

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Konsep Dasar Aplikasi**

##### **2.1.1 Pengertian Aplikasi**

Aplikasi berasal dari kata *application* yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang dibuat untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju[9].

##### **2.1.2 Aplikasi Mobile**

Aplikasi mobile berasal dari kata aplikasi dan mobile. Aplikasi yang artinya penerapan, lamaran, penggunaan. Secara istilah aplikasi adalah program siap pakai yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain dan dapat digunakan oleh sasaran yang dituju sedangkan mobile dapat di artikan sebagai perpindahan dari suatu tempat ke tempat yang lain. Aplikasi mobile dapat diartikan sebuah program aplikasi yang dapat dijalankan atau digunakan walaupun pengguna berpindah-pindah dari satu tempat ke tempat yang lain serta mempunyai ukuran yang kecil. Aplikasi *mobile* ini dapat di akses melalui perangkat nirkabel, pager, PDA, telepon seluler, smartphone, dan perangkat sejenisnya[10].

#### **2.2 Konsep dan Definisi E-Commerce**

##### **2.2.1 Pengertian e-Commerce**

Perdagangan elektronik adalah bisnis di mana teknologi informasi digunakan untuk meningkatkan penjualan, efisiensi bisnis, dan menyediakan dasar untuk produk dan layanan baru. Melalui aktivitasnya, setiap perusahaan berkomunikasi dengan banyak entitas lain: mungkin klien pribadi atau korporat, mitra bisnis, pemasok. Saat berkomunikasi satu sama lain, entitas ini bertukar berbagai jenis informasi: mereka menginformasikan diri mereka sendiri tentang produk dan layanan mereka, menegosiasikan persyaratan transaksi,

bertukar dokumen, mengirimkan dan menerima pesanan barang, mengeluh tentang layanan yang tidak pantas, mendistribusikan siaran pers, dll. Aliran informasi juga dilakukan di antara semua entitas ini : manajer dan bawahan berkomunikasi dengan perusahaan, spesialis pemasaran berbicara tentang vendor dan manajer produk. Popularitas internet telah memberikan banyak peluang baru untuk bisnis.komunikasi yang dipercepat dan di sederhanakan serta kemampuan teknis yang tidak terbayangkan telah memungkinkan wirausahawan yang cerdas untuk membuat iklan, toko, dan item menjanjikan lainnya sendiri, di Internet. Selain itu, pertumbuhan jumlah pengguna internet telah meningkatkan volume dan profitabilitas penggunaan perdagangan dan layanan[11].

Definisi e-Commerce menurut Turban dan kawan-kawan ialah “proses membeli dan menjual; atau tukar-menukar produk, jasa atau informasi melalui computer”. Sedang menurut Laudon, e-Commerce ialah penggunaan Internet dan Web untuk transaksi bisnis ; atau secara lebih formal e-Commerce didefinisikan sebagai transaksi perdagangan yang dimungkinkan secara digital antar organisasi dengan organisasi atau dengan individual serta antar individual dengan individual.

Dari kedua definisi di atas, kita dapat menyimpulkan bahwa e-Commerce berkaitan dengan transaksi jual beli yang dilakukan secara digital dengan menggunakan computer yang tersambung dengan Internet. Oleh karena itu, peranan Internet dalam e-Commerce sangat penting karena computer yang digunakan untuk bertransaksi harus dapat digunakan untuk berkomunikasi antara pihak pembeli dan penjual.

Dengan demikian, model transaksi dalam e-Commerce akan berbeda dengan transaksi bisnis konvensional. Jika transaksi bisnis memerlukan tatap muka; maka dalam e-Commerce tidak diperlukan tatap muka.

### **2.2.2 Konsep Dasar dalam e-Commerce**

Untuk memahami e-Commerce dengan baik, maka diperlukan pemahaman konsep-konsep dasar secara benar. Dibawah ini terdapat beberapa konsep dasar mengenai e-Commerce yang perlu kita ketahui yang menunjukkan proses

perkembangan proses organisasi bisnis yang bersifat konvensional sampai dengan yang benar-benar bersifat digital / online :

1. Organisasi Brick and Mortar : organisasi bisnis konvensional yang melakukan bisnis secara tradisional (offline) dengan menjual produk-produk fisik dengan agen-agen fisik.
2. Organisasi Maya (Virtual Organization): organisasi bisnis yang melakukan kegiatan bisnis hanya melalui Internet / secara online saja.
3. Organisasi Click and Mortar: organisasi bisnis yang melakukan perdagangan tertentu secara online, umumnya hanya sebagai tambahan saluran pemasaran; sedang kegiatan bisnis utamanya tetap dilakukan secara konvensional.
4. Pasar Elektronik (E-marketplace): pasar online dimana penjual dan pembeli bertemu untuk melakukan pertukaran barang, jasa, uang atau informasi.
5. Sistem Informasi Antar-Organisasi (Interorganizational Information System): sistem komunikasi yang memungkinkan proses transaksi rutin dan aliran informasi antara dua organisasi atau lebih.
6. Sistem Informasi dalam Organisasi (Intraorganizational Information System): sistem komunikasi yang memungkinkan kegiatan e-Commerce berjalan dalam organisasi-organisasi individual.
7. Intranet : jaringan komunikasi dalam suatu organisasi bisnis yang digunakan untuk berkomunikasi antar pegawai dalam menunjang bisnis utama mereka.
8. Extranet: jaringan-jaringan yang menggunakan Internet untuk menghubungkan beberapa Intranet.
9. Internet: jaringan global yang menghubungkan jaringan-jaringan local di seluruh dunia.
10. Business to Business (B2B): model e-Commerce dimana semua partisipan merupakan organisasi bisnis.
11. Business to Consumer (B2C): model e-Commerce dimana organisasi bisnis menjual ke pembeli individual.
12. E-tailing: pengecer yang melakukan kegiatannya secara online.

13. Business to Business to Consumer (B2B2C): model e-Commerce dimana organisasi bisnis menyediakan produk atau jasa untuk bisnis klien yang digunakan pelanggan klien tersebut.
14. Consumer to Business (C2B): model e-Commerce dimana individu-individu menggunakan Internet untuk menjual produk atau jasa kepada organisasi atau individu yang mencari penjual untuk menawarkan produk atau jasa yang mereka perlukan.
15. Consumer to Consumer (C2C): model e-Commerce dimana individual-individual atau kelompok berkomunikasi atau bekerja sama.
16. Collaborative Commerce (e-commerce) : model e-Commerce dimana individual-individual atau kelompok berkomunikasi atau bekerja sama.
17. E-learning: penghantaran informasi secara online untuk tujuan pelatihan atau Pendidikan.
18. E-government: model e-Commerce dimana suatu organisasi pemerintah membeli atau menyediakan barang, jasa, atau informasi dari atau untuk organisasi bisnis atau individu.
19. Call to action (CTA): pesan dalam kaitannya dengan pemasaran yang mencoba untuk membujuk seseorang yang membaca sehingga yang bersangkutan akan melakukan suatu tindakan sesuai dengan pesan tersebut. Sebagai contoh “Beli 2 produk yang ditawarkan akan mendapat 1 produk lain secara gratis”.
20. Conversation rate: persentase pengunjung yang mengambil tindakan yang diinginkan (melalui CTA).
21. Google checkout: sistem pembayaran online yang didukung oleh Google, dimana para pengguna menyimpan informasi kartu kredit mereka dalam Google account sehingga mereka dapat dengan cepat melakukan pembelian secara online pada situs-situs yang menyediakan layanan tersebut.
22. Paypal: jasa layanan pembayaran online yang memungkinkan para pengguna melakukan pembelian dan menerima pembayaran melalui alamat email yang sudah didefinisikan sesuai dengan pengguna.

23. Shopping cart: perangkat lunak yang digunakan untuk menyediakan katalog produk tersedia untuk dipesan dimana pengunjung dapat, memilih, melihat, menambah atau menghapus serta membeli barang-barang yang ditawarkan pada suatu web tertentu.

Dari beberapa konsep dasar tersebut dapat disimpulkan bahwa organisasi-organisasi bisnis maupun individu-individu melakukan perdagangan elektronik secara berbeda-beda. Secara umum perdagangan elektronik dilakukan hanya sebagian kecil saja, misalnya sebagai tambahan saluran pemasaran dan perdagangan dilakukan secara total secara online[12].

## **2.3 Fulfillment**

### **2.3.1 Model operasi order fulfillment**

Order fulfillment merupakan salah satu aktivitas penting dalam e-commerce. Secara sederhana, order fulfillment didefinisikan sebagai proses penerimaan order dari pelanggan sampai pengiriman produk ke alamat yang diinginkan customer. Dalam transaksi e-dagang, order fulfillment ini sering disebut sebagai back-office operations yang melibatkan beberapa aktivitas untuk pemenuhan pesanan customer, seperti pengepakan (packing), pengantaran (delivery), akuntansi, pengelolaan inventory, dan pengiriman (shipping). Order fulfillment sangat terkait dengan front-office operations atau customer-facing activities, seperti advertising dan pengambilan pesanan (order taking) yang secara nyata dapat dirasakan oleh customer[13].

Turban, Outland, King, et al dalam buku Electronic Commerce 2018, menjelaskan pentahapan dalam proses order fulfillment:

1. Customer melakukan order dan membayar atas order-nya;
2. Verifikasi pembayaran oleh penjual jika diperlukan;
3. Memeriksa ketersediaan stok dan memberikan notifikasi ke customer;
4. Menentukan apakah inventory sebaiknya harus diisi kembali atau apakah diperlukan produksi tambahan untuk mengganti pengurangan stok karena pemenuhan order;

5. Menentukan lokasi warehouse dimana order tersebut akan dipenuhi. Menginformasikan proses pemenuhan order ke warehouse atau fulfillment center;
6. Pengambilan order di lokasi penyimpanan dan pengepakan (pick & pack);
7. Pengiriman barang ke customer;
8. Penerimaan barang oleh customer;
9. Penjual mengelola barang return jika diperlukan.

Proses order fulfillment bisa saja bervariasi, tergantung pada karakteristik produk (seperti jenis, ukuran, kategori perishability, dll), apakah pihak ketiga terlibat dalam proses di warehousing dan pengiriman, apakah model bisnis B2C atau B2B, dan model operasi masing-masing perusahaan. Namun demikian, umumnya model operasi fulfillment memiliki bentuk sebagai berikut (Turban, et al, 2018):

1. Engineer-to-order (ETO). Produk dirancang sesuai spesifikasi keinginan customer. Umumnya produk-produk perhiasan menggunakan model operasi ETO.
2. Make-to-order (MTO). Produk baru dibuat bila benar-benar ada pemesanan dari customer.
3. Assemble-to-order (ATO). Produk dibuat berdasarkan modular, yang dirakit dari berbagai komponen yang tersedia.
4. Make-to-stock (MTS). Produk-produk yang standar dan kategori mass-production umumnya diproduksi untuk mengisi stok dalam inventory.
5. Digital copy (DC). Copy produk dilakukan dengan cara mengunduh produk-produk dalam bentuk digital dan inventory dihasilkan dari digital master.

Karena umumnya produk-produk yang diperdagangkan dalam e-dagang berupa pakaian, apparel, makanan dalam kemasan, peralatan elektronik, maka model operasi fulfillment dalam e-dagang paling banyak adalah MTS dan ATO.

Sembilan aktivitas dalam proses order fulfillment memerlukan integrasi supply chain. Order fulfillment memerlukan integrasi aliran keuangan (payment), informasi, material, dan komponen, yang memerlukan koordinasi antar departemen dalam perusahaan dan antar perusahaan yang terlibat dalam supply chain.

### **2.3.2 Peran Warehousing**

Warehousing memainkan peran penting dalam order fulfillment, khususnya untuk model MTS atas produk-produk standar. Aktivitas warehousing mencakup:

1. Penerimaan barang;
2. Penyimpanan barang;
3. Pengambilan barang bila order diterima;
4. Pengemasan barang;
5. Pengaturan pengiriman atau delivery.

Untuk mendukung proses order fulfillment dalam e-dagang diperlukan operasional warehousing yang cepat, akurat, dan fleksibel, dengan biaya yang paling efisien. Dalam beberapa decade terakhir, teknologi warehousing telah dikembangkan seperti yang dilakukan oleh Newegg, e-dagang elektronik yang sangat besar di AS:

- a. Hybrid of OPEX perfect pick;
- b. High-velocity picking menggunakan pick to light;
- c. Pick to voice;
- d. Automated print & apply system untuk order dan pengiriman;
- e. Human Machine Interface untuk sistem visibility dan monitoring;
- f. Exacta Warehouse Control System (WCS).

Pengembangan dan penerapan teknologi warehousing di Newegg memberikan hasil perbaikan kinerja warehousing yang signifikan sebagai berikut :

- a. High system throughput: rata-rata 18.000 order per hari selama peak season, 8000 per hari untuk hari biasa;
- b. Pengurangan order cycle time menjadi 20 menit;

- c. High system accuracy;
- d. Biaya pengiriman semakin murah;
- e. Security;
- f. Scalability.

### **2.3.3 Peran Delivery**

Sebagai bagian penting dalam order fulfillment model bisnis e-dagang, delivery akan menjadi tahapan penyelesaian transaksi e-dagang. Para pengelola e-dagang menghadapi permasalahan delivery dalam jumlah item banyak untuk alamat tujuan pengantaran yang sangat banyak dan tersebar luas. Umumnya mereka bekerja sama dengan perusahaan kurir seperti Pos Indonesia, JNE , TIKI, J&T dan lain-lain untuk pengantaran barang.

Tuntutan utama dalam delivery adalah kecepatan (speed). Customer menginginkan barang yang dipesan harus segera diterima dalam waktu secepat mungkin. Saat ini, customer menginginkan waktu pengantaran tidak lagi dalam hitungan hari. Customer menginginkan waktu pengantaran dalam hitungan jam. Standar waktu pengiriman same day delivery telah menjadi hanya beberapa jam (few hours) karenanya, tren kedepan pengantaran menggunakan drone dan robot.

Idealnya, e-trailer mengharapkan pengantaran barang lebih cepat daripada pelanggan mendapatkan produknya melalui pergi ke toko dan melakukan pembelian di sana. Solusi di masa depan adalah pengiriman paket melalui drone yang memerlukan waktu dalam beberapa menit. Drone merupakan pesawat tanpa awak (self-flying vehicle), sama halnya seperti mobil tanpa pengemudi (self-driving car) yang dikendalikan melalui remote.

Di Auckland, New Zealand, di mana lalu lintas cukup padat, penggunaan drone banyak dimanfaatkan untuk pengantaran pizza. Sementara itu, Amazon telah melakukan riset secara intensif penggunaan drone untuk pengantaran paket. Beberapa kendala penggunaan drone untuk pengantaran seperti yang diidentifikasi oleh Black antara lain aspek legal, teknologi sensor, dan lain lain.



Senada dengan Black, Mehra mengatakan bahwa kendala utama penggunaan drone untuk pengantaran paket adalah isu keamanan dan regulasi lalu lintas udara. Lebih jauh Mehra melihat beberapa kendala drone untuk pengantaran paket:

1. Keterbatasan berat;
2. Keterbatasan jarak tempuh pengantaran;
3. Kendala cuaca yang menyebabkan kemungkinan delay;
4. Kemungkinan penghentian layanan karena drone merupakan flying machines. Drone bisa jatuh karena kecelakaan, kondisi cuaca, masalah mesin, dan lain lain.

#### **2.3.4 Sejumlah Tantangan**

Order fulfillment diyakini menjadi kunci keberhasilan dalam bisnis e-dagang. Performa order fulfillment yang tidak baik akan memengaruhi kepuasan customer. Beberapa tantangan kerap dihadapi dalam operasional order fulfillment sebagaimana ditunjukkan dalam penelitian Van Landingham berikut ini :

- a. Order flexibility. Perubahan dan pembatalan order seringkali terjadi dalam menit-menit terakhir;
- b. Order accuracy. Kekuatan pemrosesan dan pemenuhan order dalam e-dagang merupakan hal yang penting. Keliru dalam pemrosesan order akan berakibat kehilangan customer;
- c. Multichannel order management. Saat ini hampir semua perusahaan menyediakan multichannel order secara terintegrasi, yang dikenal dengan Omni-channel;
- d. Complex distribution. Berbeda dengan offline order, setiap order dalam e-dagang biasanya dalam ukuran kecil dan sangat banyak, sehingga delivery menjadi kompleks.

Ketidakpuasan customer terjadi sebagai hasil dari ketidakakuratan order, proses order yang memerlukan waktu lama, dan skedul pengantaran yang tidak

sesuai. Permasalahan ini umumnya disebabkan oleh perencanaan dan eksekusi manajemen supply chain yang kurang efektif. Beberapa penyebab utamanya antara lain:

- a. Ketidakpastian dalam permintaan;
- b. Kesenjangan dalam information sharing;
- c. Infrastruktur logistic yang tidak memadai;
- d. Aliran keuangan yang tidak efisien.

## **2.4 Firebase Cloud Messaging**

Firestore Cloud Messaging (FCM) adalah solusi perpesanan cross platform yang memungkinkan mengirim pesan tanpa biaya[14] dan tidak memiliki kuota. Firestore cloud messaging digunakan melalui jaringan seluler atau Wi-Fi, dengan biaya hampir nol. Dengan menggunakan JSON untuk menarik (retrieve) data dari firestore cloud messaging. Firestore Cloud Messaging (FCM) menawarkan berbagai solusi perpesanan dengan opsi kemampuan yang lebih baru. Bagian ini berfokus pada dua jenis layanan-Jenis pesan dan pesan Pemberitahuan. Firestore cloud messaging adalah perpesanan default yang digunakan pada Android. Dengan menggunakan FCM, klien dapat diberi tahu jika ada data baru yang siap disinkronkan. FCM juga memungkinkan pengiriman pesan notifikasi ke setiap pengguna di segmen tertentu[3].

FCM juga menyediakan fungsi untuk melakukan push notification, yaitu notifikasi yang muncul di bagian atas layar smartphone dan dapat di seret ke bawah, untuk mengakses pesan lengkapnya pengguna cukup menekan pesan yang tampil pada notifikasinya. Penggunaan Fitur push notification dengan FCM sangat membantu karena FCM akan mengirimkan notifikasi secara realtime.

Mengirim pesan notifikasi dari server ke klien dapat dilakukan dengan dua cara, konsol notifikasi GUI dan lingkungan tepercaya. Menggunakan konsol sangat efektif untuk uji coba atau pemasaran yang sangat bertarget dan keterlibatan pengguna. Dengan cara ini tidak diperlukan coding, kita hanya perlu menginput

data-data yang dibutuhkan. Untuk mengirimkan notifikasi melalui Trusted environment dapat dilakukan dengan menggunakan Admin SDK dan HTTP / XMPP. Aplikasi perlu memperluas `FirebaseMessagingService` untuk mendapatkan notifikasi dari Firebase. service ini akan menangani semua notifikasi yang didapat aplikasi ini. Untuk mengirim notifikasi ke perangkat tertentu, aplikasi perlu memperluas `FirebaseInstanceIdService` untuk mendapatkan token pendaftaran. Token ini digunakan untuk membedakan setiap aplikasi yang dipasang di setiap perangkat[15].

Berikut adalah gambar Arsitektur FCM :



***Gambar 1 Arsitektur FCM***

Sumber : D. Firdaus, "Implementation of Push Notification for Business Incubator", 2019[15]

Firebase Cloud Messaging (FCM) adalah platform untuk membuat sebuah aplikasi mobile atau aplikasi web dengan peralatan dan infrastruktur yang memungkinkan pengembangan lebih mudah dan menghasilkan aplikasi berkualitas tinggi. FCM ini adalah salah satu fitur yang disediakan oleh Google, awalnya Firebase ini digunakan sebagai database yang real-time yang memberi sebuah API untuk menyinkronkan dan menyimpan data di berbagai perusahaan.

Fitur-fitur yang terdapat pada Firebase adalah :

- a. Firebase Cloud Messaging: Fitur ini adalah platform yang digunakan untuk pesan dan notifikasi untuk android, ios, dan aplikasi web tanpa biaya.
- b. Firebase Analytics: Fitur ini adalah solusi pengukuran aplikasi gratis yang memberikan wawasan tentang penggunaan aplikasi keterlibatan pengguna.
- c. Realtime Database: Firebase menyediakan database real-time dan backend sebagai layanan. Pengembangan dilengkapi dengan API untuk menyinkronkan data aplikasi diseluruh perusahaan dan disimpan pada Firebase's cloud[16].

## **2.5 Geofence**

### **2.5.1 Definisi Geofencing**

Menurut Muriach, geofencing adalah suatu teknik yang memungkinkan aplikasi untuk memberikan informasi dengan cara yang lebih tepat, di waktu yang tepat dan di tempat yang tepat.

Teknik geofencing ini membutuhkan 2 komponen fisik utama yaitu wilayah fisik yang dibatasi oleh pagar-pagar digital dan perangkat yang mampu menerima informasi yang diberikan melalui Teknik geofencing tersebut. Dan ketika perangkat tersebut melewati salah satu area geofence (memasuki atau keluar area tersebut) maka akan ada alert atau notifikasi yang dikirimkan ke perangkat tersebut.

Sedangkan menurut Dabhi, geofencing adalah batas virtual atau garis-garis geografis atau perimeter virtual di sekitar satu titik dengan himpunan batas yang telah ditetapkan sebelumnya di wilayah geografis yang dipetakan baik menggunakan GPS (Global Positioning System) atau RFID (Radio Frequency Identification) atau beacons atau teknologi lain. Dan geofencing juga bisa dianggap sebagai fitur program perangkat lunak GPS dengan sistem navigasi radio berbasis antariksa.

Aplikasi yang mengimplementasikan geofencing dapat mengirimkan alert kepada administrator ketika ada perangkat mobile atau objek fisik lainnya masuk

atau keluar dari area yang sudah “dipagari” (geofenced). Alert tersebut dapat berupa pesan teks, email, panggilan telepon, atau notifikasi lainnya[18].

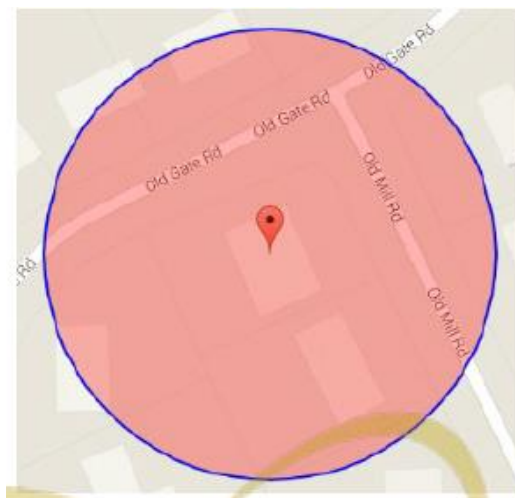
### 2.5.2 Jenis-jenis Geofencing

Ada 2 jenis geofencing yaitu sebagai berikut :

#### a. Geofencing Statis

Geofencing statis adalah geofencing yang ada pada umumnya yaitu batas virtual atau garis-garis geografis atau parameter virtual di sekitar satu titik dengan himpunan batas yang telah ditetapkan sebelumnya di wilayah geografis yang dipetakan baik menggunakan GPS (Global Positioning System) atau RFID (Radio Frequency Identification) atau beacons atau teknologi lain. Jadi pada geofencing statis ini lokasi yang dibatasi sudah jelas dan tidak berubah-ubah bentuknya.

Geofencing mengkombinasikan lokasi pengguna saat ini dengan jarak pengguna terhadap lokasi yang sudah ditentukan. Untuk menandai lokasi yang diinginkan maka dibutuhkan latitude dan longitude dari lokasi tersebut. Dan untuk menyesuaikan jarak pengguna terhadap lokasi tersebut maka dibutuhkan sebuah radius. Kemudian latitude, longitude dan radius tersebut akan menciptakan area geofences, yang mana area tersebut dapat berupa lingkaran atau polygon.



**Gambar 2. 1 Geofence Lingkaran**

Sumber : Penerapan Absen Mahasiswa Berbasis Android Menggunakan Teknologi Qr Code Dan Geofence [18]

Geofencing menggunakan kordinat GPS untuk membentuk satu area geografis secara digital dan mengambil lokasi pengguna aplikasi mobile melalui GPS untuk menentukan jarak pengguna terhadap area tersebut. Baik ketika pengguna tersebut berada di dalam atau di luar area tersebut atau hanya masuk sebentar kemudian keluar dari area tersebut. Ketika ada pengguna yang memasuki area yang sudah ditandai maka pengguna itu dapat menerima notifikasi bahwa ia telah memasuki suatu area. Kemudian geofencing dapat mengetahui seberapa lama pengguna itu berada di dalam area yang ditandai.

Ada 3 kondisi pengguna yang dapat diketahui ketika geofencing diterapkan dan pengguna dapat menerima notifikasi saat berada dalam kondisi tersebut. Ketiga kondisi tersebut yaitu :

1. Arriving

Arriving adalah kondisi ketika pengguna memasuki lokasi yang telah ditentukan

2. Dwelling

Dwelling adalah kondisi dimana pengguna masih berada di dalam lokasi yang telah ditentukan.

3. Leaving

Leaving adalah kondisi ketika pengguna telah berada di dalam lokasi yang ditentukan kemudian ia keluar dari lokasi tersebut.

- b. Geofencing Dinamis

Dynamic geofencing atau yang bisa disebut Situation Fencing berbeda dengan static geofencing yang membuat suatu pagar virtual berdasarkan pada kordinat-kordinat geografis saja. Dynamic Geofencing adalah suatu Teknik geofencing yang membuat batasan-batasan atau pagar-pagar virtual tersebut berubah secara otomatis berdasarkan situasi atau kondisi yang diperoleh dari kombinasi antara data personel pengguna dengan data yang ada secara global. Atau secara singkat adalah mekanisme pengiriman alert atau notifikasi berdasarkan situasi / kondisi yang diinginkan masing-masing pengguna.

Salah satu contohnya batasan-batasan area dengan polusi yang tinggi, curah hujan yang tinggi, kemacetan, kegiatan ekonomi yang ada disekitar, area dengan potensi flu dan lain-lain. Dan ketika ada kondisi yang sesuai dengan kondisi yang diinginkan pengguna di area tersebut maka pengguna akan mendapatkan notifikasi (alert). Misalnya ketika pengguna akan melewati area dengan polusi yang tinggi sedangkan ia tidak ingin melewati tempat dengan polusi yang tinggi maka sebelum pengguna memasuki area tersebut ia akan mendapat notifikasi untuk tidak melewatinya. Dynamic Geofencing atau Situation Fencing dapat diaplikasikan di berbagai bidang, mulai dari kesehatan dan gaya hidup, pemasaran produk, asuransi, pariwisata dan lain-lain.

Dynamic Geofencing membutuhkan kombinasi antara data yang berasal dari beraneka ragam sensor yang terdistribusi dengan data personel penggunaan serta membutuhkan tool atau alat yang mampu mengirim alert atau notifikasi ke setiap individu. Dan beruntungnya trend yang ada sekarang di bidang data dan komputerisasi sudah mendukung untuk setiap aspek tadi.

Geofencing dinamis membutuhkan variable situasi makro dan variable situasi personal. Variable situasi makro dapat diperoleh dengan mengkombinasikan data dari sensor-sensor, media social, dan geo-temporal data (contohnya: gambar dari satelit). Sedangkan variable data personal data diperoleh dari kombinasi antara profil dan kondisi pilihan pengguna dengan data dari perilaku pengguna tersebut (contohnya : tempat apa yang sering pengguna kunjungi). Kemudian variable makro dan variable personal tersebut akan dikombinasikan untuk menghasilkan situasi yang cocok bagi masing-masing pengguna dan notifikasi berbeda akan dikirimkan untuk setiap kondisi yang berbeda pula.

Di bawah ini merupakan contoh pengguna Dynamic Geofencing beserta contoh personal dan makro datanya.

<b>Kategori</b>	<b>Rekomendasi</b>	<b>Data Personal</b>	<b>Data Makro</b>
Mobilitas	Berkendara / Rute Jogging	Pilihan pengguna, sensitivitas	Level polusi, kemacetan

		terhadap polusi, kemacetan	
Makanan / minuman	Makanan / minuman	Makanan pilihan, riwayat makanan yang pernah dibeli, jumlah kalori, alergi	Ulasan tentang restoran, bahan baku makanan
Gaya hidup	Event atau tempat rekomendasi	Tempat pilihan, hari ulang tahun	Tren local , cuaca,, tingkat kriminalitas
Produk rekomendasi	Iklan, kupon diskon	Jarak ke lokaso, Minat pengguna, Saldo pengguna	Tren local, harga online produk sejenis

### 2.5.3 Manfaat Geofencing

Manfaat-manfaat yang bisa didapatkan dari penggunaan Teknik geofencing contohnya adalah sebagai berikut :

1. Orang tua dapat memantau lokasi anak apakah anaknya pergi ke sekolah atau ke lokasi lain
2. HRD dapat memantau para karyawan yang bekerja di lokasi yang sudah ditentukan
3. Karyawan dapat melakukan absen melalui smartphome meraka ketika berada di lokasi yang sudah ditentukan
4. Di bidang keamanan, pemilik mobil akan mendapatkan notifikasi ketika mobil yang diparkir ternyata dibawa oleh pencuri

### 2.6 QR- Code Scanner

Kode QR adalah suatu jenis kode matriks atau kode batang dua dimensi yang dikembangkan oleh Denso Wave, sebuah divisi Denso Corporation yang merupakan sebuah perusahaan Jepang dan dipublikasikan pada tahun 1994 dengan fungsionalitas utama yaitu dapat dengan mudah dibaca oleh pemindai QR



merupakan singkatan dari quick response atau respons cepat, yang sesuai dengan tujuannya adalah untuk menyampaikan informasi dengan cepat dan mendapatkan respons yang cepat pula. Berbeda dengan kode batang, yang hanya menyimpan informasi secara horizontal, kode QR mampu menyimpan informasi secara horizontal dan vertical, oleh karena itu secara otomatis Kode QR dapat menampung informasi yang lebih banyak daripada kode batang[19].

### **2.6.1 Fungsi QR Code**

Kode QR berfungsi bagaikan hipertaut fisik yang dapat menyimpan alamat dan URL, nomer telepon, teks dan sms yang dapat digunakan pada majalah, surat harian, iklan, pada tanda-tanda bus, kartu nama ataupun media lainnya. Atau dengan kata lain sebagai penghubung secara cepat konten daring dan konten luring. Kehadiran kode ini memungkinkan audiens berinteraksi dengan media yang ditempelinya melalui ponsel secara efektif dan efisien. Pengguna juga dapat menghasilkan dan mencetak sendiri kode QR untuk orang lain dengan mengunjungi salah satu dari beberapa ensiklopedia kode QR.

#### **1. Kepentingan Komersial**

Selain itu, desain QR memungkinkan penggunaannya untuk memasukkan logo perusahaan, klip video ataupun foto ke kode QR, tanpa menghilangkan substansi informasi apapun dari sumber yang dimasukkan.

#### **2. Kepentingan Umum**

Kode QR dapat dimanfaatkan sebagai keamanan makanan dengan cara menambahkan kode QR yang berisikan data-data mengenai kandungan nutrisi dan masa kadaluarsa pada tiap label makanan sehingga pelanggan dapat merasa lebih aman dalam memilih makanan yang dibeli sebab mereka dapat mengetahui informasi-informasi tentang makanan tersebut.

### **2.6.2 Penggunaan QR Code**

Awalnya kode QR digunakan untuk pelacakan kendaraan bagian di manufaktur, namun kini kode QR digunakan dalam konteks yang lebih luas, termasuk aplikasi komersial dan kemudahan pelacakan aplikasi berorientasi yang ditujukan untuk pengguna telepon selular. Di Jepang, penggunaan kode QR sangat populer, hampir semua jenis ponsel di Jepang bisa membaca kode QR sebab sebagian besar pengusaha di sana telah memilih kode QR sebagai alat tambahan dalam program promosi produknya, baik yang bergerak dalam perdagangan maupun dalam bidang jasa. Pada umumnya kode QR digunakan untuk menanamkan informasi alamat situs suatu perusahaan. Di samping itu, banyak pengguna menggunakan teknologi kode QR sebagai penyembunyi pesan. Berikut adalah beberapa teknologi yang dapat digabung dengan teknologi kode QR:

1. Tracking Number (Memeriksa Nomor Resi / Tracking Suatu Barang)
2. Transport Ticketing (Nomor Id Penumpang)
3. Pesan Singkat
4. Nomor Telepon
5. Email
6. Sistem Penjualan atau Sistem Diskon
7. Alamat Website
8. Kartu Nama
9. Product Number, dll.

### **2.6.3 Desain QR Code**

Desain dari kode QR berbentuk kotak persegi dengan warna mendasar putih sebagai tampilan belakang layer atau background dan diisi dengan gabungan beberapa data berbentuk kotak persegi kecil atau berbentuk pixel berwarna hitam. Rumit atau tidaknya posisi pixel pada kode QR bergantung pada jumlah serta kombinasi data yang terkandung di dalamnya. Adapun data yang dapat disimpan

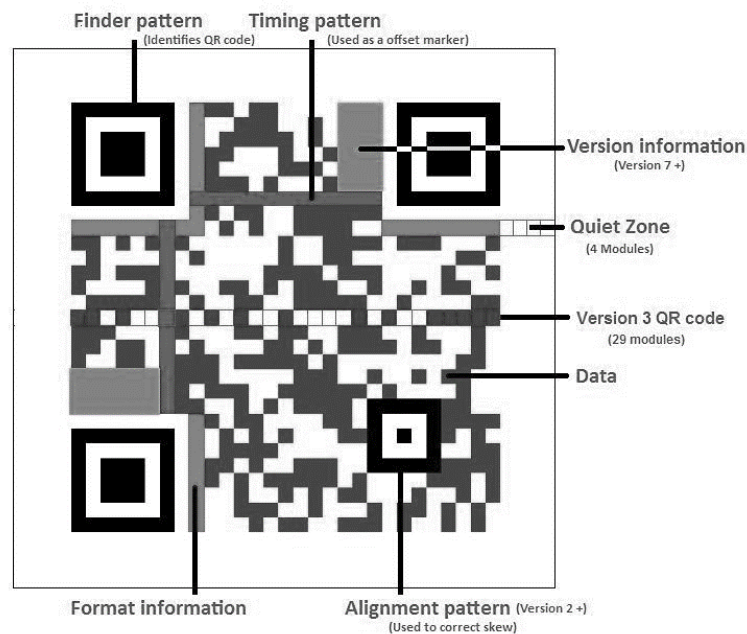
dalam kode QR berupa nomor, huruf, alphanumeric, byte, symbol dan bahkan kanji dapat dimasukkan kedalam QR code.



**Gambar 2. 2 Contoh Simbol QR Code**

Sumber : Sistem Absensi Mahasiswa Menggunakan Mobile QR Code (Studi Kasus Universitas Widyatama) [19]

Seperti pada Gambar 2.2, terdapat pola struktur tersendiri ketika membuat kode QR. Dalam pembuatannya ukuran kode QR harus dibuat seimbang atau sama sehingga membentuk persegi. Di dalam QR Code terdapat beberapa pola dimana pola tersebut tidak boleh dilakukan encoding data. Selain itu symbol harus dikelilingi oleh zona garis putih atau yang biasa disebut Quiet Zone. Berikut struktur dari kode QR :



**Gambar 2. 3 Struktur QR Code**

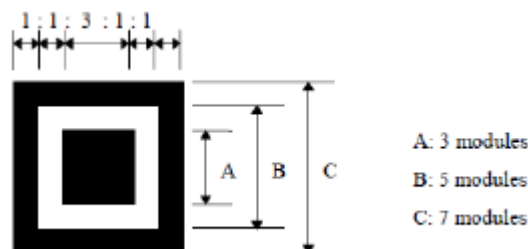
Sumber : Sistem Absensi Mahasiswa Menggunakan Mobile QR Code (Studi Kasus Universitas Widyatama) [19]

### 1. Symbol Version

Terdapat 40 ukuran dari kode QR yang disebut sebagai versi 1, versi 2, dst. Dari setiap versi tersebut memiliki kapasitas penyimpanan data yang berbeda. Semakin besar data yang disimpan maka semakin besar pula gambar kode QR yang terbentuk.

### 2. Finder Pattern

Finder pattern diletakan pada 3 buah titik khusus yang terletak di kanan atas, kiri atas dan kiri bawah gambar kode QR. Pola gambar dari finder pattern berbentuk persegi berukuran 7 x 7 warna gelap, ditambah 5 x 5 warna terang, dan ditambah 3 x 3 warna gelap. Sehingga perbandingan pola finder pattern yaitu 1:1:3:1:1 seperti yang diilustrasikan pada Gambar 2.4. Symbol ini sangat penting, dikarenakan symbol ini mendefinisikan orientasi serta lokasi pola dalam melakukan proses encoding.



**Gambar 2. 4 Pola Position Detection**

Sumber : Sistem Absensi Mahasiswa Menggunakan Mobile QR Code (Studi Kasus Universitas Widyatama) [19]

### 3. Separator

Sebuah separator atau pembatas ditempatkan diantara masing-masing pola position detection. Format pola mengelilingi position detection dengan menggunakan warna terang.

#### 4. Timing Pattern

Timing pattern berada pada setiap ujung pola position detection yang memotong dari ujung pola detection satu ke ujung pola detection lainnya. Format pola dari pembatas membentuk sebuah garis yang berisikan nilai 1 dan 0 yang nilainya dimasukan secara bergantian.

#### 5. Alignment Patterns

Merupakan sebuah pola untuk menyelaraskan posisi dengan finder pattern. Ini bertujuan agar membuat beberapa blok bagian apabila data yang disimpan sangatlah banyak. Alignment patterns biasanya digunakan pada kode QR versi 2 ke atas.

#### 6. Encoding Region

Merupakan wilayah dari data yang dimasukan dan di-encode kedalam bentuk kode QR. Pada encoding region juga terdapat Error Connection yaitu sebuah proses pelacakan kesalahan selama transmisi data berlangsung.

#### 7. Quiet Zone

Merupakan sebuah zona dimana tidak ada pola atau marking yang terdapat pada zona tersebut. Zona ini berada pada 4 sisi gambar kode QR. Fungsi dari Quiet Zone ini yaitu agar dapat menentukan zona kode pada kode QR. Apabila kode QR tidak memiliki Quiet Zone maka akan ada kemungkinan ketika pembacaan kode QR akan sulit terbaca bahkan tidak terbaca sedikitpun.

### **2.6.4 Kelebihan QR Code**

Teknologi QR Code merupakan pengembangan dari teknologi barcode. Teknologi barcode banyak sekali kekurangan terutama dalam penyimpanan data sehingga dibuatlah teknologi QR Code. Berikut merupakan beberapa kelebihan dari teknologi QR Code:

1. Kode QR memiliki kapasitas tinggi dalam data pengkodean, yaitu mampu menyimpan semua jenis data, seperti data numerik, data alfabeta, kanji, kana, hiragana, symbol, dan kode biner.
2. Kode QR mampu menyimpan data jenis numerik sampai dengan 7.089 karakter, data alfanumerik sampai dengan 4.296 karakter, kode biner sampai dengan 2.844 byte, dan huruf kanji sampai dengan 1.817 karakter
3. Kode QR memiliki tampilan yang lebih kecil daripada kode batang. Hal ini dikarenakan kode QR mampu menampung data secara horizontal dan vertikal, oleh karena itu secara otomatis ukuran dari tampilannya gambar kode QR bisa hanya sepersepuluh dari ukuran sebuah kode batang.
4. Kode QR juga tahan terhadap kerusakan, sebab kode QR mampu memperbaiki kesalahan sampai dengan 30%. Oleh karena itu, walaupun sebagian simbol kode QR kotor ataupun rusak, data tetap dapat disimpan dan dibaca. Tiga tanda berbentuk persegi di tiga sudut memiliki fungsi agar simbol dapat dibaca dengan hasil yang sama dari sudut manapun sepanjang 360 derajat.

## **2.7 Web Service**

### **2.7.1 Definisi Web Service**

Web service adalah suatu sistem perangkat lunak yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. Web service diartikan sebagai sebuah antar muka (interface) yang menggambarkan sekumpulan operasi-operasi yang dapat diakses melalui jaringan, misalnya internet dalam bentuk pesan XML (eXtensible Markup Language).

Web Service adalah aplikasi perangkat lunak yang tersedia pada web yang melaksanakan fungsi yang spesifik. Sedangkan menurut Michael C. Daconta, Web Service adalah aplikasi perangkat lunak yang dapat ditemukan, diuraikan, dan diakses berdasarkan pada XML dan protokol standard Web pada intranet, extranet, dan Internet.

Web service menyediakan standar komunikasi di antara berbagai aplikasi software yang berbeda-beda, dan dapat berjalan di berbagai platform maupun framework. Web service digunakan sebagai suatu fasilitas yang disediakan oleh suatu web untuk menyediakan layanan (dalam bentuk informasi) kepada sistem lain, sehingga sistem lain dapat berinteraksi dengan sistem tersebut melalui layanan-layanan (service) yang disediakan oleh suatu sistem yang menyediakan web service.

Teknologi pada web service dapat mengubah kemampuan transactional web, yaitu kemampuan web untuk saling berkomunikasi dengan pola program-to-program (P2P). Fokus web selama ini didominasi oleh komunikasi program-to-user dengan interaksi business-to-consumer (B2C), sedangkan transactional web akan didominasi oleh program-to-program dengan interaksi business-to-business.

Web service sebenarnya adalah kumpulan dari fungsi dan method yang terdapat pada sebuah server yang dapat dipanggil oleh klien dari jarak jauh, kemudian untuk memanggil method-method tersebut kita bebas menggunakan aplikasi yang akan dibuat dengan bahasa pemrograman apa saja yang dijalankan pada platform apa saja.

Adanya teknologi web service dapat menjembatani perbedaan-perbedaan teknologi dari masing-masing sumber. Dapat ditarik kesimpulan bahwa web service merupakan kumpulan layanan yang disediakan melalui jaringan berbasis web dengan standar yang telah ditetapkan mampu menunjang interoperabilitas, dan dapat berjalan diberbagai platform dan framework[20].

Berikut beberapa kelebihan Web Service :

1. Web Service dapat digunakan untuk mentransformasikan satu atau beberapa bisnis logic atau class dan objek yang terpisah dalam satu ruang lingkup yang menjadi satu, sehingga tingkat keamanan dapat ditangani dengan baik.
2. Web Service memiliki kemudahan dalam proses deployment-nya, karena tidak memerlukan registrasi khusus ke dalam suatu sistem operasi. Web

Service cukup di-upload ke web server dan siap diakses oleh pihak-pihak yang telah diberikan otorisasi.

3. Web service berjalan di port 80 yang merupakan protocol standar HTTP, dengan demikian web service tidak memerlukan konfigurasi khusus di sisi firewall[21].

### 2.7.2 Arsitektur Web Service

Service Publication and Discovery (UDDI)
Service Description (WSDL)
XML Based Messaging (SOAP)
Common Internet Protocol (HTTP, SMTP, TCP/IP)

**Tabel 2. 1 Lapisan dasar Web Service**

Tabel 2.1 merupakan blok bangunan web service yang mana menyediakan fasilitas komunikasi jarak jauh antara dua aplikasi yang merupakan layer arsitektur web service.

- a. Layer 1 : protocol internet standar yang digunakan sebagai sarana transportasi adalah HTTP dan TCP/IP.
- b. Layer 2 : Simple Object Access Protocol (SOAP) berbasiskan XML dan digunakan untuk pertukaran informasi antar sekelompok layanan.
- c. Layer 3 : Web Service Definition Language (WSDL) digunakan untuk mendeskripsikan attribute layanan.
- d. Layer 4 : Universal Description Discovery and Integration, yang mana merupakan direktori pusat untuk deskripsi layanan.



Komponen web service antara lain :

a. Extensible Markup Language (XML )

XML merupakan dasar yang penting atas terbentuknya Web services. Web services dapat berkomunikasi dengan aplikasi-aplikasi yang memanggilnya dengan menggunakan XML , karena XML berbentuk teks sehingga mudah untuk ditransportasikan menggunakan protokol HTTP. Selain itu, XML juga bersifat platform independen sehingga informasi di dalamnya bisa dibaca oleh aplikasi apapun pada platform apapun selama aplikasi tersebut menerjemahkan tag-tag XML.

b. Simple Object Access Protocol (SOAP)

XML saja tidak cukup agar Web services dapat berkomunikasi dengan aplikasi yang lainya. XML yang digunakan untuk saling bertukar informasi antara web services dengan aplikasi yang lainya harus menggunakan sebuah format standard yang dapat dimengerti oleh keduanya. Format itulah yang dikenal dengan nama SOAP. SOAP (Simple Object Access Protocol) merupakan suatu format standard dokumen berbentuk XML yang digunakan untuk melakukan proses request dan responses antara web services dengan aplikasi yang memanggilnya. Dokumen SOAP digunakan untuk melakukan request disebut dengan SOAP request sedangkan dokumen SOAP yang diperoleh dari Web services disebut dengan SOAP responses.

c. Web service Definition Language (WSDL)

Sebelum mengakses sebuah Web services pastinya perlu mengetahui method-method apa saja yang disediakan oleh Web services tersebut, untuk mengetahuinya memerlukan sebuah dokumen yang bernama WSDL. WSDL (Web services Description Language) adalah sebuah dokumen dalam format XML yang isinya menjelaskan informasi detail sebuah Web services. Di dalam WSDL dijelaskan method-method apa saja yang tersedia dalam Web services, parameter

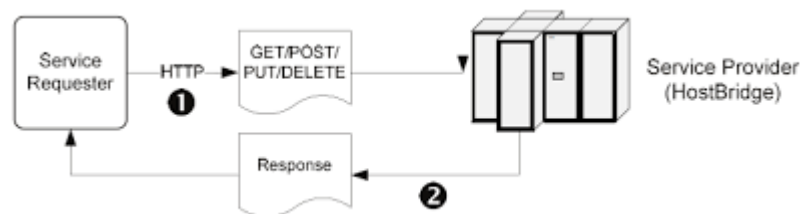
apa saja yang diperlukan untuk memanggil sebuah method, dan apa hasil atau tipe data yang dikembalikan oleh method yang dipanggil tersebut.

### 2.7.3 Teknologi REST

Ada beberapa pendekatan teknologi web services yang dapat digunakan, yaitu metode SOAP dan REST. Dalam penelitian ini akan digunakan pendekatan REST. REST merupakan singkatan dari REpresentative State Transfer. Pertama kali dikemukakan dalam disertasi seorang program doktor bernama Roy Thomas Fielding pada tahun 2000. REST adalah sebuah metode dalam menyampaikan resource melalui media web. Sedangkan resource sendiri didefinisikan sebagai segala sesuatu yang dapat disimpan didalam sebuah komputer dan ditampilkan sebagai urutan bit, misalnya sebuah dokumen, tabel dalam sistem basis data, atau hasil dari sebuah perhitungan (Sandoval, 2008). REST (REpresentational State Transfer) adalah model arsitektur yang pada dasarnya memanfaatkan teknologi dan protokol yang sudah ada seperti HTTP (Hypertext Transfer Protocol) dan XML.

### 2.7.4 Arsitektur REST

Dibawah ini merupakan model dasar dari RESTful Web services.



**Gambar 2. 5 Model dasar RESTful Web services**

Sumber : Rancang Bangun Aplikasi E-Commerce Untuk Booking Online Hotel Menggunakan Layanan Web Service [20]

Berikut merupakan langkah-langkah yang dilakukan dalam model dasar RESTful Web services:

1. Query Request Provider melalui HTTP dengan menggunakan URI (Uniform Resource Identifier). Request menggunakan methods (metode) HTTP untuk menentukan apakah request tersebut dimaksudkan untuk Create (menciptakan), Read (membaca), Update (memperbarui), atau Delete (menghapus) data.
2. HostBridge mengembalikan sebuah dokumen dalam bentuk XML untuk Requester (pemohon) dengan CICS data enclosed.

Diagram di atas menunjukkan dua web services : sebuah Provider (HostBridge) yang menyediakan layanan web services dan sebuah Request yang digunakan pada web services. Dalam model RESTful layanan dideskripsikan secara sendiri (self-describing), sehingga tidak diperlukan WSDL dan Server UDDI yang bertindak sebagai Broker.

Sehingga pada development web modern aksi CRUD tersebut dapat dimappingkan dengan HTTP method sebagai berikut : CREATE sebagai POST, RETREIVE sebagai GET, UPDATE sebagai PUT, dan DELETE sebagai DELETE. Berikut merupakan penggunaan methods(metode) HTTP dalam REST Web services:

<b>Metode</b>	<b>Deskripsi</b>
GET	Mendapatkan (read) sebuah sumber daya (resource) yang diidentifikasi dengan URI (Uniform Resource Identifier)
POST	Mengirimkan sumber daya (resource) ke server. Digunakan untuk membuat (create) sumber daya baru
PUT	Mengirimkan sumber daya (resource) ke server. Digunakan untuk memasukkan (insert) atau memperbarui (update) sumber daya yang tersimpan.
DELETE	Menghapus (delete) sumber daya (resource) yang diidentifikasi dengan URI.
HEAD	Mendapatkan metadata (response header) dari sumber daya(resource) yang diidentifikasi dengan URI.

Arsitektur REST dibangun dengan sifat sebagai berikut :

1. Addressability

Dalam prinsip ini seluruh suberdaya atau resource harus tersedia melalui sebuah alamat unik, pengalamatan ini dilakukan dengan menggunakan URI (Unique Resource Identifiers)

2. Uniform Interface

Semua interaksi sebaiknya dibangun dengan interface yang seragam. RESTful service menampilkan semua resource dan interaksinya dengan interface yang seragam, dalam metode REST antarmuka yang digunakan adalah dengan menggunakan HTTP. HTTP menawarkan semua operasi yang diperlukan, dikenal, dan tersebar luas. Semua interaksi antara klien dan sumber daya (resource) didasarkan pada metode dasar HTTP. Metode-metode interaksi yang digunakan dengan dasar HTTP ditunjukkan pada tabel 2.2

3. Representation-oriented

Representasi menjelaskan dalam bentuk apa data sedang dipertukarkan antara client dan server. Pada umumnya data dipertukarkan dalam bentuk XML, JSON, dan HTML.

4. Statelessness

Setiap interaksi antara client dan server harus memiliki state sendiri (atau dengan kata lain tidak dipengaruhi session client). Jadi server hanya akan memantau resource state bukan client session.

5. Hypermedia As The Engine Of Application State (HATEOAS)

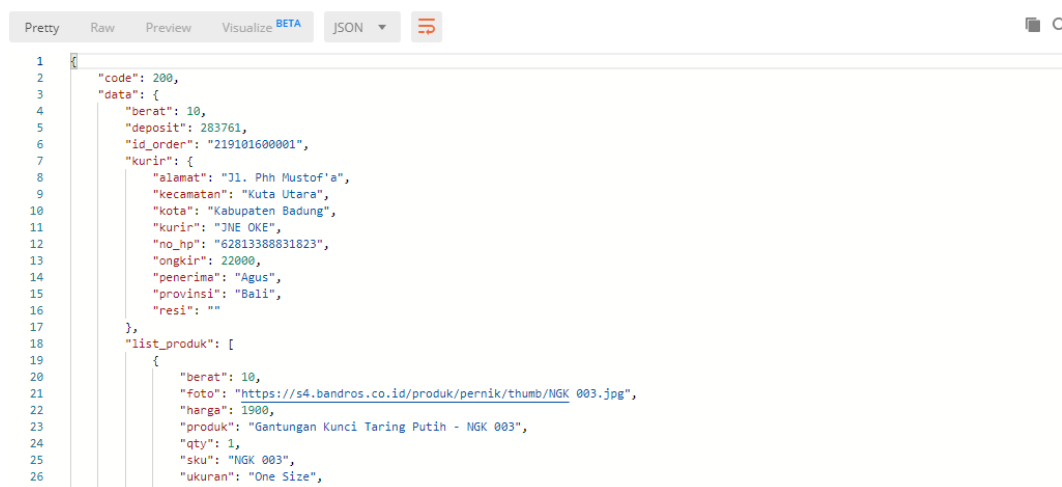
Hypermedia sebagai state dari sebuah aplikasi (HATEOAS), menyatakan REST dapat menggunakan link untuk menghubungkan sumber daya atau resource ke sumber daya lain yang berkaitan. Hal ini mirip dengan web, dimana kami menggunakan hyperlink untuk menghubungkan antara situs web[20].

## 2.7.5 JavaScript Object Notation (JSON)

JSON adalah sebuah format *data-interchange* yang ringan. Sangat mudah bagi manusia untuk membaca dan menuliskannya. Semudah mesin memarsing dan men-generate format datanya. JSON merupakan bagian dari bahasa pemrograman JavaScript. JSON merupakan format teks yang sepenuhnya independen tetapi menggunakan konvensi yang familiar dengan bahasa pemrograman dari C, PHP, C#, C++, Java, JavaScript, Perl, Python, Go dan sebagainya. Kelebihan inilah yang membuat JSON menjadi sebuah bahasa data-interchange yang ideal. JSON dibangun dalam dua struktur.

1. Beberapa pasangan dari nama/nilai. Dalam beberapa bahasa perograman biasa disebut dengan istilah object, record, struct, tabel hash, key list atau associative array.
2. Nilai-nilai yang terusun secara ordered list. Biasa disebut dengan array, vector, list atau daftar dalam bahasa pemrograman.

JSON merupakan struktur data universal, dalam artian dapat digunakan di berbagai bahasa pemrograman. Dan hampir semua bahasa pemrograman mendukung penuh JSON dalam berbagai format. Hal ini memungkinkan format data yang dapat dipertukarkan menggunakan bahasa pemrograman juga menggunakan dasar dari struktur JSON[10].



```

1  {
2    "code": 200,
3    "data": {
4      "berat": 10,
5      "deposit": 283761,
6      "id_order": "219101600001",
7      "kurir": {
8        "alamat": "Jl. Phh Mustof'a",
9        "kecamatan": "Kuta Utara",
10       "kota": "Kabupaten Badung",
11       "kurir": "JNE OKE",
12       "no_hp": "62813388831823",
13       "ongkir": 22000,
14       "penerima": "Agus",
15       "provinsi": "Bali",
16       "resi": ""
17     },
18     "list_produk": [
19       {
20         "berat": 10,
21         "foto": "https://s4.bandros.co.id/produk/pernik/thumb/NGK_003.jpg",
22         "harga": 1900,
23         "produk": "Gantungan Kunci Taring Putih - NGK 003",
24         "qty": 1,
25         "sku": "NGK 003",
26         "ukuran": "One Size",

```

**Gambar 2. 6 Contoh Format JSON**

Sumber : Pembangunan Aplikasi Manajemen Event ( Studi Kasus di Wah Wah Event )  
[10]

Format JSON mempunyai aturan sebagai berikut:

1. Object adalah satu set nama/nilai yang tidak terurut. Penulisan object dimulai dengan tanda '{' (left brace) dan diakhiri dengan tanda '}' (right brace). Setiap nama diikuti oleh tanda ':' (colon) dan pasangan nama/nilai dipisahkan dengan tanda ',' (comma).
2. Array adalah sekumpulan nilai yang teratur (An array is an ordered collection of values). Penulisan sebuah array dimulai dengan tanda '[' (left bracket) dan diakhiri dengan tanda ']' (right bracket). Nilai dipisahkan menggunakan tanda ',' (comma).
3. Nilai bisa berupa string dalam tanda kutip, atau number (angka), TRUE atau FALSE atau NULL, sebuah object atau sebuah array. Struktur ini dapat ditulis menggunakan metode bersarang.
4. String adalah rangkaian atau urutan karakter unicode yang berada dalam tanda kutip, bisa juga hanya berisi karakter kosong, menggunakan tanda '\' (backslash) untuk escape. Karakter di representasikan sebagai string tunggal.
5. Number sangat mirip dengan definisi pada bahasa C atau Java, hanya saja tipe bilangan oktal dan heksadesimal tidak digunakan.
6. Ruang kosong (*Whitespace*) dapat diselipkan diantara pasangan atau token apa saja. Kecuali detail encoding, yang secara jelas akan berfungsi untuk menjelaskan bahasa yang digunakan.

## 2.8 Android

### 2.8.1 Pengenalan Android



**Gambar 2. 7 Logo Android**

Sumber : <https://1.bp.blogspot.com/-MoXcwRdwxNA/UOwlG5wzDyI/AAAAAAAAAKqw/WlRaxDupPnY/s1600/LOGO+ANDROID.png>

Android adalah sistem operasi untuk perangkat mobile berbasis Linux yang awalnya dikembangkan oleh Android Inc. Android terdiri dari system operasi, middleware dan aplikasi. Android adalah istilah dalam Bahasa Inggris yang berarti robot yang menyerupai manusia yang dapat dilihat pada Gambar 2.6. Pada tahun 2005, Google secara resmi telah membeli Android, sehingga pengembangan Android sepenuhnya berada ditangan Google. Dalam proses pengembangan sistem operasi Android, dibentuklah organisasi Open Handset Alliance. Google merilis software open source untuk Android, sehingga dapat berkontribusi untuk mengembangkan Android.

Berikut beberapa kelebihan Android :

1. Sangat mudah mengoprasikan Smartphone Android.
2. Banyak fitur.
3. Dari segi tampilan, Android tidak kalah bagusnya dengan OS lain.
4. Operating system dibuat open source untuk pengembangan.
5. Aplikasi yang gratis hingga berbayar, dan bisa di-download di Google Play.

Android telah mengalami pembaruan, hal ini dikarenakan untuk pengembangan fitur Android sesuai dengan perkembangan zaman[21].

Berikut versi Android dari awal rilis sampai saat ini sebagai berikut [22]:

1. Android 1.0 ( Apple Pie/ Alpha ) dan Android 1.1 ( Banana Bread )

Sistem Android yang dikenal dengan nama Android Alpha dan Android Beta ini pertama kali diperkenalkan pada tahun 2007 dan baru mulai diaplikasikan pada smartphone di awal Maret 2009. Sebagai cikal bakal OS 10 Universitas 17 Agustus 1945 Surabaya Android, versi ini bisa dikatakan cukup berhasil menjadi inisiator kesuksesan besar Android.

2. Android 1.5 ( Cupcake )

Sesaat setelah mengaplikasikan sistem operasinya pada smartphone, pada Mei 2009 Android kembali merilis versi terbarunya yang diberi nama Android Cupcake. Android Cupcake menawarkan beragam kelebihan dibandingkan dengan versi terdahulunya, yakni fasilitas unggah video ke Youtube, headset bluetooth yang nirkabel serta tampilan keyboard dan gambar yang lebih atraktif.

3. Android 1.6 (Donut)

Tipe Android yang satu ini diluncurkan hanya berselang 4 bulan setelah peluncuran saudara kandungnya, Android Cupcake. Android Donut telah mengusung keunggulan lainnya yakni tampilan indikator baterai, fasilitas zoom in zoom out, penggunaan koneksi CDMA serta beberapa keunggulan lainnya.

4. Android 2.0 / 2.1 (Eclair)

Masih di tahun 2009, Android kembali meluncurkan teknologi terbarunya yang diberi nama Android Éclair. Era Android Éclair kemudian berhasil menarik para perusahaan gadget untuk mulai membuat gadget dengan sistem layar sentuh yang sebelumnya dianggap kurang user friendly bagi para pengguna smartphone.

5. Android 2.2 (Froyo = Frozen Yoghurt)

Android Froyo diluncurkan pada Mei 2010, 5 bulan setelah peluncuran Android Éclair. Pada tipe Android yang satu ini, keinginan para pengguna



smartphone untuk memiliki kapasitas memori eksternal berupa slot micro SD sudah dapat diwujudkan.

#### 6. Android 2.3 (Ginger Bread)

Android ini diluncurkan pada penghujung tahun 2010 memiliki tampilan yang atraktif dan disertai dengan penambahan fitur-fitur seperti dual camera dan video call. Tak hanya itu saja, Ginger Bread juga fokus pada peningkatan kualitas dan grafis game berbasis Android.

#### 7. Android 3.0/3.1 (Honeycomb)

Android yang diluncurkan pada Mei 2011 ini dikhususkan bagi para pengguna tablet PC berbasis Android. User interface nya pun berbeda dengan smartphone Android. Spesifikasi hardware yang tinggi serta tampilan layar yang lebih besar membuat Honeycomb cocok diaplikasikan pada tablet PC.

#### 8. Android 4.0 (Ice Cream Sandwich)

Ice Cream Sandwich diluncurkan pada tahun yang sama dengan peluncuran Honeycomb. Banyak sekali fitur-fitur baru yang disematkan pada Android versi empat ini, diantaranya adalah fitur pendeteksi wajah, fitur yang memaksimalkan kualitas fotografi, kualitas video yang lebih baik serta resolusi dan grafis gambar yang sangat memuaskan.

#### 9. Android 4.1 (Jelly Bean)

Sistem Android yang satu ini memberikan support pada fitur on screen keyboard yang membuat kegiatan mengetik menjadi lebih cepat, mudah dan responsif. Salah satu smartphone keren yang mengusung Android Jelly Bean adalah Google Nexus 7 yang diprakarsai oleh perusahaan ASUS.

#### 10. Android 4.4 (Kitkat)

Android Kitkat merupakan versi Android terbaru yang diluncurkan pada bulan September 2013. Penggunaan nama Kitkat menjadi suatu kejutan bagi para

pecinta Android, karena nama Kitkat merupakan salah satu nama snack wafer yang populer di dunia. Penggunaan nama ini juga membuat tipe Android ini semakin mudah diingat orang.

#### 11. Android v5.0 – 5.1 Lollipop

Dirilis pada tanggal 15 Oktober 2014, versi OS ini mengusung perubahan besar dari segi UI yang nampak lebih flat dengan konsep material design. Versi Android ini sudah mendukung arsitektur 64-bit sehingga sudah memungkinkan untuk penggunaan RAM diatas 3 GB pada hardware perangkat. Penggunaan prosesor 64-bit pun makin banyak diadopsi oleh para vendor, mulai dari penerapan pada perangkat flagship hingga perangkat kelas menengah kebawah.

#### 12. Android v6.0

Marshmallow Versi Android ini resmi dirilis pada bulan September tahun 2015. Bersamaan dengan dirilisnya versi ini, untuk pertama kalinya Google juga memperkenalkan 2 perangkat smartphone Nexus sekaligus yang diproduksi oleh 2 vendor yang berbeda.

#### 13. Android v7.0 Nougat

Resmi diperkenalkan pada akhir Juni 2016. Banyak netizen yang berspekulasi bahwa kemungkinan besar, pemberian nama untuk Android versi “N” ini adalah Nutella. Namun Google menepis kabar tersebut setelah resmi memperkenalkannya bersamaan dengan dipamerkannya patung icon Android yang berdiri diatas potongan Nougat (yang sepiintas lebih mirip dengan tempe itu).

#### 14. Android v8.0 Oreo

Untuk pengguna android sekarang lagi booming booming nya nih tentang OS terbarunya, ya itu adalah OS android versi 8.0 atau yang sering juga disebut sebagai Android O atau Android Oreo. Versi android ini resmi diperkenalkan oleh Google pada tanggal 22 Agustus 2017 yang lalu dan juga sudah secara resmi bisa diluncurkan langsung ke lapangan, tapi sebelum versi android ini diresmikan oleh

google, nama “Oreo” sendiri sudah terendus sejak Android O pertama kali diperkenalkan di ajang Google I/O 2017 pada Mei 2017 lalu.

#### 15. Android v9.0 Pie

Android Pie (versi 9.0) mengganti tiga tombol navigasi dengan tombol tunggal berbentuk elips. Karenanya, Android Pie pun lebih mengandalkan gestur gerakan-gerakan jari untuk melakukan hal-hal seperti multitasking. Fitur lain termasuk “Digital Wellbeing” untuk menginformasikan soal pola pemakaian perangkat, Adaptive Battery untuk membatasi pemakaian baterai oleh aplikasi, dan App Action yang langsung menjalankan fitur aplikasi dari app drawer.

### 2.8.2 Sistem Operasi Android

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008.

Antarmuka pengguna Android umumnya berupa manipulasi langsung, menggunakan gerakan sentuh yang serupa dengan tindakan nyata, misalnya menggeser, mengetuk, dan mencubit untuk memanipulasi objek di layar, serta papan ketik virtual untuk menulis teks. Selain perangkat layar sentuh, Google juga telah mengembangkan Android TV untuk televisi, Android Auto untuk mobil, dan Android Wear untuk jam tangan, masing-masingnya memiliki antarmuka pengguna yang berbeda. Varian Android juga digunakan pada komputer jinjing, konsol permainan, kamera digital, dan peralatan elektronik lainnya.

Android adalah sistem operasi dengan sumber terbuka, dan Google merilis kodenya di bawah Lisensi Apache. Kode dengan sumber terbuka dan lisensi

perizinan pada Android memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator nirkabel, dan pengembang aplikasi. Selain itu, Android memiliki sejumlah besar komunitas pengembang aplikasi (apps) yang memperluas fungsionalitas perangkat, umumnya ditulis dalam versi kustomisasi bahasa pemrograman Java. Pada bulan Oktober 2013, ada lebih dari satu juta aplikasi yang tersedia untuk Android, dan sekitar 50 miliar aplikasi telah diunduh dari Google Play, toko aplikasi utama Android. Sebuah survei pada bulan April-Mei 2013 menemukan bahwa Android adalah platform paling populer bagi para pengembang, digunakan oleh 71% pengembang aplikasi bergerak. Di Google I/O 2014, Google melaporkan terdapat lebih dari satu miliar pengguna aktif bulanan Android, meningkat dari 583 juta pada bulan Juni 2013.

Faktor-faktor di atas telah memberikan kontribusi terhadap perkembangan Android, menjadikannya sebagai sistem operasi telepon pintar yang paling banyak digunakan di dunia, mengalahkan Symbian pada tahun 2010. Android juga menjadi pilihan bagi perusahaan teknologi yang menginginkan sistem operasi berbiaya rendah, bisa dikustomisasi, dan ringan untuk perangkat berteknologi tinggi tanpa harus mengembangkannya dari awal. Sifat Android yang terbuka juga telah mendorong munculnya sejumlah besar komunitas pengembang aplikasi untuk menggunakan kode sumber terbuka sebagai dasar proyek pembuatan aplikasi, dengan menambahkan fitur-fitur baru bagi pengguna tingkat lanjut atau mengoperasikan Android pada perangkat yang secara resmi dirilis dengan menggunakan sistem operasi lain.

Pada November 2013, Android menguasai pangsa pasar telepon pintar global, yang dipimpin oleh produk-produk Samsung, dengan persentase 64% pada bulan Maret 2013. Pada Juli 2013, terdapat 11.868 perangkat Android berbeda dengan beragam versi. Keberhasilan sistem operasi ini juga menjadikannya sebagai target litigasi paten "perang telepon pintar" antar perusahaan-perusahaan teknologi. Hingga bulan Mei 2013, total 900 juta perangkat Android telah diaktifkan di seluruh dunia, dan 48 miliar aplikasi telah dipasang dari Google Play[22].

### 2.8.3 Arsitektur Android

Android dibangun dengan menggunakan asas object oriented, dimana elemen-elemen penyusun sistem operasinya berupa objek yang dapat kita gunakan kembali/reusable. Agar bisa membuat aplikasi dengan baik, tentunya kita harus mengetahui arsitektur OS Android beserta elemen elemennya. Gambar 2.5.1 merupakan skema pembagian elemen pada arsitektur Android. Secara garis besar arsitektur android terdiri dari empat layer komponen, yaitu:



**Gambar 2. 8 Komponen Utama Sistem Operasi Android**

Sumber : Pembangunan Aplikasi Manajemen Event ( Studi Kasus di Wah Wah Event ) [10]

Sistem Operasi Android memiliki komponen utama sebagai berikut:

1. Aplikasi Android berisi

Sekumpulan aplikasi utama seperti: email client, program Short Message Service (SMS), kalender, peta, browser, daftar kontak, dan lain-lain. Semua aplikasi ditulis dengan menggunakan bahasa pemrograman Java atau kotlin.

## 2. Kerangka kerja aplikasi

Kerangka kerja aplikasi yang ditulis dengan menggunakan bahasa pemrograman Java merupakan peralatan yang digunakan oleh semua aplikasi, baik aplikasi bawaan dari ponsel seperti daftar kontak, dan kotak SMS, maupun aplikasi yang ditulis oleh Google ataupun pengembang Android. Android menawarkan para pengembang kemampuan untuk membangun aplikasi yang inovatif. Pengembang bebas untuk mengambil keuntungan dari perangkat keras, akses lokasi informasi, menjalankan background services, mengatur alarm, menambahkan peringatan ke status bar, dan masih banyak lagi. Pengembang memiliki akses yang penuh ke dalam kerangka kerja API yang sama yang digunakan oleh aplikasi utama. Pada dasarnya, kerangka kerja aplikasi memiliki beberapa komponen sebagai berikut:

- a. *Activity manager*, mengatur siklus dari aplikasi dan menyediakan navigasi backstack untuk aplikasi yang berjalan pada proses yang berbeda.
- b. *Package manager*, untuk melacak aplikasi yang di-instal pada perangkat.
- c. *Windows manager*, merupakan abstraksi dari bahasa pemrograman Java pada bagian atas dari level services (pada level yang lebih rendah) yang disediakan oleh Surface Manager.
- d. *Telephony manager*, Berisi sekumpulan API yang diperlukan untuk memanggil aplikasi.
- e. *Content providers*, digunakan untuk memungkinkan aplikasi mengakses data dari aplikasi lain (seperti contacts) atau untuk membagikan data mereka sendiri.
- f. *Resource manager*, digunakan untuk mengakses sumber daya yang bersifat bukan code seperti string lokal, bitmap, deskripsi dari layout file dan bagian eksternal lain dari aplikasi.
- g. *View system*, digunakan untuk mengambil sekumpulan button, list, grid, dan text box yang digunakan di dalam antarmuka pengguna.
- h. *Notification manager*, digunakan untuk mengatur tampilan peringatan dan fungsi-fungsi lain.

### 3. *Libraries*

Android memiliki sekumpulan library C/C++ yang digunakan oleh berbagai komponen dalam sistem Android. Kemampuan-kemampuan ini dilihat oleh para pengembang melalui kerangka kerja aplikasi. Beberapa dari library utama dijelaskan sebagai berikut:

- a. *System C Library*, Merupakan implementasi turunan dari standar system library C (libc) yang diatur untuk peralatan berbasis embedded Linux.
- b. *Media Libraries*, disediakan oleh PacketVideo (salah satu anggota dari OHA) yang memberikan library untuk memutar ulang dan menyimpan format suara dan video, serta static image file seperti MPEG4, MP3, AAC, AMR, JPG, dan PNG.
- c. *Surface manager*, mengatur akses ke dalam subsistem tampilan dan susunan grafis layer 2D dan 3D secara mulus dari beberapa aplikasi dan menyusun permukaan gambar yang berbeda pada layar ponsel.
- d. *LibWebCore*, Merupakan web browser modern yang menjadi kekuatan bagi browser Android dan sebuah embeddable web view.
- e. *Scalable Graphics Library (SGL)*, SGL mendasari mesin grafis 2D dan bekerja bersama-sama dengan lapisan pada level yang lebih tinggi dari kerangka kerja (seperti Windows Manager dan Surface Manager) untuk mengimplementasikan keseluruhan graphics pipeline dari Android.
- f. *3DLibraries*, Implementasi yang didasarkan pada OpenGL ES 1.0 APIs dimana library menggunakan baik akselerasi perangkat keras 3D (jika tersedia) ataupun yang disertakan, dengan rasterisasi perangkat lunak 3D yang sangat optimal.
- g. *FreeType Library*, Digunakan untuk menghaluskan semua tulisan bitmap dan vektor.
- h. *SQLite*, Merupakan relational database yang kuat dan ringan serta tersedia untuk semua aplikasi.

#### 4. *Android Runtime*

Merupakan lokasi dimana komponen utama dari DVM ditempatkan. DVM dirancang secara khusus untuk Android pada saat dijalankan pada lingkungan yang terbatas, dimana baterai yang terbatas, CPU, memori, dan penyimpanan data menjadi fokus utama. Android memiliki sebuah tool yang terintegrasi yaitu “dx” yang mengkonversi generated byte code dari (.JAR) ke dalam file (.DEX) sehingga byte code menjadi lebih efisien untuk dijalankan pada prosesor yang kecil. Hal ini memungkinkan untuk memiliki beberapa jenis dari DVM berjalan pada suatu peralatan tunggal pada waktu yang sama. Core libraries ditulis dalam bahasa Java dan berisi kumpulan class, I/O dan peralatan lain.

#### 5. Linux Kernel

Arsitektur Android berdasarkan pada Linux 2.6 kernel yang dapat digunakan untuk mengatur keamanan, manajemen memori, manajemen proses, network stack, dan driver model. Kernel juga bertindak sebagai lapisan abstrak antara perangkat keras dan seluruh software stack[10].

### 2.8.4 **Android SDK**

Android SDK adalah tools API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, middleware dan aplikasi kunci yang direlease oleh Google. Saat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java sebagai platform aplikasi netral, Android memberi kesempatan untuk membuat aplikasi yang butuhkan yang bukan merupakan aplikasi bawaan Handphone/Smartphone[23].

### 2.8.5 **Java**

Menurut Budi Raharjo , Imam Heryanto, Arif haryono (Mudah Belajar Java 2010) java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer 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 berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/nonspesifik (general purpose), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi Java mampu berjalan di beberapa platform sistem operasi yang berbeda, Java dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun". Saat ini Java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web. Adapun kelebihan dari Java yaitu :

1. Multiplatform.

Kelebihan utama dari Java ialah dapat dijalankan di beberapa platform / sistem operasi komputer, sesuai dengan prinsip tulis sekali, jalankan di mana saja. Dengan kelebihan ini pemrogram cukup menulis sebuah program Java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin / bytecode) sekali lalu hasilnya dapat dijalankan di atas beberapa platform tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis Java dikerjakan diatas operating system Linux tetapi dijalankan dengan baik di atas Microsoft Windows. Platform yang didukung sampai saat ini adalah Microsoft Windows, Linux, Mac OS dan Sun Solaris. Penyebabnya adalah setiap sistem operasi menggunakan programnya sendiri-sendiri (yang dapat diunduh dari situs Java) untuk meninterpretasikan bytecode tersebut.

2. OOP (Object Oriented Programming - Pemrogram Berorientasi Objek)
3. Perpustakaan Kelas Yang Lengkap

Java terkenal dengan kelengkapan library/perpustakaan (kumpulan program program yang disertakan dalam pemrograman Java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas Java yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.

#### 4. Bergaya C++

Memiliki sintaks seperti bahasa pemrograman C++ sehingga menarik banyak pemrogram C++ untuk pindah ke Java. Saat ini pengguna Java sangat banyak, sebagian besar adalah pemrogram C++ yang pindah ke Java. Universitas-universitas di Amerika Serikat juga mulai berpindah dengan mengajarkan Java kepada murid-murid yang baru karena lebih mudah dipahami oleh murid dan dapat berguna juga bagi mereka yang bukan mengambil jurusan komputer.

#### 5. Pengumpulan sampah otomatis

Memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa C++ yang dipakai secara luas)[24].

### **2.8.6 Kotlin**

Kotlin adalah bahasa pemrograman yang diketik secara statis yang menargetkan Java virtual machine (JVM), Android, JavaScript, dan Native. Kotlin dikembangkan oleh JetBrains, proyek Kotlin dimulai pada tahun 2010 dan merupakan Bahasa pemrograman open source. Kotlin versi 1.0 secara resmi dirilis pada bulan Februari 2016. Dalam mengembangkan aplikasi Android, Kotlin menawarkan beberapa fitur yaitu:

1. **Kompatibilitas:** Kotlin sepenuhnya kompatibel dengan JDK 6, sehingga Kotlin dapat berjalan pada perangkat Android versi lama. Kotlin juga sepenuhnya didukung di Android Studio dan kompatibel dengan Android build system.

2. Performa: Sebuah aplikasi Kotlin dapat berjalan secepat Java, karena memiliki struktur bytecode yang sangat mirip. Dengan dukungan Kotlin untuk fungsi inline, kode menggunakan lambda sering berjalan lebih cepat daripada kode yang sama yang ditulis di Java.
3. Interoperabilitas: Kotlin 100% dapat beroperasi dengan Java, memungkinkan programmer untuk menggunakan semua Android Library dalam aplikasi, termasuk pengolahan anotasi, sehingga databinding dan Dagger dapat bekerja.
4. Footprint: Kotlin memiliki runtime library yang sangat padat/tersusun rapat, yang ke depannya dapat dikurangi melalui penggunaan ProGuard. Dalam aplikasi nyata, runtime pada Kotlin hanya menambahkan beberapa ratus metode dan kurang dari 100K untuk ukuran dari file .apk.
5. Waktu kompilasi: Kotlin mendukung kompilasi Inkremental yang efisien, sehingga sementara ada beberapa tambahan overhead untuk clean build, pembangunan bertahap ini biasanya sama cepat atau lebih cepat daripada dengan Java.
6. Learning Curve: untuk pengembang Java, memulai menggunakan Kotlin sangat mudah. Automated Java to Kotlin converter yang terdapat dalam plugin Kotlin membantu pengembang Java pada langkah pertama. Terdapat juga Kotlin Koans yang menawarkan panduan dalam penggunaan Bahasa pemrograman ini[25].

## 2.9 Go

Golang atau yang sering disebut Go, adalah bahasa pemrograman open source yang memudahkan dalam membangun software yang sederhana, andal dan efisien. Go adalah proyek open source untuk membuat program lebih produktif.(Golang.org).

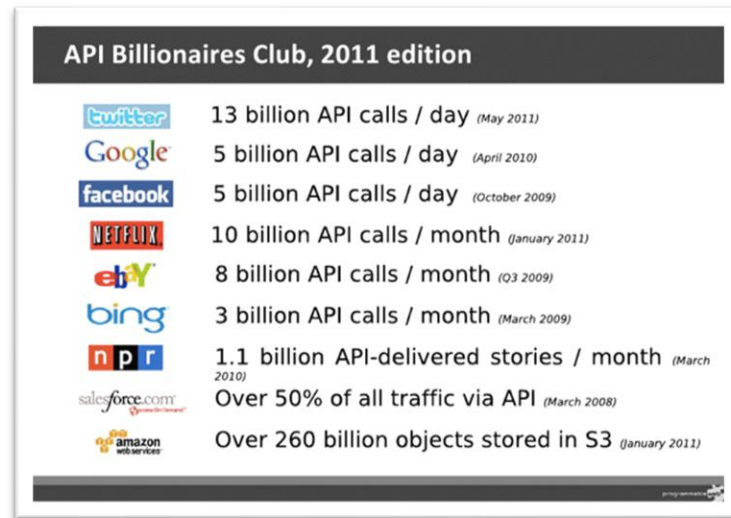
Go mulai dikembangkan pada 2007 di Google, dan diperkenalkan ke publik pada tahun 2009. Tiga pengembang utama Go di Google adalah Robert Griesemer, Rob Pike, dan Ken Thompson. Tujuan mereka adalah membuat sebuah bahasa,

berdasarkan sintaks bahasa pemrograman C yang longgar, yang akan menghilangkan “extraneous garbage” bahasa seperti C++. Oleh karena itu, Go menjauhkan banyak fitur dari bahasa modern lainnya, seperti metode dan operator overloading, aritmatika pointer dan jenis warisan. Go bukan sebuah Bahasa free-form, konvensinya menentukan banyak detail pemformatan, termasuk bagaimana identasi dan spasi digunakan. Bahasa tersebut mensyaratkan bahwa tidak satupun variabel yang dideklarasikan atau libraries yang diimpor tidak digunakan, dan semua pernyataan return bersifat wajib[26].

## **2.10 Application Programming Interface (API)**

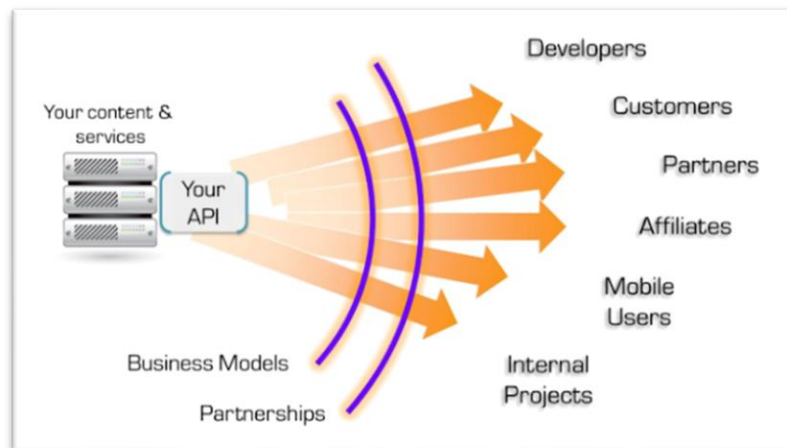
Application Programming Interface (API) adalah Sebuah bahasa dan pesan format yang digunakan oleh program aplikasi untuk berkomunikasi dengan sistem operasi atau program pengendalian lainnya seperti sistem manajemen database (DBMS) atau protokol komunikasi. API diimplementasikan dengan menulis panggilan fungsi dalam program, yang menyediakan hubungan ke subrutin yang diperlukan untuk eksekusi. Dengan demikian, API menyiratkan bahwa driver atau program modul tersedia dalam komputer untuk melakukan operasi atau perangkat lunak yang harus dihubungkan ke dalam program yang ada untuk melakukan tugas-tugas. API dapat menjelaskan cara sebuah tugas (task) tertentu dilakukan. Dalam pemrograman procedural seperti bahasa C, aksi biasanya dilakukan dengan media pemanggilan fungsi. Karena itu, API biasanya menyertakan penjelasan dari fungsi/rutin yang disediakannya.

API (Application Programming Interface) merupakan interface yang digunakan untuk mengakses aplikasi atau layanan dari sebuah program. API memungkinkan pengembangan untuk memakai fungsi yang sudah ada dari aplikasi lain sehingga tidak perlu membuat ulang dari awal



**Gambar 2. 9 API Billionaires Club, 2011**

Sumber : Rancang Bangun Application Programming Interface (Api) Ongkos Kirim Dan Tracking Ekspedisi Indonesia [27]



**Gambar 2. 10 API**

Sumber : Rancang Bangun Application Programming Interface (Api) Ongkos Kirim Dan Tracking Ekspedisi Indonesia [27]

Dari gambar 2.9 dan 2.10 dapat dilihat bahwa API banyak digunakan dan memudahkan dalam kegiatan bisnis. Keuntungan memprogram dengan menggunakan API adalah:

- a. Portabilitas. Programmer yang menggunakan API dapat menjalankan programnya dalam sistem operasi mana saja asalkan sudah ter- install API tersebut. Sedangkan system call berbeda antar sistem operasi, dengan catatan dalam implementasinya mungkin saja berbeda.
- b. Lebih Mudah Dimengerti. API menggunakan bahasa yang lebih terstruktur dan mudah dimengerti daripada bahasa system call. Hal ini sangat penting dalam hal editing dan pengembangan[27].

### **2.10.1 API TESTING**

Testing API dilakukan untuk sistem yang memiliki banyak API untuk di tes. Pengecekan dilakukan dengan mensimulasikan penggunaan API oleh pengguna. Ada beberapa harus diperhatikan dalam pengecekan API yakni:

- a. Menjelajahi kondisi batas dan memastikan bahwa tes memanfaatkan berbagai parameter dari API panggilan dengan cara yang memverifikasi fungsi dan mengekspos kegagalan.
- b. Menghasilkan kombinasi nilai parameter yang menarik untuk panggilan dengan dua parameter atau lebih.
- c. Memverifikasi perilaku API yang mempertimbangkan kondisi lingkungan eksternal seperti file, perangkat perifer, dan sebagainya
- d. Memverifikasi Urutan panggilan API dan memeriksa apakah API menghasilkan hasil yang bermanfaat dari panggilan berturut-turut.

Secara umum pengecekan yang dilakukan terhadap API yaitu

- a. Nilai yang diberikan sesuai kondisi yang diberikan.
- b. Memastikan bahwa API tidak memberikan nilai apapun selain kondisi
- c. Memastikan jika API memicu panggilan terhadap API lain, pengeluaran harus bias dilacak dan diverifikasi.
- d. Memastikan jika API mengubah struktur data[27].

## 2.11 UML

Bahasa spesifikasi standar untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak. Definisi lain juga menyatakan *Unified Modeling Language (UML)* adalah himpunan struktur dan teknik untuk pemodelan desain program berorientasi objek (*OOP*) serta aplikasinya.

UML menyediakan beberapa macam diagram untuk memodelkan aplikasi berorientasi objek, yaitu diantaranya :

1. *Use Case Diagram*
2. *Class Diagram*
3. *Behaviour Diagram*
  - a. *Activity Diagram*
  - b. *Statechart Diagram*
  - c. *Interaction Diagram*
    - i. *Sequence Diagram*
    - ii. *Collaboration Diagram*
2. *Implementation Diagram*
  - a. *Component Diagram*
  - b. *Deployment Diagram*

Berikut akan dijelaskan 3 macam diagram yang penulis gunakan dalam pembangunan aplikasi berorientasi object, yaitu *use case diagram*, *class diagram*, *sequence diagram* dan *Activity diagram*[9].


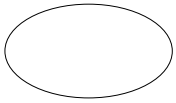
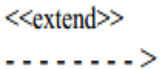
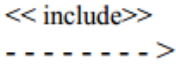

### 2.11.1 Diagram Use Case (*Use Case Diagram*)

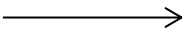
*Use Case diagram* adalah model fungsional sebuah sistem yang menggunakan *actor* dan *case* (proses). *Use Case* adalah layanan (*services*) atau fungsi-fungsi yang disediakan oleh sistem untuk pengguna-penggunanya. *Use Case* adalah suatu pola atau gambaran yang menunjukkan kelakuan atau kebiasaan sistem. Setiap *Use Case* adalah suatu urutan (*sequence*) transaksi yang saling berhubungan dan dilakukan oleh sebuah actor dan sistem dalam bentuk sebuah *dialog*. *Use Case diagram* mempresentasikan kegunaan atau fungsi-fungsi sistem dari perspektif pengguna.

*Use case diagram* menggambarkan apa yang dilakukan aktor terhadap sistem. *Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Sebuah *use case* dapat meng-*include* fungsionalitas *use case* lain sebagai bagian dari proses dalam *use case* dirinya. Sebuah *use case* juga dapat melakukan *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. Tabel 2.2 di bawah ini memuat notasi simbol-simbol yang digunakan dalam diagram use case[9].



Tabel 2. 2 Simbol dalam *Diagram Use Case*

No	Lambang	Nama	Keterangan
1		<i>Actor</i>	Aktor ( <i>actor</i> ) merupakan sebuah entitas yang berinteraksi dengan <i>use case</i> . Nama aktor dituliskan di bawah gambar tersebut. Aktor dapat berupa orang atau sistem lain diluar sistem yang tengah dianalisis.
2		<i>Use Case</i>	<i>Use case</i> menggambarkan sebuah fungsi tertentu yang disediakan oleh sistem, sebuah subsistem atau urutan pertukaran pesan antar anggota sistem dan satu atau lebih aktor melakukan aksi yang dikerjakan oleh sistem.
3		<i>Extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>usecase</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan.
4		<i>Include</i>	Relasi <i>use case</i> yang dimana proses bersangkutan akan dilanjutkan ke proses yang dituju.
5		<i>Asosiasi</i>	Komunikasi antara <i>actor</i> dan <i>use case</i> yang berpartisipasi pada


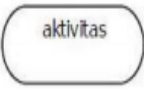
			<i>usecase</i> atau <i>usecase</i> memiliki interaksi dengan actor
6		<i>Generalisasi</i>	Dimana fungsi yang satu adalah fungsi yang umum dari yang lainnya.




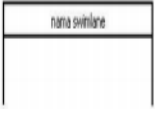


### 2.11.2 Activity Diagram

Pada dasarnya diagram *Activity* sering digunakan oleh *flowchart*. Diagram ini berhubungan dengan diagram *Statechart*. Diagram *Statechart* berfokus pada objek dalam suatu proses (atau proses menjadi suatu objek), diagram *Activity* berfokus pada aktivitas-aktivitas yang terjadi yang terkait dalam suatu proses tunggal. Jadi dengan kata lain, diagram ini menunjukkan bagaimana aktivitas-aktivitas tersebut bergantung satu sama lain.

Proses berawal dari lingkaran start hitam pada bagian atas dan berakhir di pusat lingkaran *stop* hitam atau putih pada bagian bawah. Aktivitas digambarkan dalam bentuk kotak persegi. Berikut adalah simbol yang digunakan dalam *activity diagram* [9]:

**Tabel 2. 3** Simbol *Activity Diagram*

No	Simbol	Nama	Keterangan
1		Status Awal	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
2		Aktivitas	Aktivitas yang dilakukan sistem, biasanya diawali dengan kata kerja

3		Pecabangan atau <i>decision</i>	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
4		Penggabungan atau <i>join</i>	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
5		Status akhir	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir
6		<i>Swimlane</i>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi
7		<i>Fork</i>	Digunakan untuk menunjukkan kegiatan yang dilakukan secara <i>parallel</i>
8		<i>Join</i>	Digunakan untuk menunjukkan kegiatan yang digabungkan

### 2.11.3 Class Diagram

*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut atau properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda atau fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-

lain. *Class* memiliki tiga area pokok : nama, atribut, dan Metoda. Atribut dan metoda dapat memiliki salah satu sifat berikut:

1. *Private*, tidak dapat dipanggil dari luar class yang bersangkutan.
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan anak-anak yang mewarisinya
3. *Public*, dapat dipanggil oleh siapa saja.

*Class* dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metode. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*.

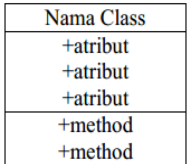




Dengan demikian *interface* mendukung resolusi metode pada saat *run time*. Sesuai dengan perkembangan *class* model, *class* dapat dikelompokkan menjadi *package*.


Hubungan antar *Class* terdiri dari sebagai berikut:

1. Asosiasi, yaitu hubungan statis antar class. Umumnya menggambarkan class yang memiliki atribut berupa class lain atau class yang harus mengetahui eksistensi class lain.
2. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas”).
3. Pewarisan, yaitu hubungan hirarkis antar class. Class dapat diturunkan dari class lain dan mewarisi semua atribut dan metoda class asalnya dan menambahkan fungsionalitas baru, sehingga bisa disebut anak dari class yang diwarisinya. Kebalikan dari pewarisan adalah generalisasi.
4. Hubungan dinamis, yaitu rangkaian pesan (message) yang dipassing dari satu class kepada class lain.

Berikut adalah simbol yang dimiliki class diagram :

**Tabel 2. 4** Simbol *Class Diagram*

No	Simbol	Nama	Keterangan
1		Kelas	Kelas adalah bentuk umum dari objek-objek yang memiliki struktur dan tingkah laku (atribut, operasi, relasi dan arti/definisi) yang sama. Kelas merupakan bentuk abstrak dari objek di dunia nyata.
2		Antar muka atau <i>interface</i>	Sama dengan konsep <i>interface</i> dalam pemograman berorientasi objek.
3		Asosiasi atau <i>association</i>	Relasi antar kelas dengan makna umum, asosiasi biasanya disertai dengan <i>multiplicity</i>
4		Asosiasi berarah atau <i>directed association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya disertai <i>multiplicity</i> .
5		<i>dependency</i>	Relasi antar kelas dengan makna kebergantungan antar kelas.

6		Agregasi atau <i>aggregation</i>	Relasi antar kelas dengan makna semua bagian ( <i>whole part</i> ).
---	---	--	---

#### 2.11.4 *Sequence Diagram*


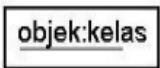


*Interaction* diagram menunjukkan bagaimana kelompok-kelompok objek saling berkolaborasi dalam beberapa *behavior*. UML memiliki beberapa bentuk *interaction diagram* dan yang paling umum digunakan adalah *sequence diagram*. Sebuah *sequence diagram* adalah diagram yang secara khusus menjabarkan urutan *behavior* sebuah skenario tunggal. Diagram tersebut menunjukkan sejumlah objek contoh dan pesan-pesan yang mewakili objek ini di dalam *use case*.

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi/metode dari *class*. *Activation bar* menunjukkan

lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message[9].

Adapun komponen-komponen yang digunakan dalam *sequence diagram* dapat dilihat pada Tabel 2.5

**Tabel 2. 5** Simbol *Sequence Diagram*

No	Simbol	Nama	Keterangan
1		<i>Lifeline</i>	a. Menandakan kehidupan objek selama urutan. b. Diakhiri tanda X pada titik dimana kelas tidak lagi berinteraksi.
2		Objek	a. Berpartisipasi secara berurutan dengan mengirimkan atau menerima pesan. b. Ditempatkan di bagian atas diagram.
3		Aktor	a. Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi dan mendapat manfaat dari sistem. b. Berpartisipasi secara berurutan dengan mengirimkan atau menerima pesan. c. Ditempatkan dibagian atas diagram.
4		Objek sedang aktif	Fokus control : a. Adalah persegi panjang yang sempit panjag ditempatkan di atas sebuah garis hidup.

		berinteraksi	b. Menandakan ketika suatu objek mengirim atau menerima pesan.
5		Pesan	Objek mengirim satu pesan ke objek lainnya.
6		1:masukan	Menyatakan bahwa suatu objek mengirimkan masukan ke objek lainnya. Arah panah mengarah pada objek yang dikirim.
7		1:keluaran	Objek atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.
8		Destroy ( )	Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> .