

BAB 2

TINJAUAN PUSTAKA

2.1 Undangan

Undangan adalah sebuah kartu, poster, atau media lainnya yang berisi pemberitahuan atau ajakan tertulis untuk mengikuti suatu acara yang akan diadakan. undangan acara berfungsi sebagai pemberitahuan sekaligus permintaan kehadiran kepada para tamu. Undangan acara dapat berupa undangan resmi atau tidak resmi, tergantung pada sifat acara yang akan dihadiri. Undangan resmi biasanya digunakan untuk acara-acara yang bersifat formal, seperti rapat, seminar, atau konferensi. Undangan resmi biasanya ditulis dengan bahasa yang formal dan sopan, serta dilengkapi dengan logo atau kop surat instansi yang mengeluarkan undangan. Undangan tidak resmi biasanya digunakan untuk acara-acara yang bersifat informal, seperti pesta, reuni, atau acara keluarga. Undangan tidak resmi biasanya ditulis dengan bahasa yang lebih bebas dan santai, serta dapat menggunakan kartu undangan atau media sosial.

Fungsi utama undangan adalah memberikan informasi detail mengenai acara yang akan diadakan, seperti tanggal, waktu, tempat, dan tujuan dari acara tersebut. Selain itu, undangan juga dapat mencakup rincian lain seperti tema acara, *dress code*, dan agenda kegiatan yang akan dilakukan. Undangan acara biasanya mencerminkan karakteristik dan identitas dari penyelenggara atau tuan rumah acara. Desain dan gaya undangan dapat bervariasi tergantung pada sifat acara, apakah itu formal, semiformal, atau informal.

Dengan perkembangan zaman sekarang yang semakin digital, kini undangan tidak hanya dicetak menggunakan media kertas, tetapi menggunakan media seperti gambar, animasi hingga *video*. Salah satu kelebihan undangan digital ini adalah penyelenggara acara dapat menyampaikan informasi secara cepat dan tepat kepada para tamu undangan. Kemampuan untuk mengirimkan undangan melalui email, pesan teks, atau bahkan media sosial membuatnya lebih efisien dan ramah lingkungan, mengurangi penggunaan kertas dan limbah. Desain undangan *online* juga dapat lebih dinamis, memungkinkan pengguna untuk menyesuaikan tema,

warna, dan elemen desain lainnya sesuai dengan karakter acara yang diinginkan.

2.1.1 RSVP

RSVP adalah singkatan dari bahasa Prancis, yaitu *Répondez s'il vous plaît* yang berarti tolong balas atau silakan konfirmasi. RSVP biasanya terdapat pada undangan acara formal, seperti pernikahan, ulang tahun, atau acara resmi lainnya. Dalam konteks undangan, RSVP biasanya diikuti oleh informasi kontak, seperti nomor telepon atau alamat *email*, agar tamu undangan dapat memberikan konfirmasi apakah akan hadir atau tidak ke acara tersebut [9]. Konfirmasi kehadiran ini penting untuk beberapa alasan. Tujuannya adalah untuk membantu penyelenggara acara agar mengetahui berapa banyak tamu yang akan datang agar bisa mempersiapkan tempat yang cukup, makanan, dan keperluan lain dengan tepat. Jika RSVP tidak dicantumkan, ada kemungkinan jumlah tamu jauh melebihi kapasitas venue yang sudah dipesan.

Jawaban RSVP bisa berupa konfirmasi kehadiran atau penolakan. Tamu yang merencanakan untuk datang diminta untuk memberikan konfirmasi, sementara yang tidak dapat hadir diminta untuk memberikan pemberitahuan sebelum acara berlangsung. RSVP pada undangan sangatlah penting agar acara yang diselenggarakan dapat berlangsung dengan baik sesuai rencana semula dan para tamu yang diundang pun diharapkan dapat membalas undangan dan memberikan konfirmasi hadir atau tidak.

2.2 *Digital Invitation*

Digital invitation atau undangan *digital* adalah bentuk undangan yang dikirimkan melalui media elektronik atau *online*, bukan dalam bentuk fisik cetak. Digital invitation dikirimkan melalui *email*, media sosial, atau *website*. Tujuannya untuk menghemat biaya percetakan dan mendistribusikan undangan dengan lebih mudah dan cepat. Pembuatan *digital invitation* bisa dilakukan dengan berbagai cara seperti membuat desain sendiri menggunakan perangkat lunak desain grafis, menggunakan *template online*, atau memanfaatkan layanan pembuatan undangan digital. Hasil desainnya bisa berupa *file* gambar, *video* animasi, atau bahkan *website*

interaktif.

Salah satu kelebihan utama *digital invitation* adalah kemudahan dalam pendistribusian. Undangan bisa disebarkan dalam hitungan detik kepada puluhan, ratusan, bahkan ribuan tamu hanya dengan menekan tombol kirim. Selain itu, *digital invitation* juga memungkinkan penerima mengonfirmasi kehadirannya melalui fasilitas *RSVP online*. Di samping kepraktisan, *digital invitation* juga memiliki dampak ramah lingkungan karena mengurangi pemakaian kertas. Undangan digital bisa didesain lebih interaktif dan dinamis seperti dengan menambahkan *video*, animasi, atau bahkan mengintegrasikannya dengan peta digital dan GPS. Secara keseluruhan, *digital invitation* merupakan cara modern mengundang tamu yang *efisien*, cepat, ramah lingkungan, dan tentunya sejalan dengan perkembangan teknologi digital masa kini.

2.3 Website

Website atau situs web adalah kumpulan halaman *web* yang saling berhubungan dan memiliki isi konten *digital* seperti teks, gambar, *video* animasi, ataupun gabungan dari semuanya [10]. Halaman web ini ditampilkan dalam sebuah web browser seperti Google Chrome, Mozilla Firefox, atau Safari. *Website* terdiri dari dua elemen utama yaitu *frontend* dan *backend*. *Frontend* adalah bagian yang terlihat oleh pengguna, melibatkan desain tata letak, *elemen visual*, dan *interaktivitas*. Sementara itu, *backend* mencakup server, *database*, dan logika pemrograman yang menjalankan website, mengelola data, dan memastikan fungsi-fungsi yang diperlukan.

Fungsi utama sebuah *website* biasanya dibedakan menjadi beberapa kategori diantaranya adalah *website* informasi yang berisikan konten *blog* atau artikel dengan tujuan memberikan edukasi dan wawasan kepada para penggunanya. Kemudian *website* bisnis dan komersil yang ditujukan untuk kepentingan profit melalui produk atau jasa layanan yang disediakan di dalamnya seperti toko *online*. Lalu *website* media sosial untuk konektivitas antar anggotanya contohnya Facebook dan Twitter. Ada juga *website* berita untuk menyebarkan informasi terkini dari berbagai peristiwa dan masih banyak lagi jenis *website* lainnya. Jadi

fungsi *website* sangatlah beragam bergantung pada kebutuhan dan tujuan pembuatannya.

2.4 Visual Studio Code

Visual Studio Code (VSCode) adalah sebuah aplikasi *editor code* gratis yang dikembangkan oleh perusahaan besar Microsoft. Dirilis pertama kali pada tahun 2015, kini VS Code menjadi salah satu *editor* koding paling populer di kalangan *developer* dan *programmer*. *Text editor* ini dirancang untuk keperluan pengembangan aplikasi dengan beragam fitur yang mempermudah serta mempercepat pekerjaan programmer atau *developer*.

Banyak orang yang mengira bahwa *Visual Studio Code* sama dengan *Visual Studio* tetapi pada faktanya berbeda. *Visual Studio* merupakan *integrated development environment* (IDE) yang membutuhkan spesifikasi yang berat dan biasanya *Visual Studio* digunakan untuk pengembangan aplikasi di platform dekstop pada Microsoft, sementara VS Code lebih ringan dan fokus sebagai *advanced text editor* dengan fitur pengembangan lintas *platform*. VS Code mendukung banyak bahasa pemrograman seperti JavaScript, TypeScript, C++, C#, Python, Go, Java, dan lainnya [11].

Salah satu keunggulan utama dari VS Code adalah ekosistem ekstensi yang sangat banyak dan beragam. Pengguna dapat memperluas fungsionalitas editor dengan menginstal ekstensi yang tersedia melalui *store*. Ekstensi ini mencakup berbagai fitur, seperti *snippet* kode, dukungan bahasa tambahan, pemformatan kode otomatis, *debugger*, integrasi dengan sistem git, dan banyak lagi. Pengguna juga dapat mengembangkan ekstensi kustom mereka sendiri untuk memenuhi kebutuhan pengembangan yang spesifik.

2.5 NuxtJs

Nuxtjs adalah *framework* JavaScript *open-source* yang dibangun di atas Vue.js untuk pengembangan aplikasi. Nuxtjs memudahkan dan mempercepat pengembangan aplikasi web dengan menyediakan fitur-fitur seperti *server-side rendering*, *routing*, dan modulasi komponen secara otomatis.

Ini berarti, pengembang dapat membangun aplikasi tanpa perlu banyak konfigurasi. Misalnya, Nuxt secara otomatis mengatur rute berdasarkan struktur direktori dari proyek Vue.js, memungkinkan pengembang untuk fokus pada logika aplikasi tanpa melakukan pengaturan konfigurasi yang rumit [12].

Salah satu keunggulan Nuxt.js adalah modularitasnya. Setiap fitur ditampung dalam *package* terpisah sehingga *developer* hanya perlu menyertakan fitur yang dibutuhkan saja. Misalnya modul *routing*, modul linting JS, CSS preprocessor, dan lainnya. Pengembang juga bisa membuat modul *custom* sendiri. Nuxt.js menyediakan *scaffolding* otomatis berupa *folder* dan konfigurasi yang siap pakai, antara lain *pages*, *layouts*, *plugins* dan *modules*. Pengembang bisa langsung fokus mengerjakan *pages* tanpa harus setup konfigurasi yang rumit. Hal ini sangat mempercepat waktu *launching website*.

Dengan dukungan penuh terhadap ekosistem Vue.js, Nuxt mempermudah penggunaan dan integrasi plugin serta modul. Pengembang dapat dengan mudah menambahkan fungsionalitas baru ke proyek yang dibuat melalui berbagai *plugin* yang telah tersedia. Secara keseluruhan, Nuxt.js menyediakan struktur proyek yang terorganisir, membantu pengembang untuk lebih cepat dan efisien dalam membangun aplikasi Vue.js. Baik untuk proyek skala kecil maupun besar, Nuxt membantu menyederhanakan pengembangan aplikasi *web* dengan Vue.js.

2.6 Laravel

Laravel adalah sebuah *framework* PHP yang open source dan dirilis di bawah lisensi MIT. Laravel dikenal luas sebagai salah satu *framework* PHP terbaik dan paling populer saat ini untuk pengembangan aplikasi *web* mulai dari proyek skala kecil hingga besar. Laravel dirancang dengan tujuan untuk membantu para pengembang web bekerja secara lebih efisien dan simpel dalam membangun aplikasi PHP.

Laravel dibangun dengan konsep MVC (*Model View Controller*) memudahkan pemisahan logika operasional yang jelas antara bagian *model*, tampilan, maupun *controller*. Selain itu Laravel juga dilengkapi *dependency injection container* dan *routing system* yang kuat namun sederhana. Laravel juga

menyertakan *package manager* bernama *Composer* sehingga memudahkan proses instalasi dan *update* dependensi dan *library* pihak ketiga. Selain itu Laravel juga didukung *RESTful controller* dan *migasi database* yang memudahkan pekerjaan *backend developer* [13].

Salah satu fitur unggulan Laravel adalah konsep *Eloquent ORM (Object-Relational Mapping)*, yang memungkinkan pengembang berinteraksi dengan database menggunakan model objek PHP. Ini tidak hanya mempermudah manipulasi data dalam *database*, tetapi juga menyediakan cara yang intuitif dan bersih untuk melakukan operasi CRUD (*Create, Read, Update, Delete*). Laravel juga memiliki *templating engine* bawaan bernama *Blade* yang memungkinkan *developer PHP* untuk menulis kode yang lebih bersih dan terstruktur dengan baik. *Blade* juga dilengkapi banyak fitur *helper* yang mempercepat kerja *frontend developer*. Secara keseluruhan, Laravel menjadi *framework PHP* pilihan terbaik untuk membangun aplikasi *web enterprise scale* karena kinerja tinggi dan *sustainability* kodenya. Komunitas *developer Laravel* yang besar juga menjamin dukungan dan evolusi *framework* ke depannya.

2.7 Database

Database adalah kumpulan data yang dikelola dan disimpan secara sistematis dan terstruktur. Data ini umumnya dikelola oleh sebuah sistem yang disebut *Database Management System (DBMS)* yakni sebuah perangkat lunak yang memungkinkan pengguna dalam mengakses, menyimpan, mengupdate, dan menghapus data. Struktur dasar dari sebuah *database* adalah tabel. Tabel terdiri dari baris dan kolom, di mana setiap baris mewakili entitas individu dan setiap kolom mewakili atribut atau karakteristik dari entitas tersebut. Entitas dan atribut ini didefinisikan dalam skema *database*, yang mencakup struktur, tipe data, dan hubungan antara tabel [14].

Terdapat beberapa jenis *database*, termasuk *database relasional*, *database NoSQL*, dan *database berbasis grafik*. *Database relasional* menggunakan model tabel dan relasi antar tabel untuk menyimpan data, sementara *database NoSQL* dapat menyimpan data yang lebih *fleksibel* dan tidak memerlukan skema yang ketat.

Database berbasis grafik memodelkan data sebagai grafik dan digunakan untuk skenario yang melibatkan hubungan kompleks.

2.8 PostgreSQL

PostgreSQL adalah sistem basis data relasional (RDBMS) yang bersifat *open source* dan gratis. postgresql dilisensikan di bawah lisensi BSD. postgresql adalah salah satu sistem basis data yang paling populer di dunia, digunakan oleh berbagai perusahaan dan organisasi, mulai dari skala kecil hingga besar. Salah satu fitur unggulan postgresql adalah dukungan penuh terhadap standar SQL. postgresql mendukung sebagian besar fitur SQL yang ditentukan oleh standar ANSI SQL, serta menyediakan ekstensi dan fitur-fitur tambahan yang memperluas fungsionalitasnya. Hal ini memungkinkan pengembang dan pengguna untuk melakukan operasi kompleks, seperti penggabungan, pengelompokan, pengurutan, dan banyak lagi.

Sebagai aplikasi *open-source*, postgresql memiliki komunitas yang aktif dan luas. Ini berarti bahwa postgresql terus diperbarui, ditingkatkan, dan didukung oleh komunitas yang sehat. Keamanan juga menjadi fokus utama dalam pengembangan postgresql, dengan adanya mekanisme keamanan seperti *autentikasi* pengguna, otorisasi akses dan *enkripsi* data [14].

2.9 QR Code 2D

QR Code adalah jenis *barcode* dua dimensi yang cukup populer dan banyak digunakan di berbagai sektor industri maupun kehidupan sehari-hari. Dalam bentuknya yang persegi dengan pola kotak-kotak hitam dan putih yang teratur, *QR Code* mampu menyimpan informasi data dalam jumlah yang jauh lebih besar daripada *barcode 1D* tradisional. Keunggulan utama dari *QR Code* adalah kemampuannya untuk menyimpan berbagai jenis data, mulai dari teks, *URL*, nomor telepon, alamat *email*, hingga informasi kontak yang lengkap. Dapat dilihat pada Gambar 2.1 QR Code 2D.



Gambar 2.1 QR Code 2D

Sumber gambar: <https://www.matthews.com.au/what-is-a-2d-barcode-and-how-does-it-work>

Dalam sejarahnya, *QR Code* pertama kali diperkenalkan di Jepang pada tahun 1994 [15]. Perusahaan *Denso Corporation*, merupakan bagian dari grup Toyota yang pertamakali menggunakan *QR*. Awalnya, penggunaan *QR Code* terbatas pada pelacakan persediaan dalam industri manufaktur suku cadang kendaraan. Namun, seiring berkembangnya teknologi dan kemudahan akses ke perangkat *mobile*, *QR Code* mulai digunakan secara luas dalam berbagai industri, termasuk pemasaran, perbankan, logistik, pariwisata, dan bahkan di dunia seni. *QR Code* dapat ditemukan di berbagai tempat, seperti iklan cetak, poster, kemasan produk, tiket acara, dan banyak lagi.

Adapun cara bagaimana *QR Code* digenerate menggunakan *useQRCode* adalah sebagai berikut:

1. Pertama, fungsi *useQRCode* diimpor dari direktori integrasi *@vueuse/integrations/useQRCode*.
2. Kemudian, *useQRCode* dapat digunakan dengan meneruskan teks yang ingin dikonversi menjadi *QR Code* sebagai argumen. Dalam contoh ini, teks '*text-to-encode*' digunakan. Hasil dari *useQRCode* adalah sebuah *ref* yang berisi data *URL QR Code* yang dihasilkan. Berikut adalah contoh *code* bagaimana *useQRCode* digunakan pada Tabel 2.1 Contoh Import Package Generate QR.

Tabel 2.1 Contoh Import Package Generate QR

```
1 import { ref } from 'vue'
2 import { useQRCode } from '@vueuse/integrations/useQRCode'
3 const text = ref('text-to-encode')
4 const qrcode = useQRCode(text)
```

3. Dalam contoh HTML yang diberikan, QR Code dapat ditampilkan dengan menggunakan elemen `<img>` dan mengikat `src` dengan `qrcode` (ref data URL QR Code). Dapat dilihat pada Tabel 2.2 Contoh Generate QR Code.

Tabel 2.2 Contoh Generate QR Code

```
1 <input v-model="text" type="text" />
2 
```

Dengan cara ini, ketika teks dalam `<input>` berubah, QR Code akan diperbarui secara dinamis.

2.10 Subscription Model Business

Subscription model business, atau bisnis dengan model langganan, adalah suatu strategi pemasaran di mana *customer* membayar secara berkala, biasanya bulanan atau tahunan, untuk mendapatkan akses terus-menerus terhadap produk atau layanan. Model langganan ini telah menjadi populer di berbagai industri, mulai dari konten digital, perangkat lunak, hingga produk fisik, karena memberikan keuntungan stabil dan berkelanjutan bagi penyedia layanan.

Salah satu keuntungan dari *subscription model* bisnis adalah *stabilitas* pendapatan jangka panjang yang dapat diberikan kepada perusahaan. Dibandingkan dengan model penjualan satu kali, di mana *customer* hanya membayar sekali untuk memperoleh produk atau layanan, *subscription model* memungkinkan perusahaan untuk memperoleh pendapatan berulang secara teratur dari customer yang berlangganan. Selain itu, *subscription model* juga memungkinkan perusahaan untuk membangun hubungan jangka panjang dengan *customer* mereka. Dapat dilihat pada Gambar 2.2 Subscription Models Business.



Gambar 2.2 Subscription Models Business

Sumber gambar: <https://bmttoolbox.net/patterns/subscription/>

Dari alur tersebut Perusahaan adalah pihak yang menyediakan produk atau layanan. Perusahaan perlu mengembangkan produk atau layanan yang berkualitas dan sesuai dengan kebutuhan *customer* agar dapat menarik *customer* untuk berlangganan. Produk atau layanan yang ditawarkan oleh perusahaan dapat berupa produk fisik, produk digital, atau layanan. Perusahaan perlu menentukan jenis produk atau layanan yang akan ditawarkan, serta fitur dan manfaat yang akan diberikan kepada *customer*.

Customer memiliki peranan penting karena Pendapatan perusahaan berasal dari pembayaran berkala yang dilakukan oleh *customer*. Perusahaan perlu mengelola pendapatan tersebut dengan baik agar dapat menghasilkan keuntungan. Maka dari itu perusahaan harus menargetkan *customer* yang tepat agar dapat menarik lebih banyak *customer* untuk berlangganan dan menyediakan sistem pembayaran yang mudah dan aman agar pelanggan dapat melakukan pembayaran dengan lancar.

Dari segala keuntungan yang ditawarkan oleh *Subscription Model Business*, Namun ada juga tantangan yang harus dihadapi dalam mengimplementasikan *subscription model* bisnis. Salah satunya adalah mempertahankan kualitas bagi *customer* agar mereka terus berlangganan. Perusahaan juga harus terus berinovasi, memberikan pembaruan yang relevan, dan memberikan layanan *customer* yang memadai untuk memastikan kepuasan dan mempertahankan kepercayaan *customer*.

2.11 Payment Gateway

Sebuah layanan yang digunakan untuk memfasilitasi transaksi antara pemilik usaha dengan *customer* untuk melakukan pembayaran secara *online*. Cara kerja dari *payment gateway* adalah menjadi penghubung antara *merchant online* seperti penyedia jasa, toko online, *platform* digital dengan berbagai *metode* pembayaran seperti kartu kredit, kartu debit, *transfer bank*, *e-wallet* seperti gopay, ovo, atau shoppepay [16]. Penggunaan *payment gateway* biasanya terdapat biaya tambahan. Biaya ini dapat berupa biaya yang sifatnya tetap atau persentase dari jumlah transaksi yang dibayarkan. Pemilik usaha juga harus memahami ketentuan biaya yang dikenakan oleh *payment gateway* karena setiap penyedia jasa *payment*

gateway memiliki potongan biaya yang berbeda-beda. Gambar layanan payment gateway dengan berbagai macam metode pembayaran dapat dilihat pada Gambar 2.3 Payment Gateway Midtrans & Xendit.

	midtrans	xendit
 Cards	   	
 Virtual Accounts	    	
 Retail Outlets	    	
 QRIS	     and more	
 E-Wallets	  	  
 Direct Debit	  	
 Cardless Credit		

Gambar 2.3 Payment Gateway Midtrans & Xendit

Sumber gambar: <https://www.xendit.co/id/alternatif-midtrans/>

Salah satu keuntungan menggunakan *payment gateway* adalah *merchant* atau pemilik usaha tidak perlu tidak lagi mengelola transaksi atau pembayaran lagi pada sistemnya karena dengan menggunakan *payment gateway* semua hal tersebut sudah di atasi secara otomatis, bahkan *Payment gateway* juga menyediakan laporan lengkap semua transaksi. *Merchant* dapat melihat berapa aliran transaksi yang masuk perharinya secara *realtime* melalui grafik dan laporan transaksi yang di sediakan. Selain itu, data sensitif *customer* seperti nomor kartu kredit dan data pribadi lainnya disimpan dengan sangat aman dan biasanya tidak disimpan pada sistem *Merchant*

2.12 API (Application Programming Interface)

API service adalah sebuah layanan yang menyediakan *API (Application Programming Interface)* untuk digunakan oleh pihak lain. *API service* berfungsi seperti perantara bagi pertukaran data antar aplikasi-aplikasi yang berbeda *platform*

maupun bahasa pemrogramannya [17]. Dengan menggunakan API service, sebuah aplikasi dapat meminta informasi dari aplikasi lain, mengirimkan data, atau meminta eksekusi tugas tertentu. API service memungkinkan pengembang untuk memanfaatkan fitur atau layanan dari aplikasi lain tanpa perlu mengetahui detail implementasi internalnya.

Contoh penggunaan API service dalam kehidupan sehari-hari misalnya layanan pemesanan tiket pesawat online. Dengan mengakses API ini maka aplikasi yang dikembangkan bisa mendapatkan fungsi dan data pemesanan tiket meskipun fitur itu awalnya tidak ada di dalam aplikasi kita. API service dapat mengadopsi berbagai bentuk, termasuk RESTful API. RESTful API menggunakan metode HTTP standar seperti GET, POST, PUT, dan DELETE untuk berkomunikasi [18].

Dalam dunia pengembangan perangkat lunak, API service merupakan pondasi penting untuk membangun aplikasi yang terintegrasi dengan baik, memungkinkan ekosistem perangkat lunak yang lebih besar untuk berfungsi secara bersamaan. API service memberikan kemampuan untuk menciptakan aplikasi yang lebih fleksibel, modular, dan dapat berkembang seiring waktu dengan mudah [1]. Sebagai contoh, implementasi REST web service telah digunakan untuk proses sales order dan sales tracking berbasis mobile

2.13 Midtrans

Midtrans adalah perusahaan teknologi finansial (*fintech*) yang menyediakan layanan gerbang pembayaran (*payment gateway*) di Indonesia. Midtrans mendukung beragam metode pembayaran digital, seperti kartu kredit, transfer bank dan e-wallet untuk memfasilitasi transaksi online pelanggan. Dengan demikian, Midtrans membantu bisnis fokus mengembangkan usahanya, bukan repot mengelola pembayaran [5].

Midtrans sudah terintegrasi dengan berbagai *platform chat* dan media sosial seperti WhatsApp, Line, Facebook Messenger, dan lainnya. Jadi penjual tetap bisa menerima pembayaran meski melalui obrolan chat tanpa perlu repot integrate *code program*. Selain itu, Midtrans juga menyediakan layanan *bulk disbursement* yang memungkinkan pengiriman dana sekaligus ke ratusan penerima dalam satu proses.

Fitur ini sangat berguna untuk pembayaran gaji karyawan, *refund*, dan keperluan bisnis lainnya yang melibatkan transfer masal.

2.13.1 Aktivasi Akun Midtrans

Midtrans menyediakan layanan *disbursement* atau pencairan dana bagi para penggunanya. Proses aktivasi *disbursement* Midtrans terdiri dari beberapa tahap diantaranya adalah sebagai berikut:

1. Pengguna diminta untuk mengunggah dokumen-dokumen yang diperlukan untuk mengaktifkan layanan *disbursement*. Dokumen-dokumen ini mencakup NPWP (Nomor Pokok Wajib Pajak) atau *Tax Identification Number* dan Dokumen identitas pemilik bisnis seperti KTP atau KITAS. Dapat dilihat seperti Gambar 2.4 Pengajuan Pencairan Akun.

Gambar 2.4 Pengajuan Pencairan Akun

2. Verifikasi Dokumen Setelah dokumen diunggah, Midtrans akan melakukan verifikasi terhadap dokumen-dokumen tersebut. Proses ini memakan waktu sekitar 3 hari kerja. Dapat dilihat seperti Gambar 2.5 Verifikasi Pencairan Akun.

Gambar 2.5 Verifikasi Pencairan Akun

3. Status Verifikasi Midtrans akan menginformasikan pengguna jika dokumen yang diunggah memerlukan perbaikan atau jika ada dokumen tambahan yang dibutuhkan.
4. Informasi Tambahan Selama proses verifikasi berlangsung, pengguna dapat mempelajari lebih lanjut tentang layanan disbursement melalui tutorial atau FAQ yang disediakan oleh Midtrans.
5. Aktivasi Setelah verifikasi berhasil, layanan disbursement akan diaktifkan dan pengguna dapat mulai menggunakan fitur ini.

2.13.2 Instalasi Midtrans

Cara kerja Midtrans cukup mudah. Pertama, bisnis mendaftarkan diri ke situs Midtrans secara gratis untuk mendapatkan akses ke panel admin. Kemudian, Midtrans akan mengintegrasikan *payment gateway*-nya ke sistem bisnis tersebut, baik website, aplikasi mobile, maupun toko fisik. Setiap ada transaksi pembayaran dari konsumen, uangnya akan masuk ke rekening penjual setelah dipotong biaya jasa Midtrans. Biaya jasa Midtrans sangat kompetitif dan transparan. Tidak ada biaya bulanan atau tahunan. Penjual hanya dikenai biaya kecil di sekitar 2-3% untuk setiap transaksi yang berhasil. Adapun cara bagaimana midtrans di implementasi pada sisi frontend adalah sebagai berikut:

1. Instalasi Midtrans

Buka terminal pada proyek NuxtJS lalu jalankan perintah `npm install midtrans-client` untuk menginstal paket Midtrans.

2. Impor Midtrans

Pada komponen Vue pembayaran yang akan menggunakan Midtrans, impor paket Midtrans dengan cara seperti Tabel 2.3 Contoh Import Package Midtrans.

Tabel 2.3 Contoh Import Package Midtrans

```
1 import midtransClient from 'midtrans-client'
```

3. Inisialisasi Midtrans

Lakukan inisialisasi Midtrans dengan konfigurasi seperti server key, client key, (sandbox atau produksi) seperti Tabel 2.4 Contoh Code Menampilkan Midtrans

Snap.

Tabel 2.4 Contoh Code Config Midtrans Snap

```
1 const midtrans = midtransClient.Snap({
2   isProduction: process.env.NODE_ENV === 'production',
3   serverKey: 'YOUR_SERVER_KEY',
4   clientKey: 'YOUR_CLIENT_KEY'
5 })
```

4. Buat Order pada Backend

Sebelum memulai pembayaran, frontend harus membuat order pada backend order ini akan menghasilkan token yang akan digunakan untuk memulai pembayaran di frontend.

5. Tampilkan Halaman Pembayaran Midtrans

Pada komponen Vue tempat ingin menampilkan pembayaran, panggil metode *midtrans.createTransaction* dengan menyertakan token dari order:

Tabel 2.5 Contoh Code Create Transaction Snap

```
1 midtrans.createTransaction(orderToken)
2   .then(transactionToken => {
3     window.snap.pay(transactionToken)
4   }) .catch(error => {
5     console.error(error)
6   })
```

2.14 Whatsapp Auto Reply

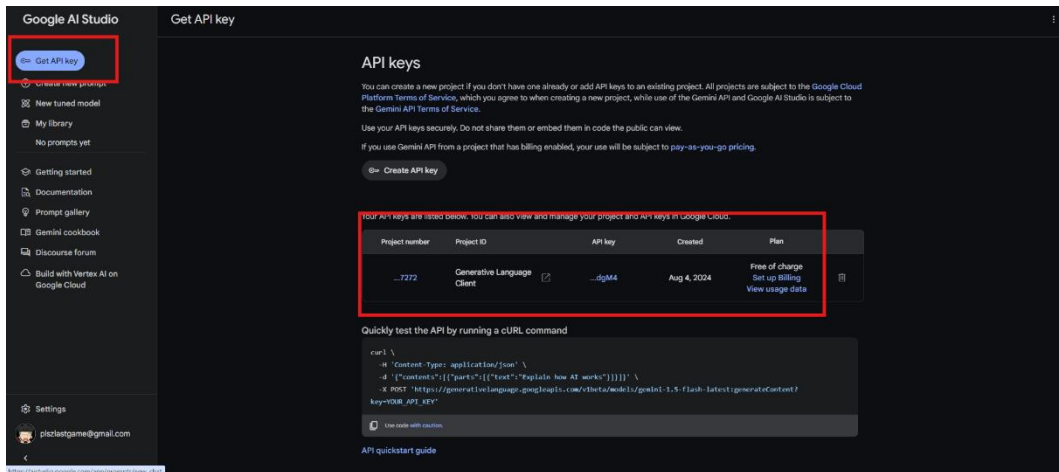
Whatsapp *Auto Reply* adalah sebuah sistem yang memungkinkan *bot* atau aplikasi untuk merespons secara otomatis terhadap pesan yang diterima melalui *platform* messaging WhatsApp menggunakan bahasa pemrograman Node.js. Salah satu keunggulan utama dari Whatsapp *Auto Reply* adalah kemampuannya untuk memberikan *respons* secara cepat dan konsisten [19]. Hal ini dapat sangat membantu dalam situasi di mana respons cepat sangat diperlukan, seperti dalam layanan pelanggan, sistem informasi, atau layanan bantuan darurat.

Selain itu, Whatsapp *Auto Reply* juga dapat diintegrasikan dengan berbagai layanan atau API lainnya, seperti basis data, layanan *cloud*, atau layanan pihak ketiga lainnya. Hal ini membuka peluang untuk membangun sistem yang lebih

canggih dan fungsional, seperti sistem yang dapat menyimpan riwayat percakapan, mengirimkan notifikasi, atau bahkan mengintegrasikan dengan layanan lain seperti pembayaran atau pemesanan.

2.14.1 Api Gemini AI

Gemini AI adalah sebuah teknologi kecerdasan buatan (AI) yang dikembangkan oleh Google, *platform* ini dibuat untuk mempermudah pengguna dalam mengembangkan dan menerapkan aplikasi AI. Dengan menggunakan teknologi *machine learning* dan *deep learning*, Gemini AI dapat memahami dan mengolah bahasa alami dengan lebih baik, sehingga memungkinkan pengguna untuk berinteraksi dengan sistem secara lebih baik. Berikut cara mendapatkan api key Gemini AI dapat dilihat pada Gambar 2.6 Api Key Gemini AI.



Gambar 2.6 Api Key Gemini AI

Berikut Cara mendapatkan API key Gemini berdasarkan Gambar 2.6 Api Key Gemini AI sebagai berikut:

1. Buka Google AI Studio
2. Klik tombol "*Create API key*" yang terlihat di tengah halaman atau disebelah kiri atas.
3. Setelah *API key* dibuat, informasinya akan muncul dalam tabel di bawah, menampilkan Project number, Project ID, API key, tanggal pembuatan, dan rencana penggunaan.
4. API key yang dibuat dapat digunakan untuk mengakses layanan Gemini AI.

Penting untuk menjaga kerahasiaan API key dan tidak membagikannya kepada publik untuk menghindari penyalahgunaan.

2.14.2 Instalasi Whatsapp Auto Reply

Implementasi Whatsapp Auto Reply dapat dilakukan dengan menggunakan berbagai bahasa pemrograman dan framework yang tersedia, seperti Node.js, Python, atau bahkan Java. Namun, salah satu yang sering digunakan adalah Node.js karena kemampuannya dalam menangani banyak koneksi secara bersamaan (*concurrent connections*) dengan efisien, serta kemudahan dalam mengintegrasikan dengan layanan lain.

Adapun cara bagaimana whatsapp auto reply di implementasi dengan menggunakan wppconnect dan gemini ai adalah sebagai berikut:

1. Instal Dependensi

Instal package `@wppconnect-team/wppconnect` dan `gemini-ai` menggunakan npm atau yarn.

2. Instal Dependensi

Impor modul `@wppconnect-team/wppconnect` dan `gemini-ai` di modul JavaScript seperti Tabel 2.6 Contoh Import Package WppConnect & GeminiAI.

Tabel 2.6 Contoh Import Package WppConnect & GeminiAI

```
1 const wppconnect = require('@wppconnect-team/wppconnect');
2 const GeminiAI = require('gemini-ai');
```

3. Instal Gemini AI

Buat instance dari Gemini AI dengan menggunakan API key yang telah dibuat pada website resmi google. Dapat dilihat pada Tabel 2.7 Contoh Create Instance GeminiAI.

Tabel 2.7 Contoh Create Instance GeminiAI

```
1 const gemini = new GeminiAI('YOUR_API_KEY');
```

4. Inisialisasi WhatsApp Web

Inisialisasi koneksi dengan WhatsApp Web menggunakan wppconnect. Dapat dilihat pada Tabel 2.8 Contoh Inisialisasi Whatsapp Web.

Tabel 2.8 Contoh Inisialisasi Whatsapp Web

```
1 wppconnect.create().then((client) => startBot(client));
```

5. Fungsi Memulai Bot

Buat fungsi `startBot` yang akan menangani pesan masuk dari WhatsApp. Dapat dilihat pada Tabel 2.9 Contoh `onMessage` Whatsapp Service.

Tabel 2.9 Contoh `onMessage` Whatsapp Service

```
1 function startBot(client) {
2   client.onMessage(async (message) => {
3     const response = await
4       gemini.sendMessage(message.body);
5     client.sendText(message.from, response.message);
6   });
7 }
```

Dengan mengikuti langkah-langkah ini, code program telah berhasil mengintegrasikan `@wppconnect-team/wppconnect` dan Gemini AI dalam aplikasi JavaScript. Bot WhatsApp pada javascript sekarang dapat menerima pesan, mengirimkannya ke Gemini AI untuk diproses, dan kemudian mengirimkan respons kembali ke pengirim pesan di WhatsApp.

2.15 GrapesJS

GrapesJS adalah sebuah *framework open-source* berbasis web yang digunakan untuk membangun *editor website builder*. GrapesJS memungkinkan pengguna untuk membuat dan mengubah tampilan antarmuka pengguna secara visual tanpa perlu menulis kode HTML, CSS, atau JavaScript secara manual [20]. GrapesJS bekerja dengan menggunakan pendekatan *drag-and-drop* dan komponen-komponen yang dapat diubah sesuai keinginan. Salah satu keunggulan utama GrapesJS adalah kemampuannya untuk mengekspor *source code* yang dihasilkan dari editor visual ke dalam format HTML, CSS, dan JavaScript. Ini memungkinkan pengguna untuk mengambil kode sumber yang dihasilkan dan mengintegrasikannya ke dalam aplikasi web dengan mudah.

Selain *editor visual*, GrapesJS juga menyediakan *code editor* yang memungkinkan pengguna untuk memodifikasi kode HTML, CSS, dan JavaScript

secara langsung. Fitur ini sangat berguna bagi *developer web* yang ingin mengubah tampilan dan fungsionalitas *web* secara lebih lanjut. Adapun cara bagaimana grapesjs di implementasi pada nuxt3 adalah sebagai berikut:

1. Instal Dependensi

Instal package grapesjs menggunakan npm atau yarn.

2. Daftarkan Plugin GrapesJS

Buka file `nuxt.config.js` di direktori proyek Nuxt. Di dalam objek `buildModules`, tambahkan `'nuxt-grapesjs'`. Dapat dilihat pada Tabel 2.10 Contoh Regist Plugin GrapesJS.

Tabel 2.10 Contoh Regist Plugin GrapesJS

```
1 export default {
2   buildModules: ['nuxt-grapesjs']
3 }
```

3. Gunakan GrapesJS di Komponen Vue seperti pada Tabel 2.11 Contoh Inisiasi GrapesJS pada Nuxt.

Tabel 2.11 Contoh Inisiasi GrapesJS pada Nuxt

```
1 <template>
2   <div>
3     <!-- GrapesJS editor will be rendered here -->
4   </div>
5 </ template>
6
7 <script>
8   import grapesjs from 'grapesjs';
9
10  export default {
11    mounted() {
12      const editor = grapesjs.init({
13        //...
14      });
15    },
16  }
17 </script>
```

Dengan mengikuti langkah-langkah di atas, code telah berhasil mengintegrasikan GrapesJS ke dalam proyek Nuxtjs. Setelah itu dapat melanjutkan dengan mengonfigurasi GrapesJS sesuai kebutuhan, menambahkan plugin atau

blok kode, dan menggunakannya di komponen Nuxt.

2.16 UML

Unified Modeling Language (UML) adalah metode pemodelan yang standar dan banyak digunakan untuk menggambarkan dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis OO(*Object-Oriented*) [21]. UML menyediakan beragam jenis diagram dan notasi visual untuk memodelkan berbagai aspek sistem perangkat lunak, mulai dari requirement, analisis dan desain, hingga implementasi dan pengujian.

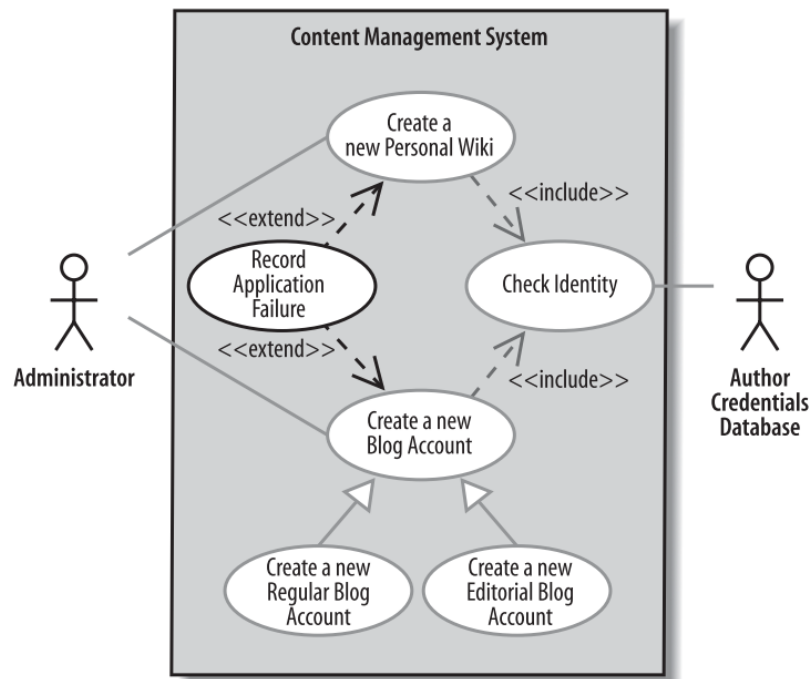
Kelebihan penggunaan UML antara lain adalah memudahkan komunikasi antar berbagai pihak yang terlibat dalam pengembangan sistem karena menyediakan pandangan visual yang standar yang bisa dimengerti oleh semua orang, memungkinkan analisis sistem dilakukan lebih awal karena dapat dimodelkan sebelum dibuat, serta mendukung sistem berorientasi objek. Beberapa diagram UML yang umum digunakan antara lain *use case* diagram untuk memodelkan fungsionalitas sistem dari perspektif pengguna, *class* diagram untuk memodelkan struktur dan hubungan antar kelas dalam sistem, *activity* diagram untuk memodelkan perilaku dan alur kerja sistem, *sequence* diagram untuk memodelkan interaksi antar objek berdasarkan urutan waktu, dan banyak lagi diagram lainnya.

2.17 Use Case Diagram

Use case diagram adalah salah satu jenis diagram UML yang menggambarkan hubungan interaksi antara aktor dan sistem. Diagram ini digunakan untuk menggambarkan apa yang dapat dilakukan oleh pengguna terhadap sistem, serta bagaimana sistem merespons tindakan tersebut. Use Case dapat mendeskripsikan tipe interaksi antara sipengguna sistem dengan sistemnya [21]. Aktor merepresentasikan sebuah peran yang dimainkan oleh user yang berinteraksi dengan sistem yang dibuat. Sedangkan use case merepresentasikan sebuah tujuan yang ingin dicapai aktor dari interaksinya dengan sistem. Semakin banyak use case yang dimiliki suatu sistem, semakin banyak juga fungsionalitas yang ditawarkan

sistem tersebut.

Beberapa komponen penting dalam use case diagram selain use case dan aktor adalah *relationships* dan *system boundary*. Relationships menunjukkan interaksi antara aktor dan use case, misalnya *association relationship*. Sedangkan *system boundary* membatasi lingkup sistem mana yang sedang dibuat use case diagramnya. Dengan memahami use case diagram, pengguna dan pengembang sistem bisa melihat fungsionalitas utama yang ada tanpa harus tahu detail bagaimana sistem bekerja, sehingga sangat membantu dalam memahami analisis kebutuhan sistem. Contoh use case diagram dapat dilihat pada Gambar 2.7 Use Case Diagram.



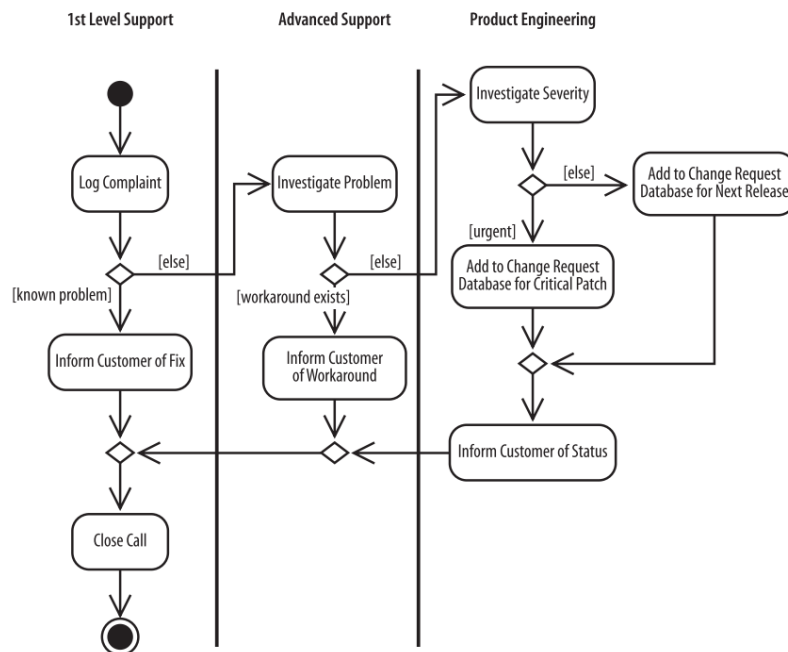
Gambar 2.7 Use Case Diagram

Sumber: *Learning UML 2.0* [21]

2.18 Activity Diagram

Activity diagram merupakan salah satu diagram dalam UML yang digunakan untuk memodelkan perilaku atau alur aktivitas dari sebuah sistem yang sedang dirancang. Pada setiap use case yang ada, maka terdapat paling sedikit satu activity diagram. Diagram ini menggambarkan proses bisnis dan urutan aktivitas dalam

sebuah proses [21]. Diagram ini menggambarkan rangkaian aktivitas yang dilakukan sistem atau proses bisnis dari awal hingga akhir. Berikut merupakan contoh penerapan activity diagram diagram dapat dilihat pada Gambar 2.8 Activity Diagram.



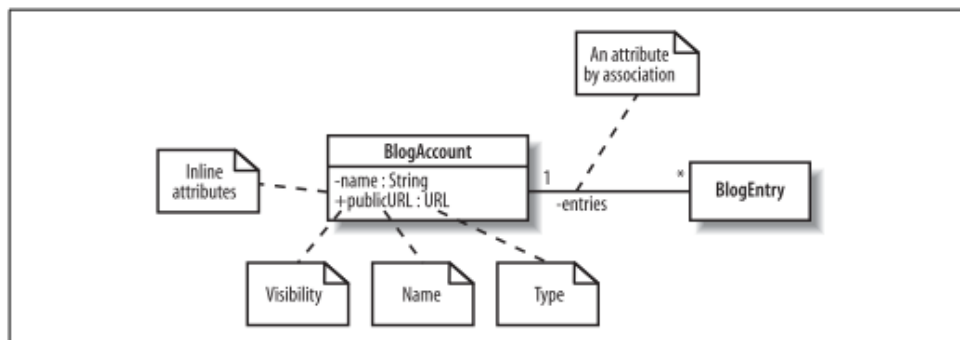
Gambar 2.8 Activity Diagram

Sumber: Learning UML 2.0 [21]

2.19 Class Diagram

Class diagram adalah diagram yang digunakan untuk menggambarkan dengan jelas struktur serta deskripsi class, atribut, metode, dan hubungan dari setiap objek [21]. Dalam istilah sederhana, Class Diagram membantu pengembang dalam memodelkan bagaimana berbagai kelas objek atau entitas dalam sistem berhubungan satu sama lain dan berinteraksi untuk mencapai tujuan tertentu.

Setiap kelas diwakili oleh suatu kotak yang dibagi menjadi tiga bagian. Bagian atas berisi nama kelas, bagian tengah berisi atribut-atribut atau properti yang dimiliki kelas tersebut, dan bagian bawah berisi metode atau operasi yang dapat dilakukan oleh kelas. Garis-garis menghubungkan kelas-kelas ini untuk menunjukkan hubungan dan ketergantungan di antara mereka. Contoh class diagram diagram dapat dilihat pada Gambar 2.9 Class Diagram.



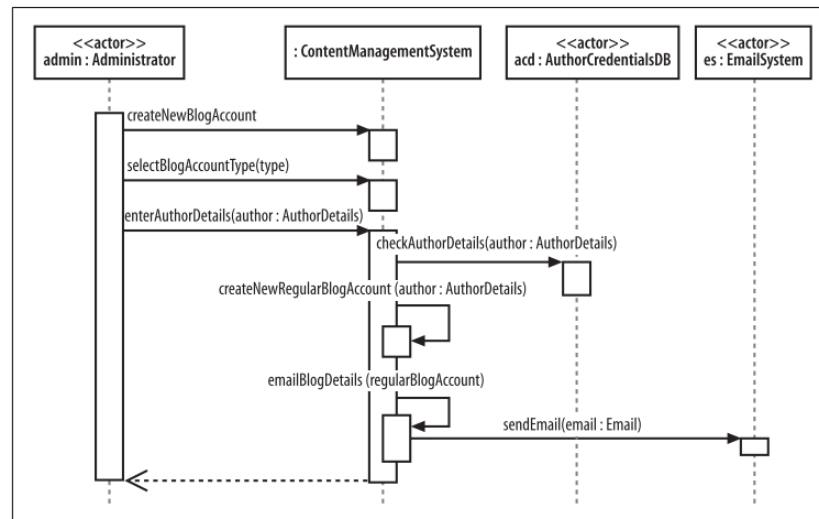
Gambar 2.9 Class Diagram

Sumber: *Learning UML 2.0* [21]

2.20 Sequence Diagram

Sequence diagram adalah diagram yang digunakan untuk menggambarkan dan menjelaskan interaksi antar objek-objek dalam sebuah sistem secara terperinci. Dengan kata lain, *Sequence* Diagram membantu dalam memodelkan bagaimana objek atau kelas berkomunikasi satu sama lain dan melibatkan proses yang terjadi dari waktu ke waktu [21].

Diagram *sequence* terdiri dari dua bagian, yaitu bagian vertikal yang menunjukkan waktu dan bagian *horizontal* yang menunjukkan objek-objek. *Lifeline* merupakan garis vertikal yang mewakili dan mengidentifikasi sebuah objek dan keberadaannya sepanjang eksekusi sistem. *Message* menunjukkan komunikasi antar objek berupa pengiriman *event*, sinyal, atau panggilan metode. Urutan *vertikal* message menunjukkan urutan kejadian dalam waktu. Contoh *sequence* diagram dapat dilihat pada Gambar 2.10 Sequence Diagram.



Gambar 2.10 Sequence Diagram

Sumber: Learning UML 2.0 [21]