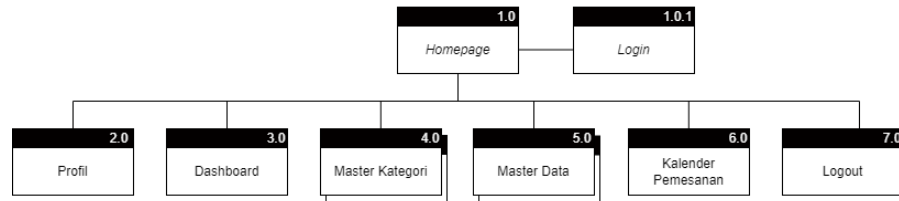
Gambar 3: *Use Case Diagram*

Pada Gambar 3. *Use case* diatas terdapat dua aktor dimana terdapat 5 *use case* untuk admin dan 6 *use case* untuk customer.

b. Struktur Sistem

Struktur sistem, digunakan sebagai penggambaran rincian dari sistem. Perancangan struktur sistem tersebut dapat disajikan pada gambar berikut ini.

1) Struktur Menu Admin



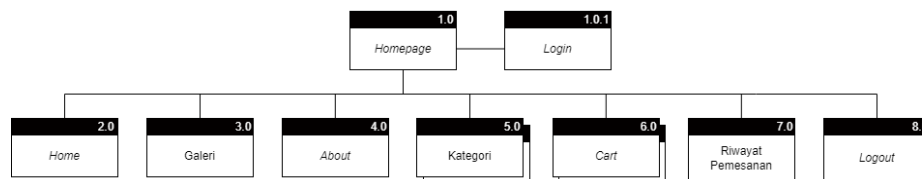
Gambar 4: Struktur Menu Admin

Adapun penjelasan dari setiap menu admin disajikan pada Tabel 5. Berikut:

Tabel 5: Deskripsi Menu Admin

No.	Nama Menu	Keterangan
1.0	Home Page	Menu <i>homepage</i> adalah menu yang mengarahkan Admin ke halaman pertama <i>website</i> .
1.0.1	Login	Berisi <i>username</i> dan <i>password</i> untuk memasuki laman <i>dashboard</i> Admin.
2.0	Profil	Menampilkan detail profil dari pengguna yang bisa di edit.
3.0	Dashboard	Menampilkan jumlah dari <i>customer</i> , transaksi yang pending dan transaksi yang sudah selesai
4.0	Master Kategori	Berisi informasi kategori paket yang bisa di tambah,edit dan hapus.
5.0	Master Data	Berisi informasi data <i>customer</i> , data galeri dan data pemesanan.
6.0	Kalender Pemesanan	Berisi informasi penjadwalan mengenai pemesanan paket WO.
7.0	Logout	Menu untuk keluar dari sistem.

2) Struktur Menu *Customer*

Gambar 5: Struktur Menu *Customer*

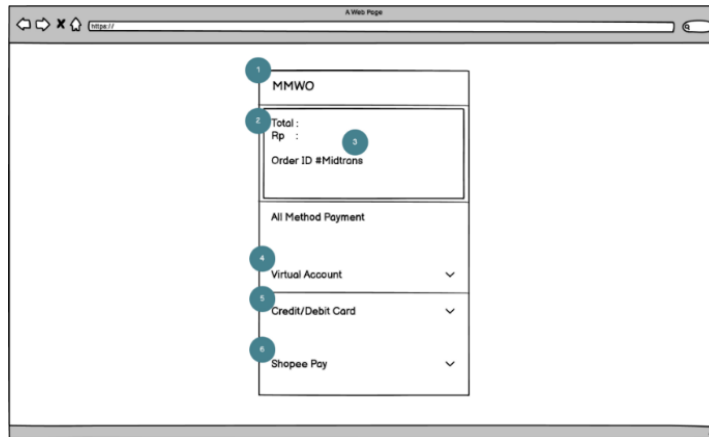
Adapun penjelasan dari setiap menu *customer* disajikan pada Tabel 6. berikut:

Tabel 6: Deskripsi Menu *Customer*

No.	Nama Menu	Keterangan
1.0	Home Page	Menu <i>homepage</i> adalah menu yang mengarahkan ke halaman pertama <i>website</i>
1.0.1	Login	Berisi <i>username</i> dan <i>password</i> untuk memasuki laman <i>dashboard customer</i> .
2.0	Home	Menampilkan halaman <i>home page</i> .
3.0	Galeri	Menampilkan gambar dari paket yang tersedia.
4.0	About	Berisi informasi mengenai Mantumanten WO.
5.0	Kategori	Berisi informasi kategori paket pernikahan yang terdiri dari <i>WO Only</i> dan <i>All In</i> .
6.0	Cart	Berisi informasi mengenai keranjang pemesanan dan melakukan pemesanan (<i>checkout</i>)
7.0	Riwayat Pemesanan	Menampilkan riwayat pemesanan yang dilakukan <i>customer</i> .
8.0	Logout	Menu untuk keluar dari sistem.

c. Rancangan *Interface*

Berikut merupakan rancangan *interface* untuk halaman metode pembayaran yang dapat dilihat pada Gambar 6.



Gambar 6: Rancangan *Interface* Metode Pembayaran

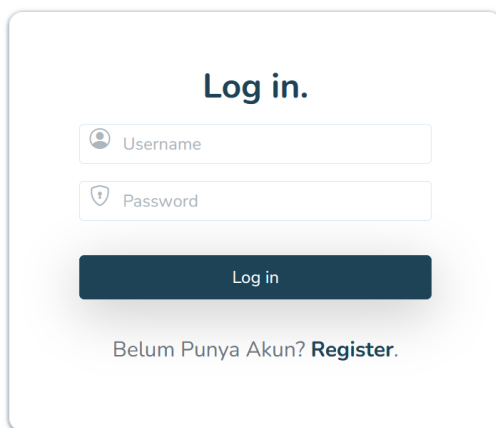
Adapun penjelasan dari rancangan *interface* metode pembayaran dapat disajikan pada Tabel 8. berikut:

Tabel 7: Deskripsi *Interface* Metode Pembayaran

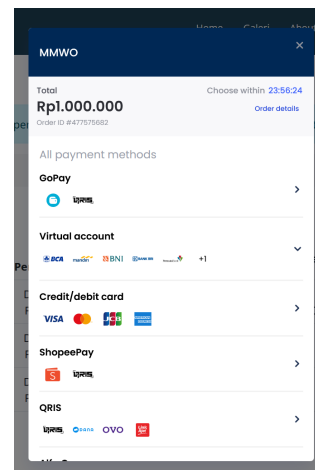
No.	Nama Menu	Keterangan
1.	Nama <i>Project</i>	Digunakan untuk judul <i>project</i> yang digunakan
2.	Total	Jumlah total harga yang harus dibayar
3.	ID	ID <i>midtrans</i>
4.	Metode VA	Metode pembayaran dengan <i>Virtual Account</i>
5.	Metode <i>Credit/Debit Card</i>	Metode pembayaran dengan <i>Credit/Debit Card</i>
6.	Metode Shopee Pay	Metode pembayaran dengan Shopee Pay

d. Implementasi Bahasa Pemrograman

Berikut merupakan hasil implementasi dari perancangan *interface* yang telah dibuat kedalam Bahasa pemrograman.



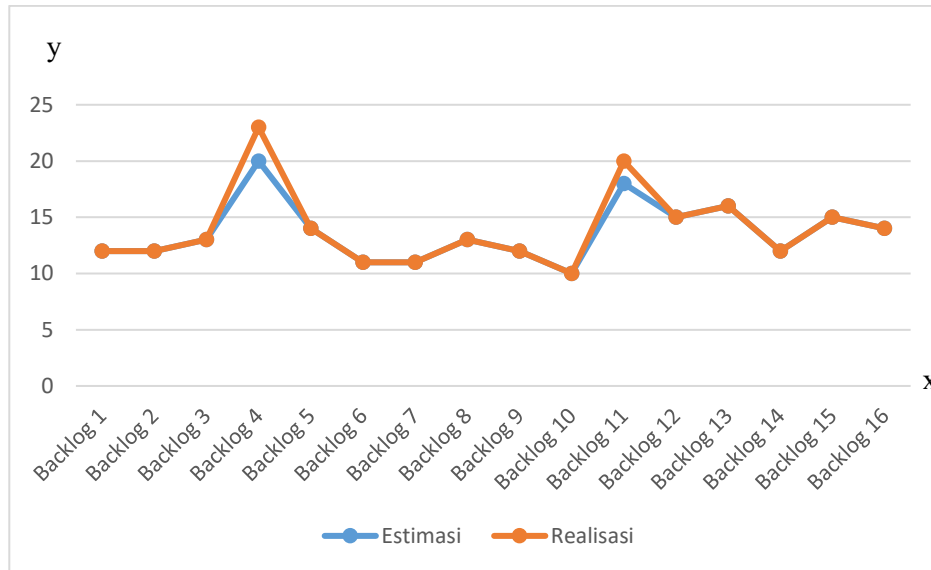
Gambar 7: Tampilan *Login*



Gambar 8: Tampilan Metode Pembayaran

6. *Sprint Retrospective*

Pada tahap *sprint retrospective* dilakukan evaluasi terhadap *sprint* yang telah berjalan yang digambarkan menggunakan *burndown chart* untuk mengetahui *sprint* sebelumnya.

Gambar 9: *Sprint Retrospective*

Dapat dilihat pada *Burndown Chart* di atas untuk sumbu x merupakan *backlog* dan sumbu y merupakan estimasi/waktu/jam dari hasil aktivitas yang telah dilakukan selama *sprint* berlangsung. Berdasarkan Gambar 11. di atas total estimasi waktu (/jam) yaitu 218 jam atau setara dengan 27 hari dan secara realita total pengerjaan menghabiskan sekitas 223 jam atau setara dengan 28 hari dengan perhitungan waktu 8 jam kerja perharinya. Terdapat perbedaan yang cukup signifikan antara estimasi dan realisasi, dimana perbedaan yang signifikan terjadi pada *backlog* ke-4 untuk rancangan *interface*, dimana untuk estimasi yang direncanakan awalnya 20 jam sedangkan pada realisasinya menyelesaikan sekitar 23 jam hal ini dikarenakan terdapat perubahan-perubahan yang terjadi selama perancangan sehingga hal tersebut memakan banyak waktu, Perubahan lain juga terjadi pada *backlog* ke-11 untuk master kategori yang dimana untuk estimasi waktu yaitu 18 jam dan realisasinya 20 jam hal tersebut terjadi karena masalah saat menampilkan salah satu data dikategori.

B. Pembahasan Hasil

Implementasi *payment gateway* pada aplikasi *wedding organizer* ini menggunakan *midtrans* sebagai pihak ketiganya. *Midtrans* merupakan suatu layanan *payment gateway* yang dapat membantu keperluan pelaku bisnis *online* dengan tersedianya beragam opsi metode pembayaran [15]. Pemanfaatan *payment gateway Midtrans* membantu memudahkan pelanggan dalam melakukan pembayaran online saat memesan produk melalui *e-commerce*. Dengan demikian, *customers* tidak perlu lagi melakukan konfirmasi pembayaran, dan pada saat yang sama, sistem ini mempermudah admin dalam memeriksa status pembayaran [16]. *Midtrans* juga memiliki fitur yang terintegrasi dengan *e-commerce* untuk memenuhi kebutuhan transaksi pembayaran *online*, termasuk penggunaan kartu debit, kartu kredit, serta penarikan dan pengiriman uang tunai [17]. Berikut merupakan gambar yang merupakan cara untuk mengambil data Api dari *midtrans* dan hasil pengimplementasiannya.

```

public function createSnapToken(array $params)
{
    $midtrans\Config::$serverKey = env('MIDTRANS_SERVERKEY');
    // Set to Development/Sandbox Environment (default). Set to true for Production Environment (accept real transaction).
    $midtrans\Config::$isProduction = false;
    // Set sanitization on (default)
    $midtrans\Config::$isSanitized = true;
    // Set 3DS transaction for credit card to true
    $midtrans\Config::$is3ds = true;

    $nama = explode(' ', $this->nama);
    $firstname = @$nama[0];
    $lastname = @$nama[1];

    // setting expiry 1 hari
    $startTime = Carbon::now()->format('Y-m-d H:i:s O');
    $unit = 'day';
    $duration = 1; // 24 jam dalam menit (24 * 60)

    if (@$params['startTime'] != null) {
        $startTime = $params['startTime'];
    }

    if (@$params['unit'] != null) {
        $unit = $params['unit'];
    }

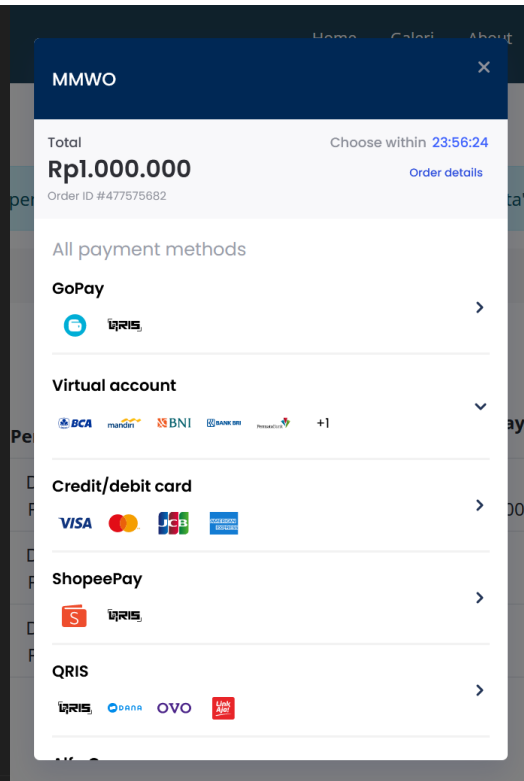
    if (@$params['duration'] != null) {
        $duration = $params['duration'];
    }

    $params = [
        'transaction_details' => [
            'order_id' => $params['order_id'],
            'gross_amount' => $params['gross_amount'],
        ],
        'expiry' => [
            'start_time' => $startTime,
            'unit' => $unit,
            'duration' => $duration,
        ],
        'customer_details' => [
            'id_booking' => @$params['id_booking'],
            'first_name' => $firstname,
            'last_name' => $lastname,
            'email' => $this->email,
            'phone' => $this->no_tlp,
        ],
    ];

    $snapToken = $midtrans::Snap::getSnapToken($params);
    return $snapToken;
}

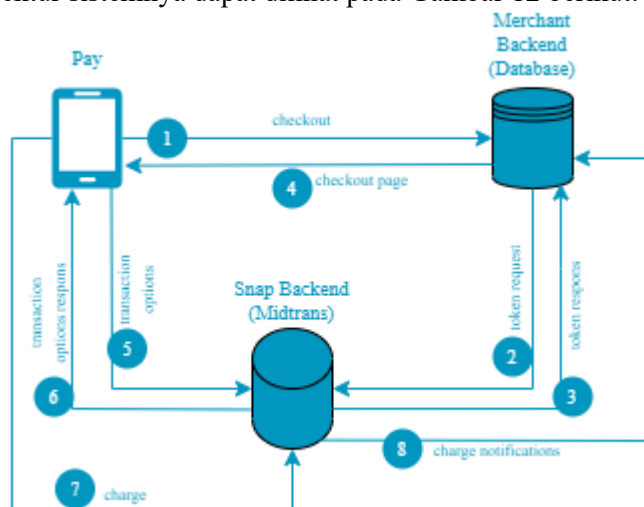
```

Gambar 10: Code Integrasi Midtrans



Gambar 11: Tampilan Payment Gateway

Adapun untuk gambaran arsitektur sistemnya dapat dilihat pada Gambar 12 berikut:



Gambar 12: Arsitektur Sistem

Berdasarkan Gambar 12 dijelaskan mengenai arsitektur sistem yang merupakan alur bagaimana sistem terkoneksi atau memanggil API *midtrans* untuk melakukan transaksi. Proses ini dimulai dengan langkah pertama di mana pelanggan melakukan proses *checkout* di halaman *website*. Setelah itu, langkah kedua melibatkan sistem pemesanan server yang akan mengirimkan permintaan API ke *backend Snap* guna memperoleh *token snap*. *Backend Snap*, yang merupakan bagian dari *platform Midtrans*, akan memberikan respons dengan *token snap* tersebut pada langkah ketiga. Setelahnya, langkah keempat melibatkan *server* sistem pemesanan dalam pembangunan halaman HTML yang dikirimkan kembali ke *browser* untuk menampilkan jendela *pop-up* pembayaran kepada pelanggan. Di tahap ini, *customer* dapat memilih metode pembayaran yang diinginkan dan menekan tombol pembayaran. Setelah pembayaran berhasil dikonfirmasi, langkah kelima melibatkan sistem untuk mengirimkan rincian pembayaran ke *backend Snap* guna diproses.

Respons dari *backend Snap* yang menampilkan status pembayaran merupakan langkah keenam. Jika pembayaran berhasil, langkah ketujuh melibatkan *backend Snap* untuk memberi tahu *server* sistem pemesanan mengenai status pembayaran. Langkah terakhir, yaitu langkah kedelapan, melibatkan perubahan status pemesanan oleh *snap backend* setelah menerima informasi pembayaran yang berasal dari pelanggan ke *Midtrans* [18][5].

IV. KESIMPULAN

Berdasarkan hasil dari penelitian yang dilakukan, maka diperoleh kesimpulan bahwa pembahasan dari tugas akhir ini menghasilkan aplikasi *wedding organizer* berbasis *web* yang menerapkan teknologi *payment gateway* sebagai transaksi pembayaran. Dengan adanya aplikasi ini dapat membantu admin dalam mengelola proses pembayaran tanpa harus verifikasi pembayaran *customer*. Selain itu aplikasi ini dapat memberikan informasi mengenai layanan paket pernikahan yang tersedia sehingga *customer* tidak harus datang langsung ketempat WO. Adapun penulis memiliki saran dalam menghasilkan aplikasi *wedding organizer* berbasis *web* yang menerapkan *payment gateway* ini agar dapat menjadi lebih baik kedepannya dengan menambahkan fitur chat untuk *customer* bisa berkomunikasi secara langsung dengan admin mengenai informasi layanan paket.

DAFTAR PUSTAKA

- [1] R. Anggraeni and I. E. Maulani, "Pengaruh Teknologi Informasi Terhadap Perkembangan Bisnis Modern," *Jurnal Sosial dan Teknologi*, vol. 3, no. 2, pp. 94–98, 2023.
- [2] R. Setiawan, A. Sutedi, S. Mulyawati, and L. Fitriani, "Perancangan dan Implementasi Aplikasi Reservasi Jasa Karunia Tour and Travel," *Teknika*, vol. 11, no. 3, pp. 218–224, 2022.
- [3] L. A. Ridardo, "Rancang Bangun Aplikasi Wedding Organizer Berbasis Website," in *Prosiding Seminar Nasional Teknologi Informasi dan Komunikasi (SENATIK)*, 2022, pp. 489–495.
- [4] Y. E. Nisrina, W. H. N. Putra, and B. T. Hanggara, "Pengembangan E-Commerce Dengan Pemanfaatan Sistem Payment Gateway (Studi Kasus: Wisata Kampung Sapi Adventure)," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, e-ISSN*, pp. 9419–9425, 2019.
- [5] Y. Prasetyo, "Implementasi Layanan Payment Gateway Pada Sistem Informasi Transaksi Pembayaran," *University of Technology Yogyakarta.*, 2020.
- [6] P. S. Alfian and L. Magdalena, "Penerapan Payment Gateway pada Aplikasi Marketplace Waroeng Mahasiswa Menggunakan Midtrans," *J. Inform. Univ. Pamulang*, vol. 5, no. 3, pp. 387–393, 2020.
- [7] R. Wulandari, R. Setiawan, and A. Mulyani, "Perancangan Sistem Informasi Manajemen Wedding Organizer Online Menggunakan Scrum," *Jurnal Algoritma*, vol. 16, no. 2, pp. 139–150, 2019.
- [8] H. Ginanjar and R. Setiawan, "Perancangan Aplikasi Wedding Organizer Berbasis Android," *Jurnal Algoritma*, vol. 17, no. 1, pp. 155–162, 2020.
- [9] D. Gunadi, "Analysis and Design of Web-Based Wedding Planner Applications Case Studies on Resta Wedding Organizers," *bit-Tech*, vol. 5, no. 1, pp. 13–24, 2022.
- [10] T. M. M. Puspitasari and D. Maulina, "Implementasi payment gateway menggunakan midtrans pada marketplace travnesia. com," *Mobile and Forensics*, vol. 1, no. 1, pp. 22–29, 2019.
- [11] M. C. Layton, "Scrum For Dummies," 2015. [Online]. Available: www.allitebooks.com
- [12] A. Andipradana and K. D. Hartomo, "Rancang Bangun Aplikasi Penjualan Online Berbasis Web Menggunakan Metode Scrum," *Jurnal Algoritma*, vol. 18, no. 1, pp. 161–172, 2021.
- [13] D. Kurniadi, R. Setiawan, and G. P. Ginanjar, "Pengembangan Sistem Informasi Monitoring Dan Pelaporan Kinerja Karyawan Perusahaan Menggunakan Balanced Scorecard Dan Scrum," *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 10, no. 1, pp. 366–380, 2023.
- [14] Z. I. Sholihati, "Pengembangan aplikasi tiga-tingkat menggunakan metode scrum pada aplikasi presensi karyawan glints academy," *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, vol. 6, no. 1, pp. 169–176, 2022.
- [15] C. D. Asti, W. H. N. Putra, and W. Purnomo, "Pengembangan Website E-Commerce dengan Pemanfaatan Sistem Payment Gateway Midtrans (Studi Kasus: Butik Rizza Collection)," *Jurnal*

- Pengembangan Teknologi Informasi dan Ilmu Komputer e-ISSN*, vol. 2548, p. 964X, 2021.
- [16] S. Z. Salma, S. Aswamati, and A. M. Tanniewa, "Pengembangan Sistem Informasi Konsultasi Psikologi Online Dengan Api Midtrans Sebagai Payment Gateway," *Jurnal Teknologi dan Sistem Informasi*, vol. 4, no. 2, pp. 127–137, 2023.
- [17] Y. Fatman, N. K. Nafisah, and P. B. J. Pambudi, "Implementasi Payment Gateway dengan Menggunakan Midtrans pada Website UMKM Geberco," *Jurnal KomtekInfo*, pp. 64–72, 2023.
- [18] A. W. Ramadhan, A. Susanto, and G. W. Saraswati, "Implementasi Digital Payment Gateway Midtrans Pada Sistem Agribisnis Di Temanggung (SIADIT)," *J-SAKTI (Jurnal Sains Komputer dan Informatika)*, vol. 7, no. 1, pp. 95–107, 2023.