

BAB 2

LANDASAN TEORI

2.1 Transportasi Bus

2.1.1 Pengertian Transportasi

Memindahkan, menggerakkan, mengangkut, atau mengalihkan suatu objek dari satu tempat ke tempat lain sehingga objek tersebut lebih bermanfaat atau dapat digunakan untuk tujuan tertentu disebut transportasi. Transportasi adalah pergerakan manusia dan/atau barang dari tempat yang satu ke tempat lain, pergerakan timbul karena adanya aktivitas di dalam Masyarakat[8]. Terdapat 5 unsur pokok transportasi, yaitu :

- a) Manusia, yang membutuhkan transportasi.
- b) Barang, yang diperlukan manusia.
- c) Kendaraan, sebagai sarana transportasi.
- d) Jalan, sebagai prasarana transportasi.
- e) Organisasi, sebagai pengelola transportasi.

Secara umum, ada dua kelompok besar moda transportasi, yaitu :

1.) Kendaraan Pribadi yaitu gaya transportasi yang dibuat khusus untuk seseorang sehingga mereka dapat pergi ke mana saja dan kapan saja mereka mau, bahkan jika mereka tidak memakainya sama sekali (seperti jika mobil disimpan di garasi). Contoh kendaraan pribadi seperti :

- a) Sepeda pribadi
- b) Sepeda motor pribadi
- c) Mobil pribadi

2.) Kendaraan Umum yaitu jenis transportasi yang dimaksudkan untuk digunakan oleh orang banyak, untuk kepentingan bersama, untuk memberikan layanan, dan untuk memiliki rute dan titik tujuan yang sama. Transportasi umum juga terikat dengan peraturan trayek dan jadwal yang sudah ditentukan. Contoh kendaraan umum seperti :

- a) Bus (kota dan antar kota).

- b) Kereta API.
- c) Pesawat terbang.
- d) Kapal laut.

2.1.2 Pengertian Bus

Bus adalah sarana transportasi perkotaan yang dapat memudahkan perjalanan orang banyak. Bus adalah kendaraan darat yang dapat menampung hingga 300 orang[9]. Bus memiliki berbagai macam jenis, yaitu :

- 1) Motorcoach adalah kendaraan yang dibuat untuk bepergian lebih jauh dari bus biasa, dan memiliki kursi yang lebih nyaman, ruang untuk bagasi, dan mesin yang lebih besar. Kendaraan ini memiliki AC, kamar mandi, dan sistem audio/video. Coach atau motorcoach adalah kendaraan yang serbaguna yang dapat digunakan untuk perjalanan wisata, liburan, atau antar kota.
- 2) Bus tingkat dirancang dengan dua lantai agar dapat memuat lebih banyak penumpang.

2.2 Aplikasi

Aplikasi merupakan alat atau sarana yang digunakan sebagai perantara untuk menyampaikan informasi[10]. Jenis aplikasi adalah kategori aplikasi berdasarkan cara pengembangan, penggunaan, atau fungsinya. Beberapa jenis aplikasi yang umum adalah :

1) Aplikasi Desktop

Aplikasi berbasis desktop adalah aplikasi yang berjalan local di desktop dan hanya dapat digunakan oleh pengguna desktop. aplikasi desktop berbeda dengan aplikasi web yang dapat diakses dari manapun melalui jaringan. Aplikasi berbasis desktop membantu pengguna komputer atau laptop mengelola data dan menghasilkan laporan yang cepat, tepat, dan akurat. aplikasi desktop adalah aplikasi yang berjalan di lingkungan desktop dan digunakan pada pengguna desktop untuk mempermudah bagi pengguna komputer untuk mengeluarkan laporan[11].

2) Aplikasi Web

Aplikasi web adalah aplikasi yang dapat diakses melalui jaringan internet atau intranet melalui mesin pencari (browser). Aplikasi web juga merupakan perangkat lunak atau software yang menggunakan Bahasa pemrograman seperti javascript, php, dan lainnya. Selain itu, aplikasi web biasanya tidak membutuhkan banyak memori perangkat[12]. Pengguna dapat mengakses basis data pribadi dari berbagai perangkat termasuk komputer, tablet, dan ponsel yang terhubung ke internet karena semua basis data disimpan di server. Pengguna tidak perlu mengunduh atau menginstall aplikasi khusus, sehingga lebih mudah dijangkau oleh banyak orang. Oleh karena itu, pengalaman pengguna akan buruk jika menggunakan aplikasi web yang memiliki koneksi yang buruk.

Salah satu keuntungan dari aplikasi web adalah bahwa dapat diakses melalui berbagai platform dan disajikan dalam web browser pada sistem operasi mana saja[12].

3) Aplikasi Mobile

Aplikasi mobile adalah jenis perangkat lunak aplikasi yang dirancang untuk dapat berjalan pada perangkat seluler, seperti smartphone dan juga tablet. Aplikasi mobile terdiri dari tiga kategori :

- a) Aplikasi *Native* adalah aplikasi yang dikembangkan khusus untuk platform tertentu dengan menggunakan teknologi pengembangan dan *Application Programming Interface (API)*.
- b) Aplikasi Web adalah aplikasi yang dibuat dengan pengembang web seperti HTML, CSS, dan JavaScript seperti sebuah website tetapi dirancang untuk diakses melalui browser smartphone.
- c) Aplikasi *Hybrid* adalah kombinasi aplikasi native dan web yang dibuat dengan Bahasa pemrograman web seperti HTML, CSS, dan JavaScript. Dirancang untuk berfungsi di berbagai platform perangkat seluler. Aplikasi ini dapat mengakses hardware seperti kamera, GPS, dan lainnya. Metode pengembang hybrid yang menggabungkan teknologi

web dengan aplikasi native memungkinkan untuk mengoptimalkan kelebihan kedua teknologi[13].

2.3 Konsep Dasar Pemesanan

Konsumen melakukan pemesanan sebelum membeli dan Perusahaan harus memiliki sistem pemesanan yang baik untuk memastikan bahwa konsumen puas. Menurut Kamus Besar Bahasa Indonesia, “Pemesanan adalah proses, perbuatan, cara memesan (tempat, barang, dan sebagainya) kepada orang lain”. Pengguna memilih tiket bus dengan rute dan jadwal tertentu dan harus memberikan informasi pribadi seperti nama, nomor telepon, dan email. Informasi ini diperlukan untuk data penumpang bus.

2.4 Website

Website adalah kumpulan halaman web yang berisi informasi data digital, seperti teks, gambar, animasi, suara, dan video yang disediakan melalui jalur internet, sehingga setiap orang di seluruh dunia dapat mengaksesnya dan melihatnya. Dalam kebanyakan kasus, sebuah situs web terdiri dari setidaknya satu server web, yang dapat diakses melalui jaringan seperti Internet atau jaringan wilayah local (LAN) melalui penggunaan URL. Gabungan semua situs web yang dapat diakses public di internet disebut World Wide Web (WWW). Halaman web dibuat menggunakan Bahasa standar yang dikenal sebagai HTML. Web browser menerjemahkan script HTML ini sehingga informasi dapat ditampilkan dalam bentuk yang dapat dibaca oleh semua orang dan untuk mengaksesnya menggunakan perangkat lunak yang disebut browser[14].

Hypertext Markup Language (HTML) adalah Bahasa standar yang digunakan untuk menampilkan halaman web[15]. Yang bisa dilakukan dengan HTML yaitu:

1. Mengatur tampilan dari halaman *web* dan isinya.
2. Membuat tabel dalam halaman *web*.
3. Mempublikasikan halaman *web* secara online.
4. Membuat *form* yang bisa digunakan untuk registrasi dan transaksi via *web*.

5. Menambahkan objek-objek seperti citra, audio, video, dan animasi dalam web.
6. Menampilkan area gambar di *browser*.

Didalam sebuah website, ada beberapa komponen pendukung agar sebuah *website* dapat diakses oleh pemakai, diantaranya :

1) *Internet*

Internet, sebagai jaringan komunikasi yang saling terhubung, memberi dampak positif bagi seluruh Masyarakat pengguna internet. Namun, ada dampak negative dari penggunaan internet yang salah, yang jelas merugikan.

Internet adalah jaringan yang memungkinkan banyak komputer terhubung satu sama lain untuk mengirim berita, memperoleh informasi, atau mentransfer data. Selain itu, internet dapat digunakan sebagai media pembelajaran untuk mengakses informasi yang bersifat edukatif. Dianggap sebagai media pembelajaran yang paling baru, efektif, mudah, dan terpercaya untuk memenuhi kebutuhan informasi khususnya tentang pembelajaran tanpa terkendala oleh ruang dan waktu[16].

2) *Web Browser*

Web Browser adalah aplikasi yang digunakan untuk menampilkan halaman web beserta kontennya. Beberapa aplikasi browser yang dapat digunakan antara lain Firefox, Chrome, dan Opera[16].

3) *Web Server*

Web Server berfungsi sebagai inti dari *World Wide Web* (WWW). Jika seseorang menggunakan browser seperti Internet Explorer, Mozilla, Chrome atau program browser lainnya, *web server* akan memproses permintaan tersebut dan kemudian mengembalikan data yang diinginkan ke browser tersebut[17].

2.5 **Application Programming Interface (API)**

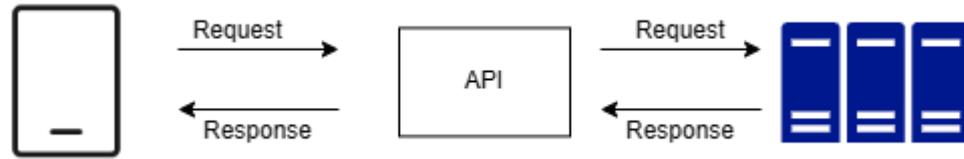
API adalah antarmuka yang memungkinkan pengembang mengakses fungsi aplikasi atau layanan dari program. Ini memungkinkan pengembang untuk menggunakan fungsi yang sudah ada di dalam aplikasi lain tanpa perlu membuat

ulang program dari awal. API untuk website adalah pemanggilan fungsi melalui Hyper Text Transfer Protocol (HTTP) dan mendapatkan respon berupa Extensible Markup Language (XML) atau JavaScript Object Notation (JSON)[18].

API digunakan untuk berbagi data antar aplikasi. Tujuan lain dari API adalah untuk mempercepat proses pengembangan aplikasi dengan memberikan fitur khusus sehingga pembangun tidak perlu lagi merancang fitur yang sama. API yang beroperasi pada Tingkat sistem operasi membantu aplikasi berkomunikasi dengan layer dasar dan berkomunikasi satu sama lain sesuai dengan berbagai protokol yang disesuaikan[18]. API terbagi menjadi empat jenis, yaitu :

1. Open API, adalah antarmuka pemrograman untuk aplikasi sumber terbuka yang dapat diakses melalui protocol HTTP. API public, atau titik akhir API, memiliki format permintaan dan respons yang telah ditentukan.
2. Partner API, pengembang dapat menghubungkan mitra bisnis strategis dengan partner API melalui portal pengembang API public, tetapi pengembang harus menyelesaikan proses orientasi dan mendapatkan kredensial login sebelum dapat mengakses API mitra.
3. Internal API, internal API tetap tersembunyi dari pengguna eksternal. API privat ini hanya tersedia untuk tim pengembangan internal yang bekerja sama untuk meningkatkan komunikasi dan produktivitas.
4. Composite API, composite API menggabungkan beberapa API data atau layanan. API ini memungkinkan programmer untuk mengakses beberapa titik akhir dalam satu panggilan, Composite API berguna dalam arsitektur layanan mikro dimana untuk melakukan satu tugas mungkin memerlukan informasi dari beberapa sumber.

API biasanya berfungsi sebagai sumber data untuk kebutuhan sistem atau aplikasi tertentu. Ini memungkinkan data di transfer antar aplikasi. Ilustrasi dari cara kerja API ditampilkan pada Gambar 2.1.



Gambar 2.1 Cara Kerja API

Berikut adalah penjelasan mengenai ilustrasi dari cara kerja API :

- 1) Aplikasi Mengakses API.
- 2) API melakukan request ke server.
- 3) Server memberi respon ke API.
- 4) API menyampaikan respon ke aplikasi.

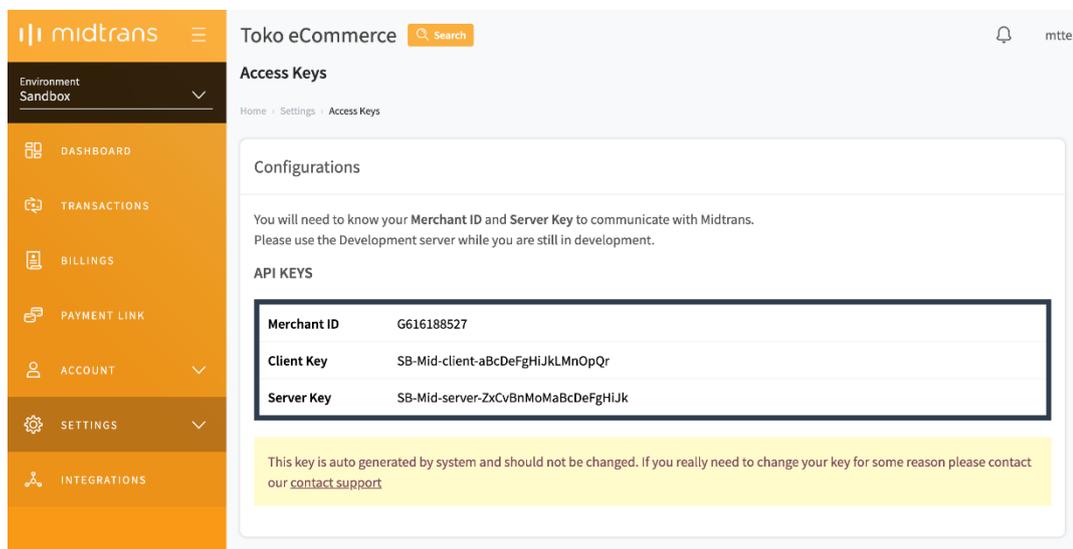
2.5.1 Midtrans API

Pembayaran Elektronik, juga dikenal sebagai E-Payment adalah metode pembayaran non-tonai yang dilakukan melalui koneksi internet yang memungkinkan pelanggan membayar melalui sistem pihak ketiga. Sistematika ini memproses, memverifikasi, dan memvalidasi pembayaran kartu kredit atas nama penjual. Saat ini, orang dapat melakukan transaksi tanpa uang tunak seperti *e-wallet*, bank transfer, direct debit, over the counter, dan jenis transaksi lainnya dengan tingkat keamanan yang lebih tinggi, bahkan menawarkan sarana pembayaran gateway internasional seperti www.midtrans.com. Dengan menggunakan Midtrans dapat membuat pembayaran lebih mudah dengan memasukkan kode transaksi ke situs website penjualan. Selain itu, midtrans memiliki banyak fasilitas yang menawarkan banyak pilihan pembayaran. Pelayanan ini memungkinkan para pelaku industry lebih mudah beroperasi dan meningkatkan penjualan[19].

Midtrans bergerak di bidang finansial berbasis teknologi yang menawarkan Solusi *Payment Gateway* di Indonesia. Mendukung banyak metode pembayaran *online* dan kirim dana (*disbursement*) untuk memudahkan pelanggan lakukan transaksi bisnis. Dengan midtrans terdapat beberapa metode pembayaran seperti GoPay, QRIS, *Virtual Account*, kartu debit atau kredit, ShopeePay, dan

tunai di gerai seperti Indomaret dan Alfamart. Kemudahan lainnya dalam menggunakan *payment gateway midtrans* adalah semua sistem telah diotomisasi. Langkah-langkah untuk menggunakan Midtrans adalah sebagai berikut :

1. Membuat akun midtrans pada tautan <https://account.midtrans.com/register>
2. Setelah menyelesaikan pendaftaran, masuk ke Merchant Administration Portal (MAP) di mana dapat mengakses Sandbox dan *Production Environment*.
3. Untuk mendapatkan akses ke Client Key dan Server Key dapat pergi ke bagian **Setting > Access Keys**.



Gambar 2.2 Access Key Midtrans

Sumber: (<https://docs.midtrans.com/docs/midtrans-account>)

4. Untuk mendapatkan snap token install library Midtrans di server, kemudian konfigurasi *Client Key* dan *Server Key* yang diperoleh dari dashboard Midtrans.
5. Buat transaksi dengan parameter seperti jumlah pembayaran dan detail pelanggan, kemudian kirim transaksi ke API untuk mendapatkan token pembayaran.
6. Token pembayaran yang didapatkan dari API Midtrans adalah Snap Token. Gunakan snap token untuk menampilkan formulir pembayaran Midtrans.

Contoh penggunaan Snap Token:

```

1. const midtrans = require('midtrans-client');
2. // Set API Key dan Server Key
3. const serverKey = 'YOUR_SERVER_KEY';
4. const clientKey = 'YOUR_CLIENT_KEY';
5. // Buat object transaksi
6. const transaction = {
7.   amount: 100000, // Jumlah pembayaran dalam Rupiah
8.   payment_type: 'gopay', // Metode pembayaran
9.   customer_details: {
10.    name: 'John Doe',
11.    email: 'johndoe@example.com',
12.    phone_number: '08123456789',
13.   },
14. };
15. // Buat object request
16. const request = {
17.   server_key: serverKey,
18.   transaction: transaction,
19. };
20. // Dapatkan Snap Token
21. midtrans.generateToken(request)
22.   .then((response) => {
23.     const snapToken = response.token;
24.     console.log('Snap Token:', snapToken);
25.     // Gunakan Snap Token untuk menampilkan formulir pembayaran Midtrans
26.     // ...
27.   })
28.   .catch((error) => {
29.     console.error('Error:', error);
30.   });

```

2.5.2 Mapbox API

Mapbox adalah sebuah platform pemetaan open source yang bekerja dan merilis sebagai kode sebanyak mungkin. Sebagian besar data Mapbox menggunakan bantuan serta berinvestasi pada berbagai macam sumber data misalnya OpenStreetMap, USGS, Landsat, dan OpenAddress[20]. Mapbox mendukung banyak aplikasi yang akan digunakan oleh penggunanya, baik mobile maupun online, termasuk peta, satelit, geocoding, server atlas, dan banyak lagi. Mapbox telah diintegrasikan ke dalam aplikasi Foursquare, Pinterest, dan Evernote untuk membantu pengguna menandai lokasi mereka kapan pun dan dimana pun mereka mau. Pengguna yang mendaftar di Mapbox memiliki akses ke berbagai macam data dengan berbagai pilihan biaya, yang jelas memengaruhi berbagai fasilitas yang dapat mereka akses[21].

Mapbox merupakan aplikasi yang menawarkan alat yang dapat diintegrasikan ke dalam aplikasi. Banyak pengembang telah memasukkan peta ke

dalam aplikasi mereka untuk meningkatkan pengalaman pengguna, peta ini harus selalu terhubung ke internet untuk melihat perubahan lokasi setempat. Salah satu keunggulan Mapbox adalah kemampuan untuk membangun peta menggunakan perangkat lunak open source dan memberikan lisensi kepada pengembang untuk dimasukkan ke dalam aplikasi[22].

2.5.3 Geolocation API

Geolocation API merupakan sebuah fitur yang dapat mengidentifikasi letak geografis dari sebuah objek, seperti ponsel atau terminal komputer yang terhubung ke internet. Posisi terkait erat dengan lokasi, yang menekankan penentuan lokasi seperti Alamat, jalan, dan arah, yang bukan hanya koordinat geografis. Dalam sistem geolocation, dapat diketahui melalui *Internet Protocol* (IP), Alamat *MAC*, *RFID*, keterangan pada hardware atau nomor produksi, *embedded software number*, faktur, atau perangkat GPS dapat digunakan untuk mengetahui lokasi geografis[23].

Geolocation bekerja secara otomatis mencari Alamat IP melalui layanan WHOIS dan mengumpulkan Alamat fisik pendaftar. Data alamat IP geolocation dapat mencakup informasi seperti negara, Kawasan, kota, kode pos, koordinat lintang, koordinat bujur, serta zona waktu. Dengan kata lain geolocation adalah cara untuk menemukan lokasi di seluruh dunia. Geolocation menggunakan data latitude dan longitude yang dimiliki komputer atau telepon seluler untuk menemukan lokasi. Metode ini termasuk menggunakan Alamat IP, sambungan nirkabel atau BtS, dan GPS khusus atau tertanam pada telepon seluler[23].

2.6 Global Positioning System (GPS)

GPS adalah singkatan dari *Global Positioning System*, yang merupakan sistem navigasi dengan menggunakan teknologi satelit yang dapat menerima sinyal dari satelit. Sistem ini menggunakan 24 satelit yang mengirimkan sinyal gelombang mikro ke Bumi. Alat penerima di permukaan, dimana GPS penerima ini akan mengumpulkan informasi dari satelit GPS. Penerima GPS harus mengunci sinyal minimal 3 satelit untuk mengetahui posisi 2d (latitude dan

longitude) dan rute pergerakan. Jika penerima GPS dapat menerima sinyal dari empat atau lebih satelit, maka penerima dapat menghitung posisi 3d (latitude, longitude, dan altitude). GPS dapat menghitung informasi tambahan seperti kecepatan, arah, jalur, tujuan perjalanan, jarak, matahari terbenam, dll. Setelah mengidentifikasi posisi pengguna. Waktu perjalanan akan dihitung dengan menggunakan sinyal satelit ke GPS. Waktu kedatangan (TOA) adalah istilah lain untuk waktu perjalanan ini[24].

Sistem ini awalnya dikembangkan oleh departemen pertahanan Amerika. Nama asli dari sistem GPS, yang awalnya dimaksudkan untuk sistem waktu satelit dan rangking global, terdiri dari tiga segmen, yaitu satelit (Space Segment), pengendali (Control Segment), dan penerima/pengguna (User Segment). Ada 24 satelit GPS di Bumi, 21 diantaranya aktif bekerja, 3 sisanya adalah Cadangan. Tugas segmen pengendali termasuk mengirimkan data satelit, menentukan dan memprediksi orbit satelit, dan menyinkronkan waktunya. Sedangkan segmen penerima bertanggung jawab untuk mengumpulkan dan memproses data satelit untuk menentukan posisi, arah, jarak, dan waktu yang dibutuhkan pengguna[25].

Arsitektur dari sistem GPS disetujui oleh Departemen Pertahanan Amerika Serikat pada tahun 1973. Sistem GPS resmi dimulai beroperasi pada tahun 1994, setelah satelit pertama diluncurkan pada tahun 1978. Satelit GPS secara terus menerus mengirimkan sinyal radio digital kepada penerima yang berhubungan, yang mengandung data waktu dan lokasi satelit. Jam atom pada satelit GPS memiliki ketepatan waktu satu per satu juta detik. Semakin lama sinyal membutuhkan waktu untuk sampai ke penerima di bumi, berarti semakin jauh posisi satelit dari stasiun penerima, sehingga stasiun penerima dapat mengetahui waktu yang dibutuhkan untuk mengirim sinyal[26].

Alat yang disebut GPS receiver diperlukan untuk mengetahui posisi seseorang. Ini menerima sinyal dari satelit GPS. Way-point adalah titik-titik koordinat lintang dan bujur dari posisi seseorang, yang kemudian ditampilkan di layar peta elektronik. Selama melihat langit, GPS dapat membantu menemukan arah di mana pun berada. Data yang dikirim dari satelit berupa sinyal radio dan

data digital dikirim ke alat yang disebut GPS receiver. Alat GPS receiver berbentuk modul dan menghasilkan data NMEA yang berisi data posisi. Modul GPS receiver sendiri terdiri dari beberapa integrated circuit (IC)[26].

GPS Tracker, atau sering disebut sebagai GPS Tracking, adalah teknologi AVL (Automated Vehicle Locator) yang memungkinkan pengguna untuk melacak posisi kendaraan, armada, dan mobil secara real-time. Ini melakukannya dengan menggunakan kombinasi teknologi GSM dan GPS untuk menentukan lokasi sebuah objek, dan kemudian mengubah lokasi tersebut menjadi peta digital[26].

2.7 Database

Database adalah suatu kumpulan data yang berhubungan secara logika dan secara deskripsi dari data-data yang dirancang untuk memenuhi kebutuhan informasi dalam suatu organisasi. Karena *database* didefinisikan secara terpisah dari program aplikasi yang menggunakannya, database menawarkan keuntungan dalam penyimpanan data dengan format yang independen dan fleksibel. Ruang lingkup database dapat diubah tanpa memperngaruhi program aplikasi yang menggunakannya[27]. Fungsi database adalah untuk menyimpan, mengelola, dan mengorganisasi data sehingga orang dapat mengaksesnya dengan mudah, efisien, aman dan terstruktur untuk data.

Perangkat lunak yang biasa digunakan dalam *database* (basis data) yaitu *PostgreSQL*. *PostgreSQL* merupakan *software* dari *Object Relational Database Management System* (ORDBMS) yang dapat diakses secara gratis[28]. Konsep sistem basis data dapat dijelaskan sebagai berikut :

- a.) Dalam bukunya “Database Organization”, James Martin mendefinisikan basis data sebagai kumpulan data yang saling terhubung yang disimpan Bersama dalam suatu media tanpa tumpeng tindih atau struktur data khusus, yang membuatnya lebih mudah untk digunakan dan diambil kapan saja.
- b.) Dalam buku “Database System for Management”, James F. Courtney Jr. dan David B. Paradiance menggambarkan sistem basis data sebagai

kumpulan basis data yang dirancang dan dikelola dengan Teknik perancangan dan pengelolaan basis data yang didukung oleh teknologi komputer.

Para ahli umumnya setuju bahwa database adalah kumpulan data yang terstruktur, saling berhubungan, dan disimpan dalam media penyimpanan komputer[29].

2.8 PostgreSQL

PostgreSQL merupakan sebuah *Object Relational Database Management System* (ORDBMS) yang memiliki fitur khas dari DBMS tradisional, tetapi memiliki beberapa perbaikan untuk kinerja dan fungsionalitas. Basis data untuk SIG *PostgreSQL* adalah salah satu DBMS yang digunakan untuk menyimpan data yang bersifat open source. Kekurangan *PostgreSQL* adalah kurangnya popularitas dan belum tersedianya replikasi. Sedangkan kelebihanya yaitu memiliki kecepatan loading, tipe data geometri, ketersediaan semua fitur database, kemampuan konektivitas dengan database lain, rute, kemampuan menampung data spasial, dan kemudahan perluasannya[30].

Penulis memilih *PostgreSQL* karena terpercaya dan stabil, merupakan objek relasional, menangani persaingan dengan lebih baik. *PostgreSQL* juga mendukung berbagai Bahasa pemrograman, termasuk C++, PHP, Java, Javascript dan lain sebagainya.

2.9 Visual Studio Code

Visual Studio Code (*VsCode*) adalah text editor ringan dan kuat yang dibuat oleh *Microsoft* untuk sistem operasi multiplatform, artinya juga tersedia untuk versi Linux, Mac, dan Windows. Text editor ini mendukung Bahasa pemrograman JavaScript, TypeScript, dan Node.js serta Bahasa pemrograman lainnya dengan bantuan plugin yang dapat diinstal melalui marketplace *VsCode*. Adapun kelebihan dari *VsCode* diantaranya seperti teks yang bersifat terbuka dan sumber kode yang memiliki link Github yang sangat mudah dibaca. Oleh karena itu *VsCode* disukai oleh para developer dalam membuat codingan[31].

Editor ini mencakup semua fitur lingkungan pengembangan terpadu (IDE) yang dimaksudkan untuk pengembang yang bekerja dengan teknologi *cloud* yang terbuka *Microsoft*. Didasarkan pada gaya explorer umum, antarmuka ini mudah digunakan dan memiliki panel di sebelah kiri yang menampilkan semua file dan folder. Di sebelah kanan juga dapat mengakses panel editor, yang menampilkan isi dari file yang telah dibuka. Dalam hal ini, editor telah dikembangkan dengan baik dan menarik. Selain itu, ia memiliki fungsi yang baik, dengan *intellisense* dan *autocomplete* yang bekerja dengan baik untuk *JSON*, *CSS*, *HTML*, dan *Node.JS*.

2.10 JavaScript Object Notation (JSON)

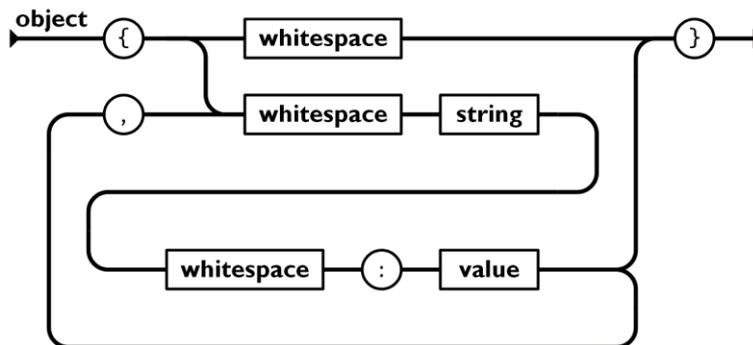
JSON merupakan singkatan dari *JavaScript Object Notation*. JSON adalah format pertukaran data yang ringan, mudah dibaca, dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer, dipakai oleh banyak sistem dan sepakat untuk digunakan komunikasi data[32]. Format ini dibuat berdasarkan bagian dari Bahasa pemrograman JavaScript, standar ECMA-262 Edisi ke-3 Desember 1999. Karena JSON menggunakan gaya Bahasa yang umum digunakan oleh programmer keluarga C, seperti C++, C#, Java, JavaScript, Perl, Python, dan lain-lain. Format teks ini tidak bergantung pada Bahasa pemrograman apa pun. Oleh karena sifat-sifat tersebut, JSON menjadi ideal sebagai Bahasa pertukaran data. JSON terbuat dari dua struktur, yaitu :

- 1) Kumpulan pasangan nama/nilai. Pada beberapa Bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), kamus (*dictionary*), tabel hash (*hash table*), atau *associative array*.
- 2) Daftar nilai terurutkan (*anordered list of values*). Pada kebanyakan Bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Struktur-struktur data ini disebut sebagai struktur data universal. Semua Bahasa pemrograman modern mendukung struktur data ini dalam bentuk yang sama atau berbeda. Hal ini masuk akal karena format data dapat dengan mudah

ditukar dengan Bahasa pemrograman yang juga berdasarkan struktur data ini. JSON menggunakan bentuk sebagai berikut :

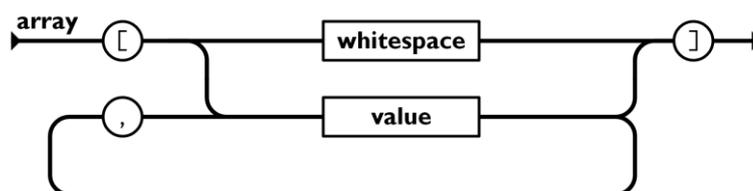
- a.) Objek adalah sepasang nama/nilai yang tidak terurutkan. Objek dimulai dengan *kuurng kurawal buka* “ { ” dan diakhiri dengan *kuurng kurawal tutup* “ } ” setiap nama diikuti dengan *titik dua* “ : ” dan setiap pasangan nama/nilai dipisahkan oleh *koma* “ , ”.



Gambar 2.3 Objek Pada JSON

Sumber : (JSON)

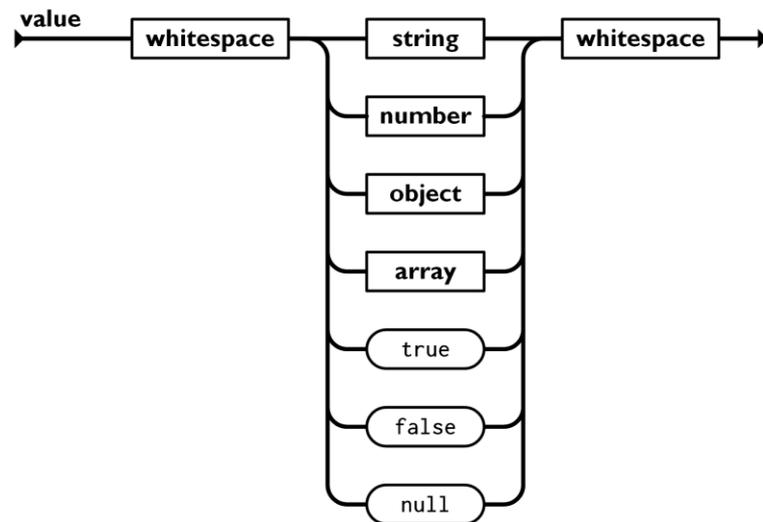
- b.) Larik adalah kumpulan nilai yang terurutkan. Larik dimulai dengan *kuurng kotak buka* “ [” dan diakhiri dengan *kuurng kotak tutup* “] ”. setiap nilai dipisahkan oleh *koma* “ , ”.



Gambar 2.4 Larik Pada JSON

Sumber : (JSON)

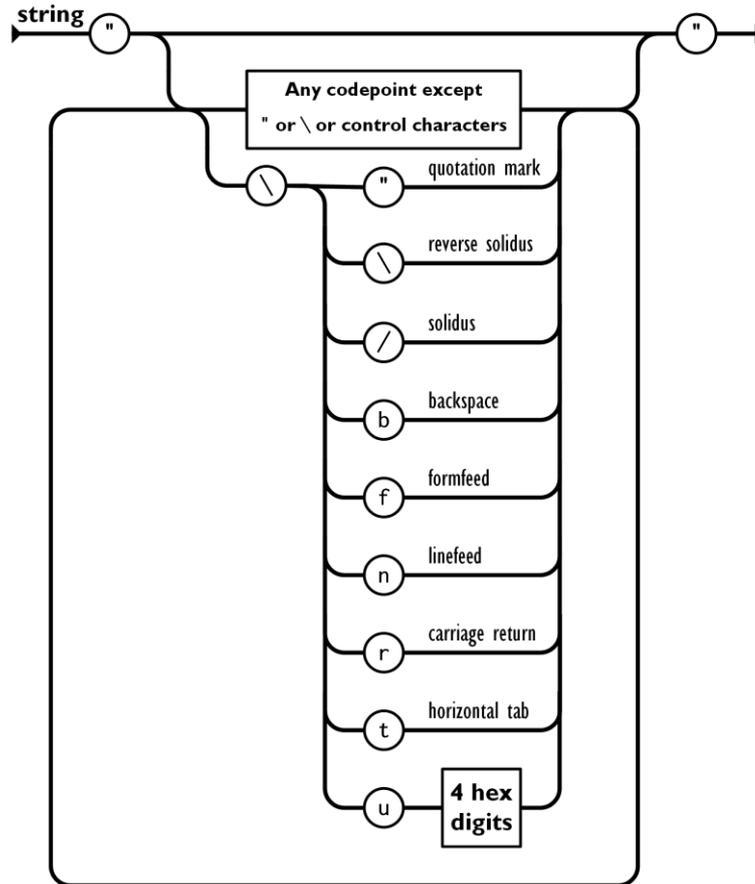
- c.) Nilai (*value*) dapat berupa sebuah string dalam tanda kutip ganda, atau angka, atau *true*, atau *false* atau *null*, atau sebuah *objek* atau sebuah *larik*.



Gambar 2.5 Nilai Pada JSON

Sumber: (JSON)

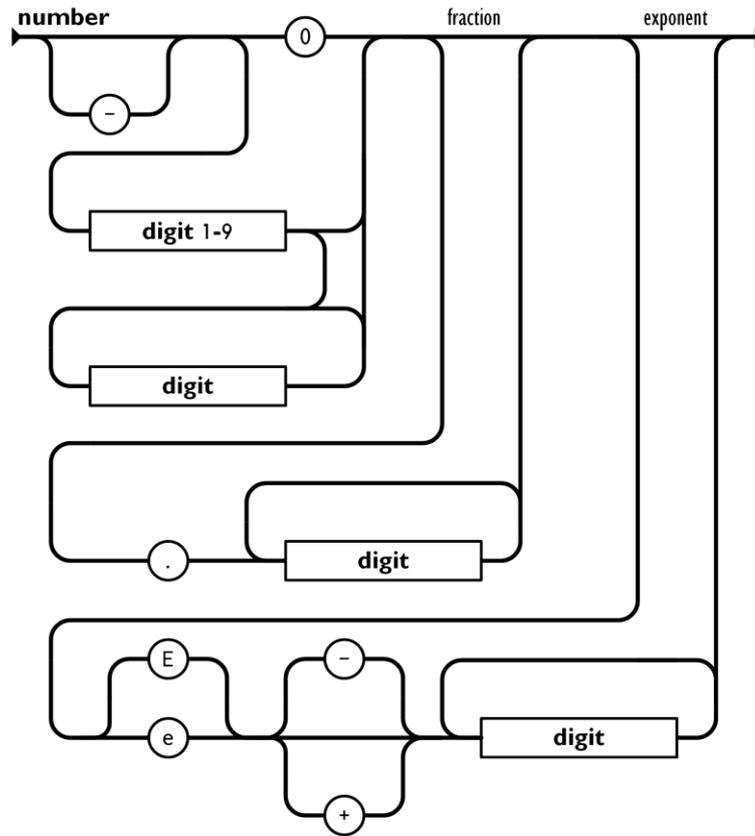
- d.) String adalah kumpulan dari nol atau lebih karakter Unicode, yang dibungkus dengan tanda kutip ganda. Di dalam string dapat digunakan *backslash escape* “ \ ” untuk membentuk karakter khusus. Sebuah karakter mewakili karakter tunggal pada string.



Gambar 2.6 String Pada JSON

Sumber : (JSON)

- e.) Angka pada JSON sangat mirip dengan angka di C atau Java, kecuali oktal dan heksadesimal tidak digunakan



Gambar 2.7 Angka Pada JSON

Sumber : (JSON)

Spasi kosong (*whitespace*) dapat disisipkan di antara pasangan tanda-tanda tersebut, kecuali beberapa detail *encoding* yang secara lengkap dipaparkan oleh Bahasa pemrograman yang bersangkutan.

2.11 Express JS

Express JS merupakan *framework* aplikasi web Node JS yang fleksibel dan minimal, serta menyediakan fitur yang kuat untuk web dan mobile. Express JS adalah sebuah modul *framework* pada Node JS untuk mempermudah menyediakan HTML web pada *file* tersendiri dengan format “.ejs”. Express.js digunakan dalam aplikasi ini untuk mengelola routing, middleware, dan respon HTTP, memungkinkan pengembang untuk fokus pada logika bisnis tanpa harus mengelola kompleksitas server HTTP secara langsung[33].

Express JS adalah sebuah kerangka kerja aplikasi web Node.JS yang minimal dan fleksibel yang menyediakan serangkaian fitur yang kuat untuk aplikasi web dan mobile[34]. Framework ini dibuat untuk berjalan dalam mesin V8 Chrome, membuatnya berjalan beriringan dengan Node sehingga dapat melakukan kerja dengan sangat cepat.

Framework Express JS menggunakan proses paradigma kerja yang disebut dengan *event-driven programming*. Paradigma ini berpengaruh pada flow sistem yang telah dibuat dengan membuat semua fungsi berjalan berdasarkan *event*. Semua *event* akan terus diawasi oleh code yang disebut *event handler* yang bekerja dengan cara mendeteksi *event* aktif kemudian memanggil *handler*. Kebanyakan *handler* memiliki tipe sebagai *function* atau *method*. Seluruh proses tersebut tidak terjadi secara otomatis. Terdapat sebuah proses yang terjadi secara bersamaan sehingga browser akan diharuskan untuk menunggu. Multi proses yang terjadi di server ini dihandle oleh *Node* dengan menggunakan *Node Event Loop*.

Node menyediakan *Node Event Loop* sebagai bagian dari Bahasa pemrograman miliknya. Ketika node dipanggil, *programmer* tidak perlu membuat *code* yang berfungsi memanggil fungsi lain untuk memulai sebuah *loop*. Express JS akan langsung memulai *loop* otomatis begitu Node berjalan. Proses *loop* ini akan terus menerus berjalan hingga programmer membuat *code* berisi kondisi untuk program agar berhenti. Berikut adalah komponen yang dapat digunakan untuk menjalankan *loop* dalam Express JS :

1. *Controllers*, Mendefinisikan *route handler* aplikasi dan logika bisnis.
2. *Middleware*, Digunakan untuk menginterpretasikan semua permintaan yang datang sebelum diproses oleh *route handler*.
3. *Models*, Merupakan perantara antara *controllers* dan database.
4. *Public*, Tempat menyimpan gambar, javascript dan css file.
5. *Views*, Didalamnya terdapat tempat yang akan diproses oleh server
6. *index.js*, Merupakan file utama yang dijalankan pertama kali Ketika aplikasi dijalankan.

7. *Package.JSON*, Mengelola package yang diinstall melalui *Node Package Manager* (NPM), perintah menjalankan aplikasi, serta versi proyek.

2.12 Next JS

Next.js merupakan kerangka kerja fleksibel yang dapat digunakan untuk membuat aplikasi web dengan cepat. Next.js membutuhkan React sebagai library JavaScript untuk membuat antarmuka pengguna (*user interface*) yang interaktif. Dengan menggunakan framework Next.js, beberapa hal dapat dilakukan. Misalnya, rendering halaman web dilakukan di sisi server, atau server side rendering. Ini berarti halaman web ditampilkan lebih cepat dalam browser web, dan proses setup dan deployment proyek menjadi lebih mudah[35]. Berikut beberapa fitur utama Next JS yaitu :

1. *Routing*, Router berbasis sistem file yang dibangun di atas Komponen Server yang mendukung tata *layouts, nested routing, loading states, error handling*.
2. *Rendering*, *Rendering* sisi klien dan sisi server dengan komponen klien dan server. Dioptimalkan lebih lanjut dengan rendering statis dan dinamis di server dengan Next JS.
3. *Data fetching*, Pengambilan data yang disederhanakan dengan *async/await* di komponen server, dan pengambilan API yang diperluas untuk mengoptimalkan permintaan, penyimpanan data, dan validasi ulang.
4. *Styling*, Dukungan untuk metode *styling*, termasuk Modul CSS, Tailwind CSS, dan CSS-in-JS.
5. *Optimizations*, Pengoptimalan gambar, font, dan skrip untuk meningkatkan vitalitas web inti dan pengalaman pengguna aplikasi.
6. *TypeScript*, Dukungan yang lebih baik untuk *typescript*, dengan pemeriksaan tipe yang lebih baik dan kompilasi yang lebih efisien, serta plugin *typescript* khusus.

Next JS memiliki dua router yang berbeda, yaitu *app router* dan *pages router*. *App router* adalah router yang lebih baru yang memungkinkan anda untuk menggunakan fitur-fitur terbaru React, seperti komponen server. *Pages router* adalah router Next JS yang asli, yang memungkinkan untuk membangun aplikasi

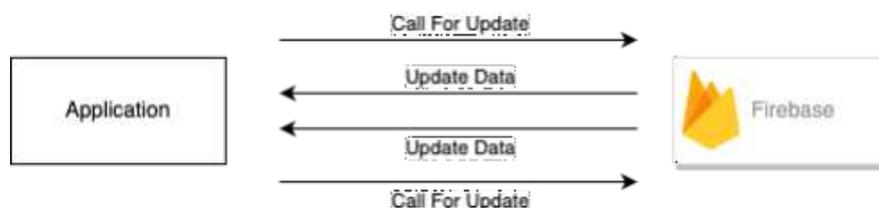
React yang dirender di server dan terus didukung untuk aplikasi Next JS yang lebih lama[36].

2.13 Firebase

Firebase adalah penyedia layanan cloud yang memiliki back-end dan berbasis di San Fransisco, California. Firebase menghasilkan sejumlah produk yang dapat digunakan untuk mengembangkan aplikasi mobile dan web. Firebase pertama kali dibuat oleh Andrew Lee pada tahun 2011 dan dirilis dengan database cloud real-time pada tahun 2012[37].

Firebase adalah aplikasi yang menawarkan layanan database dan backend realtime dan memungkinkan pengembang membuat API dan menyimpannya di cloud. Firebase memiliki banyak library yang memungkinkan pengembang mengintegrasikan layanan dengan Android, Ios, Javascript, Java, dan Node.Js. Selain itu, database Firebase dapat diakses melalui REST API, yang menggunakan protocol Server-Sent Event untuk menghubungkan HTTP untuk menerima pemberitahuan push dari server. Pengembang dapat memposting data kemudian melalui Firebase client library, yang telah dipasang pada aplikasi yang dibangun yang dapat mengumpulkan data secara real-time[37].

Firebase Realtime Database merupakan database yang dapat memperbaharui secara otomatis data yang telah dibaca ketika terjadi perubahan data yang terjadi pada database, sehingga tidak perlu lagi melakukan pemanggilan berulang agar mendapatkan data terbaru yang ada didalam database[38]. Proses kueri yang digunakan dalam firebase menggunakan ilustrasi pada Gambar 2.8



Gambar 2.8 Proses Kueri Firebase Realtime Database

Sumber : (Muhammad Imbalo Zaki Hasibuan, 2022)

Teknologi Firebase telah diterapkan pada banyak aplikasi yang sudah populer di masyarakat. Salah satunya adalah aplikasi E-Tourism, yang memberikan informasi tentang lokasi wisata secara real-time, serta letak titik koordinat dan jalur yang harus ditempuh pengguna untuk mencapai lokasi wisata yang diinginkan[38].

2.14 Unified Modeling Language (UML)

UML merupakan Bahasa pemodelan standar untuk pengembangan perangkat lunak dan sistem. Tujuan pemodelan adalah untuk mengelola kompleksitas sistem. UML memiliki notasi (cara mengekspresikan model) dan meta-model (deskripsi arti dari notasi). Ini yang membuat UML menjadi Bahasa pemodelan. Pendekatan UML untuk mendefinisikan, memverifikasi, dan memvalidasi proses organisasi. Pendekatan ini dapat juga diterapkan pada lingkungan pembelajaran proses untuk mahasiswa dan anggota proyek[39].

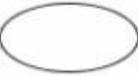
Pada perkembangan Teknik pemrograman berorientasi objek, muncullah sebuah standarisasi Bahasa pemodelan untuk Pembangunan perangkat lunak yang dibangun dengan menggunakan Teknik pemrograman berorientasi objek, yaitu *Unified Modeling Language* atau yang lebih dikenal dengan UML digunakan sebagai salah satu alat untuk melakukan perancangan atau memodelkan sistem. Karena sifatnya yang tidak mempengaruhi perangkat lunak, perangkat keras, sistem operasi, jaringan, basis data, dan bahasa pemrograman yang digunakan, UML sering digunakan[40]. UML muncul karena adanya kebutuhan pemodelan *visual* untuk menspesifikasikan, menggambarkan membangun dan dokumentasi dari sistem perangkat lunak.

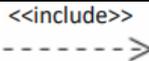
2.14.1 Use Case Diagram

Use case diagram adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. Use case bekerja dengan cara mendeskripsikan tipikal interaksi antara *user* (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan Langkah-langkah yang menerangkan anatara pengguna dan sistem disebut scenario. Setiap scenario mendeskripsikan urutan kejadian. Setiap urutan siinisialisasi oleh orang, sistem

yang lain, perangkat keras atau urutan waktu. Dengan demikian secara singkat bisa dikatakan use case adalah serangkaian scenario yang digabungkan Bersama-sama oleh tujuan umum pengguna. Use case diagram digunakan untuk menangkap aspek dinamis dari sistem secara lebih spesifik, use case diagram digunakan untuk mengumpulkan kebutuhan dari sebuah sistem baik karena pengaruh internal maupun eksternal[41]. Pada Tabel 2.1 menunjukkan symbol yang digunakan.

Tabel 2.1 Simbol Use Case Diagram

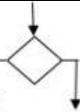
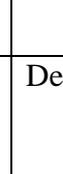
No.	Simbol	Nama	Keterangan
1		Actor	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi itu sendiri.
2		System	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
3		Use Case	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor
4		Collaboration	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
5		Note	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi
6		Depedency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (independent) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak

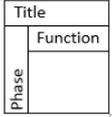
			mandiri (independent).
7		Generalization	Hubungan generalisasi dan spesialisasi (umum-khusus) antar dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya.
8		Association	Komunikasi antar aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
9		Include	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya
10		Extend	Relasi use case tambahan ke sebuah use case, dimana use case yang ditambahkan dapat berdiri sendiri.

2.14.2 Activity Diagram

Activity Diagram adalah bagian penting dari UML yang menggambarkan aspek dinamis dari sistem. Logika procedural, proses bisnis dan aliran kerja suatu bisnis bisa dengan mudah dideskripsikan dalam activity diagram. Activity diagram mempunyai peran seperti halnya flowchart, akan tetapi perbedaannya dengan flowchart adalah activity diagram bisa mendukung perilaku paralel sedangkan flowchart tidak bisa. Tujuan dari activity diagram adalah untuk menangkap tingkah laku dinamis dari sistem dengan cara menunjukkan aliran pesan dari satu aktifitas ke aktifitas lainnya[41]. Pada Tabel 2.2 adalah simbol-simbol yang sering digunakan pada saat pembuatan activity diagram.

Tabel 2.2 Simbol Activity Diagram

No.	Simbol	Nama	Deskripsi
1		Initial Code	Menandakan awal dimulainya suatu proses
2		Actions	Menggambarkan urutan dari proses atau akitivitas.
3		Flow	Menunjukkan berjalannya suatu proses.
4		Decision	Memberikan dua pilihan kondisi dengan sebuah arah <i>flow</i> masuk dan dua atau lebih arah <i>flow</i> yang keluar.
5		Merge	Menyatukan <i>flow</i> yang sebelumnya terpisah dari proses <i>decision</i> .
6		Fork	Ketika sebuah <i>flow</i> masuk dan terdapat dua <i>flow</i> keluar yang menjalankan dua aktivitas bersamaan.
7		Join	ketika terdapat dua <i>flow</i> masuk yang menjalankan dua aktivitas bersamaan dan hanya terdapat sebuah <i>flow</i> yang keluar.
8		Activity Final	Menunjukkan akhir dari suatu proses.
9		Subactivity Indicator	Menandakan suatu proses atau aktivitas yang dipecah pada activity diagram lain. Hal ini bertujuan agar activity diagram tidak terlalu kompleks
10		Connector	Mempermudah ketika activity diagram sudah terlalu kompleks. Sebuah flow masuk ke suatu

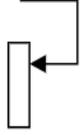
			connector yang telah diberi nama dan flow keluar dari connector dengan nama sama yang menunjuk suatu decision atau actions.
11		Swimlane	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.14.3 Sequence Diagram

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah scenario. Diagram ini menunjukkan sejumlah contoh objek dan *message* (pesan) yang diletakkan diantara objek-objek ini di dalam use case. Komponen utama sequence diagram terdiri atas objek yang dituliskan dengan kotak segiempat Bernama. *Message* diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan progress vertical[41]. Pada Tabel 2.3 menunjukkan symbol yang digunakan.

Tabel 2.3 Simbol Sequence Diagram

No.	Simbol	Nama	Deskripsi
1		Aktor	Mewakili peran orang, user, sistem yang lain atau alat berkomunikasi dengan sistem
2		Entity Class	Class yang digunakan untuk memodelkan informasi dan perilaku terkait yang harus disimpan
3		Boundary Class	Menandakan hubungan suatu form, report, hardware interface.

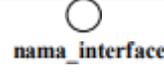
4		Control Class	Class yang digunakan untuk memodelkan perilaku satu atau lebih use case.
5		Activation	Menyatakan objek dalam keadaan aktif dan berinteraksi dengan pesan.
6		Message	Menandakan pengiriman pesan.
7		Self Message	Spesifikasi dari komunikasi untuk menunjukkan kegiatan memuat proses informasi pada aktifitas sendiri.

2.14.4 Class Diagram

Class diagram adalah diagram statis. Ini mewakili pandangan statis dari suatu aplikasi. Class diagram tidak hanya digunakan untuk memvisualisasikan, menggambarkan, dan mendokumentasikan berbagai aspek sistem tetapi juga untuk membangun kode eksekusi (*executable code*) dari aplikasi. Class diagram menggambarkan atribut, operation dan juga constraint yang terjadi pada sistem. Diagram UML yang lain seperti *activity diagram* dan *sequence diagram* hanya dapat menggambarkan urutan aliran aplikasi. Namun class diagram sedikit berbeda. Class diagram adalah satu-satunya diagram yang dapat memetakan secara langsung ke Bahasa pemrograman berorientasi objek. Karenanya banyak digunakan pada saat coding. Tujuan dari class diagram adalah untuk memodelkan pandangan statis suatu aplikasi. Pada Tabel 2.4 menunjukkan symbol yang digunakan.

Tabel 2.4 Simbol Class Diagram

No.	Simbol	Nama	Keterangan
-----	--------	------	------------

1		Package	Package merupakan sebuah bungkusan dari satu atau lebih kelas
2		Kelas	Kelas pada struktur sistem
3		Antarmuka	Sama dengan konsep interface dalam pemrograman berorientasi objek
4		Asosiasi	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity
5		Asosiasi berarah	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity
6		Generalisasi	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)
7		Kebergantungan	Relasi antar kelas dengan makna kebergantungan antar kelas
8		Agregasi	Relasi antar kelas dengan makna semua-bagian (whole-part)