BAB 2
TINJAUAN PUSTAKA

2.1 Sejarah *Traveling*

Untuk memenuhi kebutuhannya, secara naluri orang jaman dulu melakukan perjalanan dikarenakan adanya kebutuhan makan dan minum, kebutuhan rasa aman/tempat tinggal, perasaan ingin tahu, ambisi kehormatan dan kekuasaan sehingga dengan dorongan tersebut, mereka melakukan perjalanan yang mungkin kembali ketempat asal maupun tinggal menetap di tempat baru.

Sejak jaman purba, dimana manusia masih mengandalkan kehidupannya dengan memanfaatkan apa yang ada disekelilingnya untuk mempertahankan kehidupannya (secara naluri), hanya memakan apa yang ada disekitarnya, melindungi diri dari panas, hujan dan dingin mereka menutupi badannya dengan dedaunan/kulit pohon ataupun kulit binatang, istirahat tidur dengan mencari tempat aman diatas pohon rindang, atau dalam goa.

Secara naluri, bila merasakan kurang nyaman lagi ditempat semula (dimana buah-buahan, umbi-umbian liar, binatang buruan sudah habis dan tempat tinggalnya sudah diketahui oleh binatang buas atau musuhnya) mereka berusaha untuk bergerak mencari tempat baru yang lebih nyaman dan aman. Demikian seterusnya sampai akhirnya mereka menemukan teman dan berkumpul saling membantu menghadapi serangan binatang buas maupun musuh-musuh saingannya dalam mempertahankan hidup. Mereka berkomunitas membentuk group/kelompok, semakin kuat dan menetap disatu tempat untuk bersama-sama bahu membahu mempertahankan kelompoknya untuk terus hidup. Dalam perkembangannya, terjadilah suatu komunitas yang semakin banyak anggotanya, kemudian terbentuklah sebuah masyarakat dengan pemimpinnya.

Kondisi alam menuntut untuk tetap melakukan perjalanan, walaupun tidak langsung serempak, mereka mengutus dulu satu atau dua orang yang pemberani untuk ”mencari” daerah baru/hutan yang masih banyak makanan (karena mereka belum mengenal bercocok tanam), bila telah menemukan, barulah dengan

berbondong-bondong (merasa lebih aman dan kuat) kelompok tersebut ”berpindah” ke tempat baru tersebut.

Untuk mempertahankan kepentingannya, mereka saling menaklukkan antar kelompok, kemudian akan menjadikan kelompok yang menang menjadi kelompok besar, sampai akhirnya menjadi suku dalam lokasi yang menetap/desa dan menjadi lebih besar lagi menjadi negeri. Perjalanan tetap dilakukan dalam dinamika kehidupan, seakan-akan mereka sudah mengenal bahwa setiap langkah akan menjadikan seseorang/kelompok ”lebih besar” (lebih pandai, lebih banyak informasi karena lebih banyak yang dilihat).

Hornby cs, dalam bukunya yang berjudul “*The Advance Learn’s Dictionary Of Current English*”, menyatakan *“Travel is go from place to place, make a journey, up a long one, in search of pleasure, adventure or as commercial traveller”*. Travel diartikan sebagai pergerakan dari suatu tempat ke tempat lain, melakukan perjalanan atau ke tempat yang jauh dengan maksud untuk mencari kesenangan, berpetualang atau untuk perjalanan mencari keuntungan/laba/berdagang).

2.1.1 Tanda-tanda Perjalanan/*Traveling*

Tanda-tanda yang membuktikan terjadinya perjalanan pada masa itu :

- Tahun 221 - 122 SM, masa pemerintahan dinasti Chou di Tiongkok telah dibangun jalan raya untuk kepentingan lalu lintas bangsanya.terutama untuk perjalanan berdagang.
- Tahun 560-330 SM, masa kerajaan Persia Di Timur Tengah dibangun jalan raya yang menghubungkan antara kaki gunung Zagrep ke laut Algean.
- Kerajaan ROMAWI membangun jalan raya dengan nama THE APPIAN WAY sepanjang kurang lebih 350 mil.
- Tahun 334 SM di Ephesus (Turki) dikembangkan oleh Alexander The Great (Iskandar Zulkarnaen) mengadakan pertunjukkan akrobatik adu binatang buas, sihir dan sulap yang banyak dikunjungi orang.

- Perjalanan Ziarah ketempat-tempat yang dihormati merupakan perjalanan yang didorong oleh motivasi agama dan ritual penghormatan leluhur.
- Bangsa Phunisia Dan Polanesia merupakan bangsa yang pertama kali melakukan perjalanan untuk muhibah wisata dengan tujuan perdagangan. Dan Ratu Elisabeth I kaum ningrat Inggris juga mengadakan perjalanan ke Eropa, dengan berbagai tujuan antara lain kunjungan perkenalan, mencari ide-ide baru, mempelajari seni budaya, sistem pemerintahan dan pendapat bangsa lain.
- Bangsa Austronesia melakukan imigrasi ke kepulauan Indonesia melalui Malaya ke Jawa dan juga Formosa, Filipina ke sebagian Kalimantan dan Jawa, bangsa Austronesia ini yang akan menjadi nenek moyang langsung bangsa Indonesia. Mereka menghidupi diri dengan dengan berladang dan berburu binatang menggunakan anak panah dan busur dan tombak. Alat yang digunaka anak panah dan tombak. Mereka hidup di goa-goa berpindah dari satu tempat awal ketempat lain yang dianggap lebih nyaman.
- 500 SM - Terjadi perpindahan gelombang kedua. Perpindahan yang dilakukan sudah dengan membawa kebudayaan Dongson melalui jalan barat lewat Malaysia barat. Kebudayaan Dongson yaitu kebudayaan yang telah memakai logam sebagai alat bantu kehidupan dikenal dengan sebutan zaman perunggu. Mereka sudah mulai tinggal menetap.
- Abad ke-7 hingga abad ke-14 Jawa dan Sumatra. kedatangan pelautpelaut Tiongkok yang dipimpin oleh Laksamana Cheng Ho/Zheng He perang dan berdagang), serta para pedagang-pedagang Arab dari Gujarat, India, yang membawa agama Islam.
- Awal abad ke-16, orang-orang Eropa datang, mereka menemukan beberapa negara-negara kecil dan menguasainya melalui perdagangan rempah.

- Pada abad ke-17, Belanda muncul sebagai yang terkuat di antara negara-negara Eropa lainnya dan sedikit demi sedikit mulai menjajah Indonesia.

Perjalanan yang mereka lakukan berdampak pula pada pesatnya arus dan kebutuhan perjalanan, baik untuk sekedar kunjungan persahabatan, mencari pengetahuan maupun expansi perdagangan, Dari perjalanan tersebut kita mengenal beberapa tokoh sebagai traveller antara lain:

1. Marcopolo (1254-1324) yang telah mengadakan perjalanan dari Eropa hingga Tiongkok
2. Ibnu Battutah (1325) yang terkenal sebagai "The First Traveller Of Moslem" telah melakukan perjalanan dari Tanger (Afrika) ke Mekkah
3. Christopher Colombus (1451-1506) melakukan pelayaran dari Spanyol ke Barat dan menemukan Cuba dan Haiti (Amerika).
4. Vasco Da Gama (1498) yang menjelajahi 5 benua
5. Captain James Cook (1728 -1779) menjelajahi Selandia Baru dan Australia Timur.

Perjalanan dapat diartikan sebagai pergerakan manusia dari satu tempat ke tempat lain dengan berbagai tujuan. Pada jaman dahulu, perjalanan dilakukan karena naluri kebutuhan dari seseorang yang seringkali mengajak kelompoknya untuk mencari lokasi baru yang lebih nyaman dan aman. Seringkali perjalanan bisa dikembangkan menjadi perpindahan yang menetap (imigran) bila ditempat baru, mereka merasa lebih nyaman.

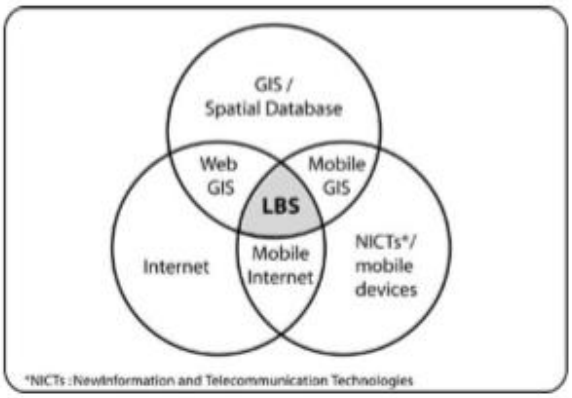
2.2 Landasan Teori

Landasan teori menjelaskan beberapa definisi dan teori yang berkaitan dengan pembangunan aplikasi sebagai dasar pemahaman dalam sebuah sistem serta metode yang digunakan untuk kegiatan pembangunan aplikasi tersebut.

2.2.1 *Location Based Service*

Location Based Service adalah aplikasi yang bergantung pada lokasi tertentu dan didefinisikan pula sebagai layanan informasi dengan memanfaatkan teknologi yang mengetahui posisi sesuatu. Layanan berbasis lokasi menggunakan teknologi *Positioning System*, teknologi ini memungkinkan para pengguna dapat memperoleh informasi lokasi sesuai dengan kebutuhannya.[5]

LBS termasuk dalam kategori teknologi yang sama dengan *Geographic Information System* (GIS), dan aplikasi *Global Positioning System* (GPS), yaitu dikenal dengan teknologi *geospatial*. Teknologi ini terdiri atas perangkat untuk mengumpulkan, menyimpan, menganalisa dan mendistribusikan data yang sesuai dengan kebutuhan pengguna terhadap sistem koordinat bumi. Layanan ini menjadi sangat penting bagi penggunanya karena mampu menghubungkan antara lokasi *geographic* informasi terhadap lokasi penggunanya, hal ini sangat mendukung era mobilitas seperti pada masa ini. Keberadaan aplikasi LBS merupakan hasil penggabungan dari tiga buah teknologi yaitu *New Information and Communication Technologies* (NICTS), internet dan GIS dengan menggunakan database spasial. Teknologi LBS ini terdiri atas perangkat-perangkat yang digunakan untuk mengumpulkan, menyimpan, menganalisa dan mendistribusikan data dan informasi pada berdasarkan sistem koordinat geografik bumi secara real-time. Identifikasi kordinat pengguna memungkinkan aplikasi LBS untuk menyediakan layanan bagi pengguna perangkat *mobile*.[5]



Gambar 2.1 *Location Based Service* Sebagai Simpang Tiga Teknologi

Secara garis besar jenis layanan berbasis lokasi juga dapat dibagi menjadi dua, yaitu:

1. *Pull Service*, Layanan diberikan berdasarkan permintaan dari pelanggan akan kebutuhan suatu informasi. Jenis layanan ini dapat dianalogikan seperti mengakes suatu web pada jaringan internet.[5]
2. *Push Service*, Layanan ini diberikan langsung oleh *service provider* tanpa menunggu permintaan dari pelanggan, tentu saja informasi yang diberikan tetap berkaitan dengan kebutuhan pelanggan.[5]

2.2.2 Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk membuat aplikasi mereka sendiri. Pada awalnya dikembangkan oleh Android Inc, sebuah perusahaan pendatang baru yang membuat perangkat lunak untuk ponsel yang kemudian dibeli oleh Google Inc. Untuk pengembangannya, dibentuklah Open Handset Alliance (OHA), konsorsium dari 34 perusahaan perangkat keras, perangkat lunak, dan telekomunikasi termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.[6]

2.2.3 RESTful Web Service

REST (*REpresentational State Transfer*) merupakan standar arsitektur komunikasi berbasis web yang sering diterapkan dalam pengembangan layanan berbasis web. Umumnya menggunakan HTTP (*Hypertext Transfer Protocol*) sebagai *protocol* untuk komunikasi data. REST pertama kali diperkenalkan oleh *Roy Fielding* pada tahun 2000.

Pada arsitektur REST, REST server menyediakan *resources* (sumber daya/data) dan REST client mengakses dan menampilkan *resource* tersebut untuk penggunaan selanjutnya. Setiap *resource* diidentifikasi oleh URIs (*Universal Resource Identifier*s) atau global ID. *Resource* tersebut direpresentasikan dalam

bentuk format teks, JSON atau XML. Pada umumnya formatnya menggunakan JSON dan XML. Berikut adalah keuntungan dari REST :

1. Bahasa dan platform agnostic.
2. Lebih sederhana/simpel untuk dikembangkan ketimbang SOAP.
3. Mudah dipelajari, tidak bergantung pada tools.
4. Ringkas, tidak membutuhkan layer pertukaran pesan (messaging) tambahan.
5. Secara desain dan filosofi lebih dekat dengan web.

Sedangkan untuk kelemahan dari REST adalah sebaga berikut ini :

1. Mengasumsi model point-to-point komunikasi - tidak dapat digunakan untuk lingkungan komputasi terdistribusi di mana pesan akan melalui satu atau lebih perantara.
2. Kurangnya dukungan standar untuk keamanan, kebijakan, keandalan pesan, dll, sehingga layanan yang mempunyai persyaratan lebih canggih lebih sulit untuk dikembangkan ("dipecahkan sendiri").
3. Berkaitan dengan model transport HTTP.

Berikut metode HTTP yang umum digunakan dalam arsitektur berbasis REST:

1. *GET*, menyediakan hanya akses baca pada *resource.*
2. *PUT*, digunakan untuk menciptakan *resource* baru.
3. *DELETE***,** digunakan untuk menghapus *resource.*
4. *POST***,** digunakan untuk memperbarui *resource* yang ada atau membuat *resource* baru.
5. *OPTIONS***,** digunakan untuk mendapatkan operasi yang di-*support* pada *resource.*

Web service adalah standar yang digunakan untuk melakukan pertukaran data antar aplikasi atau sistem, karena aplikasi yang melakukan pertukaran data bisa ditulis dengan bahasa pemrograman yang berbeda atau berjalan pada platform yang berbeda. Contoh implementasi dari *web service* antara lain adalah SOAP dan REST. *Web service* yang berbasis arsitektur REST kemudian dikenal

sebagai RESTful web services. Layanan web ini menggunakan metode HTTP untuk menerapkan konsep arsitektur REST.

2.2.4.1 Cara Kerja RESTful Web Service

Sebuah client mengirimkan sebuah data atau request melalui HTTP Request dan kemudian server merespon melalui HTTP Response. Komponen dari http request :

1. Verb, HTTP method yang digunakan misalnya GET, POST, DELETE, PUT dll.
2. *Uniform Resource Identifier* (URI) untuk mengidentifikasikan lokasi resource pada server.
3. HTTP Version, menunjukkan versi dari HTTP yang digunakan, contoh HTTP v1.1.
4. *Request Header*, berisi metadata untuk HTTP Request. Contoh, type client/browser, format yang didukung oleh client, format dari body pesan, seting cache dll.
5. *Request Body*, konten dari data.

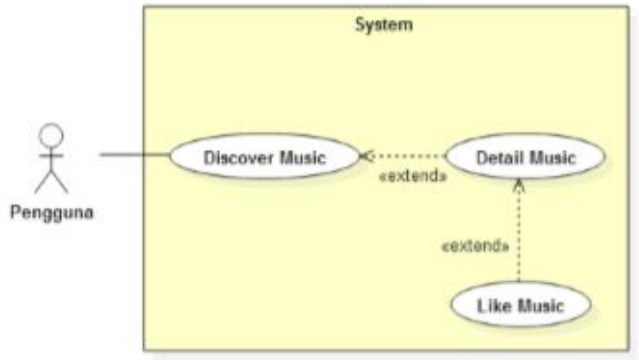
Sedangkan komponen dari HTTP *Response* :

1. Status/Response Code, mengindikasikan status server terhadap resource yang direquest. misal : 404, artinya *resource* tidak ditemukan dan 200 *response* OK.
2. HTTP Version, menunjukkan versi dari HTTP yang digunakan, contoh HTTP v1.1.
3. *Response Header*, berisi metadata untuk HTTP *Response*. Contoh, type server, panjang content, tipe content, waktu *response*, dll
4. *Response Body*, konten dari data yang diberikan.

2.2.6.1 *Use Case Diagram*

Use Case Diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *Use case* merepresentasikan sebuah interaksi antara aktor dengan sistem. Sebuah *use case* dapat meng-include fungsionalitas *use case* lain

sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa *use case* yang di-*include* akan dipanggil setiap kali *use case* yang meng-*include* dieksekusi secara normal. Sebuah use case dapat di-*include* oleh lebih dari satu *use case* lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang serupa. Sebuah *use case* juga dapat meng-*extend use case* lain dengan *behaviour*-nya sendiri. Sementara hubungan generalisasi antar *use case* menunjukkan bahwa *use case* yang satu merupakan spesialisasi dari yang lain. Dasar menentukan sebuah *use case* adalah *use case* merupakan sesuatu yang menyediakan beberapa hasil terukur kepada pengguna atau sistem eksternal. *Use case* harus memiliki sangat jelas kriteria lulus / gagal. Pengembang, tester, penulis teknis, dan pengguna harus secara eksplisit tahu apakah sistem memenuhi kasus penggunaan atau tidak. Setiap bagian dari *use case* yang memenuhi tes sederhana ini mungkin menjadi kandidat yang baik untuk use case.[9]

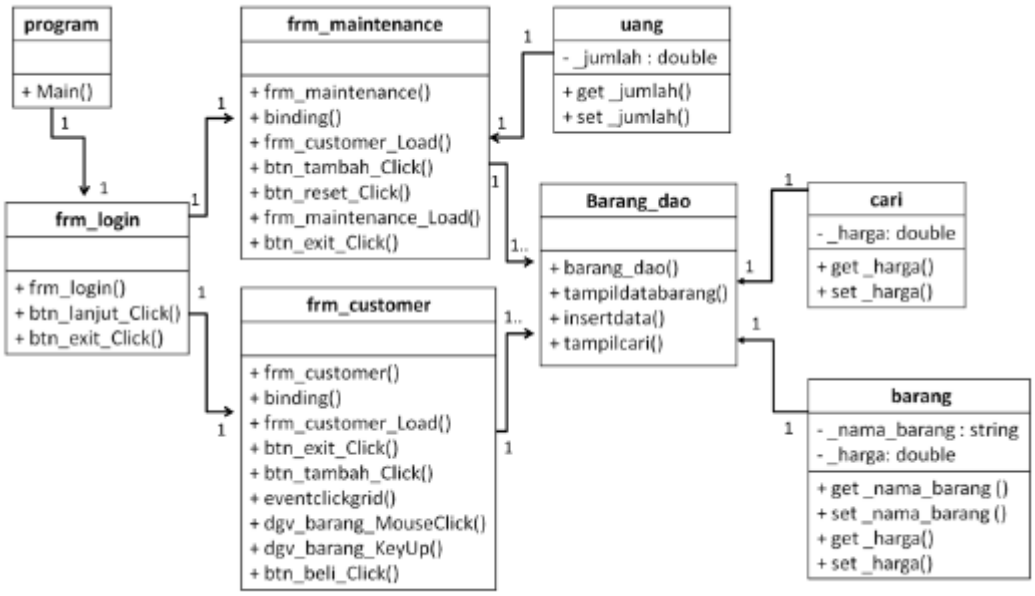


Gambar 2.2 Contoh *Use Case Diagram*

2.2.6.2 *Class Diagram*

Class diagram merupakan diagram yang selalu ada di permodelan sistem berorientasi objek. *Class diagram* menunjukan hubungan antar *class* dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan. Kelas pada kelas diagram terdiri dari 3 bagian utama yaitu nama kelas, isi property dari kelas beserta metode yang ada pada kelas tersebut. Kelas juga memiliki jenis - jenis hubungan seperti asosiatif, dependensi, agregasi,

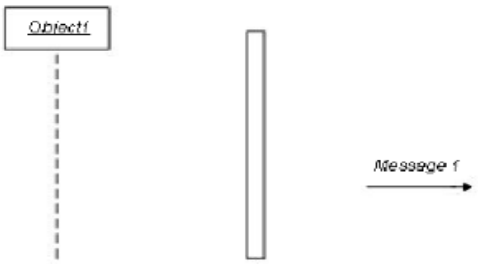
komposisi, spesifikasi dan generalisasi. Hubungan ini digunakan untuk menggambarkan bagaimana hubungan dan interaksi yang terjadi antar kelas.



Gambar 2.3 Contoh *Class Diagram*

2.2.6.3 *Sequence Diagram*

Sequence diagram adalah suatu diagram interaksi yang menekankan pada pengaturan waktu dari pesan-pesan. Diagram ini menampilkan sekumpulan peran dan pesan-pesan yang dikirim dan diterima oleh instansi yang memegang peranan tersebut. *Sequence diagram* menangkap objek dan *class* yang terlibat dalam skenario dan urut-urutan pesan yang ditukar antara objek diperlukan untuk



melaksanakan fungsionalitas skenario. *Sequence diagram* berasosiasi dengan *use case* selama proses pengembangan. Dalam *Unified Model Language* (UML), objek dalam *sequence diagram* di dengan segiempat yang berisi nama objek yang diberi garis bawah. Objek dapat diberi nama dengan tiga cara : (nama objek), (nama objek dan *class*) atau (hanya nama *class* (*anonymous object*)).

2.2.4 *Object Oriented Analisys and Design*

Konsep OOAD mencakup Analisis dan Desain sebuah sistem dengan pendekatan objek, yaitu analisis berorientasi objek (OOA) dan desain berorientasi objek (OOD). OOA adalah metode analisis yang memeriksa *requirement* (syarat / keperluan) yang harus dipenuhi sebuah sistem dari sudut pandang kelas kelas dan objek – objek yang ditemui dalam ruang lingkup sistem. Sedangkan OOD adalah metode untuk mengarahkan arsitektur software yang didasarkan pada manipulasi objek – objek sistem atau subsistem.

2.2.5 Perangkat Lunak Pendukung

Berikut adalah beberapa perangkat lunak pendukung untuk membangun aplikasi yang akan dibuat:

2.2.8.1 Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai computer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada *C* dan *C++* namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin aras bawah yang minimal. Aplikasi-aplikasi berbasis Java umumnya dikompilasi ke dalam *p-code* (*bytecode*) dan dapat dijalankan pada berbagi mesin virtual java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin.

Java merupakan bahasa berorientasi objek (OOP) yaitu cara ampuh dalam pengorganisasian dan pengembangan perangkat lunak. Pada OOP, program komputer sebagai kelompok objek yang saling berinteraksi. Deskripsi OOP secara ringkas adalah mengorganisasikan program sebagai kumpulan komponen yang disebut objek. Objek-objek ini ada secara independen, mempunyai aturan-aturan berkomunikasi dengan objek lain dan untuk memerintahkan objek lain guna meminta informasi tertentu atau meminta objek lain mengerjakan sesuatu.

2.2.8.2 MySQL

MySQL adalah sebuah aplikasi *Relational Database Management Server* (RDBMS) yang sangat cepat dan kokoh. Dengan menggunakan MySQL Server maka data dapat diakses oleh banyak pemakai secara bersamaan sekaligus dapat membatasi akses para pemakai berdasarkan *privilege* (hak akses) yang diberikan. MySQL menggunakan bahasa SQL (*Structured Query Language*) yaitu bahasa standar yang digunakan untuk pemrograman *database*. *Statement* atau *command* yang termasuk bahasa SQL secara umum dibagi menjadi dua kategori utam atau *data sublanguage* yaitu DDL dan DML. Setiap *sublanguage* mengkhususkan diri terhadap aspek tertentu dari bahasa tersebut. DDL menyertakan *statement* yang mendukung definisi atau kreasi dari objek data base, seperti tabel, *indeks, sequence,* dan tampilan. Beberapa statement adalah *CREATE, ALTER, dan DROP*. DML menyertakan *statement* yang memperbolehkan proses atau manipulasi objek database. Beberapa *staetment* DML yang sangat umum digunakan diantaranya *SELECT, INSERT, DELETE,* dan *UPDATE*.

2.2.6 Payment Gateway

Payment gateway adalah salah satu cara untuk memproses transaksi elektronik. *Paymentgateway* menyediakan alat – alat untuk memproses pembayaran antara *customer, businesses* dan *banks*. *Payment gateway* merupakan bagian terpenting dari suatu transaksi antar*customer*, *business,* dan lembaga – lembaga perbankan yang keduanya digunakan. *Payment Gateway* digunakan untuk memfasilitasi transaksi elektronik. Beberapa fitur – fitur utama dari *payment gateway* meliputi :

1. Aplikasi perangkat lunak dirancang khusus untuk *ecommerce,* walaupun bisa digunakan juga untuk mengotorisasi pembayaran dalam bisnis tradisional batu bara dan semen (*brick and mortar*).
2. Enkripsi pembayaran dan data pribadi.
3. Komunikasi antara lembaga – lembaga keuangan yang terkait, *business* dan *customer*.
4. Otorisasi pembayaran.

Beberapa *tools* mempunyai fitur *payment gateway* yang dapat membantu pelanggan mengetahui biaya pengiriman dan penanganan, serta pajak penjualan. Ada juga pendeteksian untuk penipuan dan fitur – fitur lainnya yang dapat digunakan dengan *payment gateway*. Banyak *web host ecommerce* menawarkan *payment gateway* sebagai bagian dari paket *hosting*.

Payment gateway mengambil keuntungan dari internet untuk mengirim dan menerima informasi. *Payment gateway* aplikasi yang dirancang secara khusus untuk memfasilitasi transaksi pembelian. Banyak bisnis tradisional menggunakan aplikasi tersebut dengan baik, karena bisa lebih akurat dan merupakan media otorisasi untuk pembayaran. Koneksi internet diperlukan, karena kebanyakan *payment gateway* menggunakan saluran komunikasi yang tersedia di internet terutama VPN. VPN (*virtual private network*) adalah jaringan pribadi (bukan untuk akses umum) yang menghubungkan medium nonpribadi (misalya *internet*) untuk menghubungkan antar *remote – site* secara aman. Dengan adanya VPN, maka resiko kebocoran informasi penting yang terjadi selama transaksi *online* bisa diminimalisir.