

BAB II

LANDASAN TEORI

2.1 Sistem Penjualan *Taking Order*

Menurut (Vinno Christmantara, 2015) Sistem penjualan taking order adalah metode penjualan yang sangat praktis, sebab produk-produk yang dipesan oleh pelanggan sudah siap untuk dikirimkan. Apabila ada barang yang kosong, salesman sudah dapat menginformasikan segera. Metode taking order ini adalah salah satu cara memenuhi order pelanggan sesuai kebutuhannya. Berapa banyak barang yang dikirim sangat berhubungan dengan banyaknya faktur penjualan.

2.2 Tempat Makan dan *Taking Order*

Secara umum, tempat makan atau restoran merupakan tempat yang dikunjungi orang untuk mencari berbagai macam makanan dan minuman. Restoran biasanya juga menyuguhkan keunikan tersendiri sebagai daya tariknya, baik melalui menu masakan, hiburan maupun tampilan fisik bangunan. Di dalam restoran ada beberapa aktivitas penting salah satunya adalah Pemesanan atau *Taking Order*. *Taking Order* menurut (Dian Galih, 2013) “adalah kegiatan menerima dan mencatat pesanan tamu, dalam hal ini makanan dan minuman, yang selanjutnya akan diteruskan ke bagian yang terkait, antara lain dapur, bar, dan kasir”. *Taking Order* meliputi beberapa kegiatan antara lain:

- a. Menampilkan informasi yang akurat mengenai semua makanan dan minuman yang tersedia di daftar menu.
- b. Mencatat menu yang dipesan, jumlah yang dipesan, nama pemesan dan lainnya.
- c. Mengkonfirmasi pesanan kepada pelanggan.
- d. Meneruskan pesanan ke bagian terkait.

2.2.1 Pengertian Taking Order

Menurut (Bagyono, 2006) Taking Order dapat diartikan menerima dan mencatat pesanan tamu, dalam hal ini makanan dan minuman, yang selanjutnya akan diteruskan ke bagian yang terkait, antara lain : kitchen , bar, dan pastry. Taking order merupakan bagian dari service, biasa dilakukan di room service, restaurant, bar, dan banquet, Petugas yang melakukan taking order disebut order taker.

Seorang order taker dituntut untuk :

- a. Menguasai bahasa asing terutama bahasa inggris
- b. Mengetahui pengetahuan food & beverage
- c. Menguasai technics of selling (teknik menjual)
- d. Menguasai technics in writting the order (teknik mencatat pesanan)
- e. Memiliki memori yang baik

2.2.2 Tujuan Taking Order

Menurut (Bagyono, 2006) Adapun beberapa tujuab dari talking order antara lain sebagai berikut:

- a. Memberikan informasi akurat mengenai semua makanan dan minuman yang tersedia dan terdaftar di daftar menu
- b. Meningkatkan penjualan
- c. Meningkatkan pendapatan
- d. Meningkatkan keuntungan

2.2.3 Taking Order di Room Service

Menurut (Bagyono, 2006) Taking order di room service dilakukan oleh order taker, dibawah kontrol captain, assistant head waiter, dan head waiter manager. Hal-hal yang perlu diperhatikan dan dicatat antara lain :

- a. Jenis menu yang dipesan
- b. Jumlah tamu
- c. Nama tamu yang menginap dikamar tersebut
- d. Menggunakan nampan atau trolley

- e. VIP guest atau regular guest
- f. Peralatan dan segala sesuatu yang harus disiapkan
- g. Cara service (on plate, bowl, glass, heater, dsb)
- h. Cara clear-up
- i. Cara pembayaran
- j. Jam / waktu pengantaran pesanan yang diminta

2.2.4 Taking Order di Restaurant dan Bar

Taking order bisa dilakukan oleh head waiter, assistant head waiter, captain, waiter, bartender. Hal-hal yang perlu diperhatikan dan dicatat antara lain :

- a. Macam menu yang dipesan : buffet, a'la carte atau table d'hotel.
- b. Tanggal, nomor meja
- c. Jumlah tamu
- d. Cara menulis order
- e. Timing of cooking (waktu yang dibutuhkan untuk memasak)
- f. Tamu VIP atau biasa
- g. Cara clear-up
- h. Cara payment

2.2.5 Kapan Melakukan Taking Order (di Restaurant dan Bar)

- a. Sesudah menyambut tamu
- b. Sesudah tamu duduk
- c. Sesudah pramusaji menuangkan minuman air es
- d. Sesudah memberikan / mempresentasikan menu
- e. Sesudah tamu membaca menu

2.2.6 Teknik Melakukan Taking Order

- a. Sambil membawa captain order book dan alat tulisnya, mendekatlah kesisi kiri tamu, tersenyum dan memberi salam.

- b. Untuk tamu yang jumlahnya lebih dari 2 orang, tanyakan terlebih dahulu apakah pemesanan dilakukan oleh masing-masing tamu atau satu tamu yang bertindak sebagai tuan rumah yang memilhkan pesanan untuk semua tamu.
- c. Berikan saran makanan dan minuman apa saja yang paling sesuai dengan selera tamu
- d. Tulislah pesanan mereka secara jelas dan sistematis untuk memudahkan pihak kitchen dan bar dalam menyajikan pesanan sesuai instruksi yang tertulis di captain order book.
- e. Untuk tamu yang jumlahnya lebih dari 2, tandailah salah satu tamu dengan nomor kode sebagai patokan, misalnya nomor 1, kemudian tamu disebelahnya nomor 2, begitu seterusnya searah jarum jam.
- f. Dalam menulis pesanan tamu di captain order book, gunakan istilah atau singkatan kata yang biasa dipakai dan dipahami baik oleh bagian kitchen, bar, maupun pramusaji.
- g. Ulangi pesanan tamu dengan membacakan kembali untuk meyakinkan kebenaran pesanannya sebelum diproses lanjut.
- h. Tanyakan apakah bill nantinya dibuat jadi satu untuk semua pesanan atau terpisah untuk masing-masing tamu.

2.2.7 Cara Taking Order yang Baik dan Benar

Pengetahuan dan Kedisiplinan dalam cara penulisan saat taking order sangat menentukan kecepatan dan keakuratan penyajian order atau pesanan makanan dan minuman. Jika terjadi kesalahan sedikit saja dalam penulisan order atau pesanan tamu, maka sudah dipastikan akan terjadi Complaint dari tamu karena ke-tidak-sesuaian antara apa yang dipesan dan disajikan. Disini peran Waiter dan Waiteress sangat menentukan mutu layanan sebuah Restoran.

Disini, jelas sangat diperlukan Pengetahuan dan Kedisiplinan dalam Taking Order. Untuk itu, Restofocus mencoba memberikan cara taking order yang baik dan benar yang berguna bagi staff waiter dalam menjalankan tugas nya setiap hari.

Adapun cara taking order yang baik dan benar adalah sebagai berikut :

1. Berdiri sempurna minimal 1 langkah dari meja tamu. Jangan menekuk atau melipat kaki. Sikap anda yang disiplin akan memberi kesan yang baik kepada tamu.
2. Tanyakan dengan sopan apa pesanan tamu. Berbicaralah dengan nada yang halus dan tidak terlalu keras. Jangan lupa untuk tersenyum sebagai tanda anda senang melayani tamu.
3. Dengarkan dengan seksama dan tuliskan di buku atau bill Order. Tuliskan dengan rinci jika ada pesanan yang bersifat khusus misalnya tidak pakai Cabe, tidak pakai Merica, dan hal lainnya.
4. Pisahkan antara pesanan Makanan dan Minuman. Berilah jarak baris untuk memudahkannya. Ini berguna saat nanti anda mengulang pesanan tamu, dengan susunan menu makanan dan minuman, tamu akan terinformasikan dengan baik mengenai apa saja yang telah mereka pesan.
5. Konfirmasikan pesanan tamu. Setelah tamu selesai pesan, maka bacakanlah kembali apa yang sudah dicatat di buku atau bill Order untuk konfirmasi apabila terjadi kesalahan atau kekeliruan pencatatan pesanan.
6. Jika tamu sudah mengiyakan atau menyatakan bahwa apa yang sudah dibacakan itu benar. barulah katakan terima kasih untuk pesannya dan permisi untuk meninggalkan tamu. Terkadang ada tamu yang sibuk mengobrol atau bicara sendiri sebelum anda mendapat kepastian dari tamu apakah pesannya sudah benar atau belum. Jangan sungkan untuk menanyakannya kembali kepada tamu untuk menghindari kesalahan. Ini jauh lebih baik daripada anda disalahkan karena salah mencatat pesanan karena dianggap tidak konfirmasi.

2.3 Zhafar Donut

Zhafar Donat resmi didirikan pada tanggal 19 September 2016. Zhafar Donut merupakan salah satu toko donat yang didirikan oleh seorang wirausahawan yang bertujuan untuk memasarkan donat buatan sendiri. Banyaknya pemesanan secara tidak langsung yang membuat peneliti memilih toko Zhafar Donut sebagai tempat

penelitian karena pada toko ini menyediakan inovasi pada donat, dengan memberikan kata-kata atau ucapan yang ditulis diatas donat.

2.4 Perangkat Lunak yang Digunakan

Menurut (Pressman, 2002) Rekayasa Perangkat Lunak adalah pembuatan dan penggunaan prinsip-prinsip keahlian teknik untuk mendapatkan perangkat lunak yang ekonomis yang handal dan bekerja secara efisien pada mesin yang sesungguhnya.

2.4.1 Dream Weaver CS 5

Adobe Dreamweaver merupakan program penyunting halaman web keluaran Adobe Systems yang dulu dikenal sebagai Macromedia Dreamweaver keluaran Macromedia. Program ini banyak digunakan oleh pengembang web karena fitur-fiturnya yang menarik dan kemudahan penggunaannya. Versi terakhir Macromedia Dreamweaver sebelum Macromedia dibeli oleh Adobe Systems yaitu versi 8. Adobe Dreamweaver CS5 yang dirilis pada 12 April 2011, sedangkan versi terbaru dari Dreamweaver yaitu Adobe Dreamweaver CS6 yang dirilis pada 21 April 2012.

Menurut (Madcoms, 2011) memberikan batasan bahwa “dreamweaver adalah sebuah HTML editor profesional untuk mendesain web secara visual dan mengelola situs atau halaman web”. Saat ini terdapat software dari kelompok adobe yang belakangan banyak digunakan untuk mendesain suatu situs web. Versi terbaru dari Adobe Dreamweaver saat ini adalah Dreamweaver CS5, terdapat beberapa kemampuan bukan hanya sebagai software untuk desain web saja tetapi juga untuk menyunting kode serta pembuatan aplikasi web dengan menggunakan berbagai bahasa pemrograman web, antara lain: JPS, PHP, ASP dan *ColdFusion*.

2.4.2 Tampilan Ruang Kerja Dreamweaver

Tampilan ruang kerja pada dreamweaver CS5 diantaranya:

1. Aplikasi Bar

Berada dibagian paling atas jendela aplikasi Dreamweaver CS5. Baris ini berisi tombol *workspace (workspace switcher)*, menu, dan aplikasi lainnya.

2. *Toolbar* Dokumen

Berisi tombol-tombol yang digunakan untuk mengubah tampilan jendela dokumen, sebagai contoh tampilan *design* atau tampilan kode. Juga dapat digunakan untuk operasi-operasi umum, misalnya untuk melihat hasil sementara halaman *web* pada jendela *browser*.

3. Jendela Dokumen

Lembar kerja tempat anda membuat dan mengedit desain halaman *web*.

4. *Workspace Switcher*

Digunakan untuk mengubah tampilan ruang kerja (*workspace*) Dreamweaver CS5.

5. *Panel Groups*

Adalah kumpulan panel yang saling berkaitan, panel-panel ini dikelompokkan pada judul-judul tertentu berdasarkan fungsinya. Panel ini digunakan untuk memonitor dan memodifikasi pekerjaan.

6. *Tag Selector*

Diletakkan dibagian bawah jendela dokumen, satu baris dengan status bar. Bagian ini menampilkan hirarki pekerjaan yang sedang terpilih pada jendela dokumen, dapat juga digunakan untuk memilih objek pada jendela design berdasarkan jenis atau kategori objek tersebut.

7. *Panel Property*

Digunakan untuk melihat dan mengubah berbagai properti objek atau teks pada jendela *design*.

8. *Toolbar Coding*

Berisi tombol-tombol yang digunakan untuk melakukan operasi kode-kode standar

9. *Panel Insert*

Berisi tombol-tombol untuk menyisipkan berbagai jenis objek, seperti *image*, tabel, atau objek media ke dalam jendela dokumen.

10. *Panel Groups*

Adalah kumpulan panel yang saling berkaitan satu sama lainnya yang dikelompokkan dibawah satu judul, seperti: Insert, Files, CSS Style, dan sebagainya.

11. *Panel Files*

Digunakan untuk mengatur file-file dan folder-folder yang membentuk situs web anda.

2.4.3 Pengertian HTML

Program adalah kata, ekspresi, pernyataan atau kombinasi yang disusun dan dirangkai menjadi satu kesatuan prosedur yang berupa urutan langkah untuk menyelesaikan masalah dan diimplementasikan dengan menggunakan bahasa pemrograman sehingga dapat dieksekusi oleh komputer. Pemrograman adalah proses mengimplementasikan urutan langkah untuk menyelesaikan suatu masalah dengan menggunakan suatu bahasa pemrograman. Bahasa pemrograman adalah suatu prosedur penulisan program. Sedangkan pemrograman terstruktur adalah metode untuk mengorganisasikan dan membuat kode-kode program supaya mudah untuk dimengerti, mudah ditest dan mudah dimodifikasi. Salah satu contoh pemrograman adalah pemrograman di internet.

HTML merupakan salah satu unsur penting dalam pemrograman di internet. HTML (*Hypertext Markup Language*) adalah bahasa pengkodean untuk menghasilkan dokumen-dokumen hypertext untuk digunakan di *World Wide Web*. Namun HTML bukanlah sebuah bahasa pemrograman, tetapi hanyalah berisi perintah-perintah yang telah terstruktur berupa tag-tag penyusun. HTML merupakan pengembangan dari standar pemformatan dokumen teks yaitu *Standard Generalized Markup Language (SGML)*. HTML sebenarnya adalah dokumen ASCII atau teks biasa, yang dirancang untuk tidak tergantung pada suatu sistem operasi tertentu.

HTML ini merupakan sebuah format bahasa dari *world wide web (WWW)* untuk menulis dan menampilkan dokumen. Sehingga dokumen tersebut dapat dengan mudah tersebar ke PC melalui media jaringan komputer atau internet.

HTML ini berupa kode tag yang menginstruksikan *browser* untuk menghasilkan tampilan sesuai dengan yang diinginkan. Sehingga, untuk dapat membuka dokumen yang ditulis dengan HTML maka diperlukan sebuah *tool* yaitu *browser* misalnya *Opera*, *Mozilla*, *Google Chrome*, *Netscape*, dan banyak lagi yang lainnya. HTML juga dapat dikenali oleh aplikasi pembuka email ataupun dari PDA dan program lain yang memiliki kemampuan *browser*.

Adapun pengertian dari HTML5 itu sendiri adalah versi terbaru dari HTML (*HyperText Markup Language*) yang mana dikembangkan oleh W3C atau *World Wide Web Consortium*. HTML5 diciptakan dalam pengembangan bahasa HTML untuk memperbaiki konten dan memperbarui teknologi multimedia yang sudah ada pada HTML 4 dan versi sebelumnya agar mudah dijalankan oleh browser dan mudah dimengerti. HTML5 merupakan kerjasama antara *World Wide Web Consortium* (W3C) dan *Web Hypertext Application Teknologi Working Group* (WHATWG). WHATWG bekerja dengan bentuk web dan aplikasi, dan W3C bekerja dengan XHTML 2.0. Tujuan utama pengembangan HTML5 adalah untuk memperbaiki teknologi HTML agar mendukung teknologi multimedia terbaru, mudah dibaca oleh manusia dan juga mudah dimengerti oleh mesin.

Di sisi lain HTML5 sering juga disebut dengan Semantic Web. Semantic web berasal dari bahasa Yunani (Greek), merupakan bahasa pembelajaran semiotic, yakni pembelajaran untuk memahami penanda. Semantic sendiri mempunyai arti bahasa yang berfokus pada penanda untuk mengetahui arti yang terkandung di dalamnya. Jadi, yang dimaksud semantic web adalah bahasa pemrograman yang mempunyai penanda khusus dalam implementasinya dengan tujuan agar mampu mendeskripsikan apa yang terkandung dalam website tersebut. Web semantic ini bukan hanya dikembangkan di Web 2.0, namun sudah ke Web 3.0 bahkan akan berkembang ke Web 4.0. Perkembangan teknologi memang cepat sekali berubah.

2.4.4 XAMPP

Menurut (Wicaksono, 2008) “XAMPP adalah sebuah *software* yang berfungsi untuk menjalankan *website* berbasis PHP dan menggunakan pengolah

data MySQL dikomputer local”. XAMPP berperan sebagai server web pada komputer anda. XAMPP juga dapat disebut sebuah *CPanel server virtual*, yang dapat membantu anda melakukan *preview* sehingga dapat memodifikasi *website* tanpa harus *online* atau terakses dengan *internet*.

2.5. Web Service dan JSON (Javascript Object Notation)

Web service merupakan suatu komponen software yang merupakan self-containing, aplikasi modular self-describing yang dapat dipublikasikan, dialokasikan dan dilaksanakan pada web (Wahli dkk., 2006). Web Service adalah teknologi yang mengubah kemampuan internet dengan menambahkan kemampuan transactional web, yaitu kemampuan web untuk saling berkomunikasi dengan pola program-to-program (P2P) (Gottschalk, 2002). 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.

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python, PHP, dan lain-lain.

JSON menggunakan bentuk objek dan larik. Objek adalah sepasang nama/nilai yang tidak terurutkan, dimulai dengan “{” dan diakhiri dengan “}”, setiap pasangan nama/ nilai dipisahkan oleh koma. Larik adalah kumpulan nilai yang terurutkan. Larik dimulai dengan “[” dan diakhiri dengan “]”, setiap nilai dipisahkan oleh koma. Contoh JSON:

```
{“kain”:[{“kodeBarang”:"8321",“warna”:"Hitam"},
]}
```

2.6. Android

Android menurut (Nazarudin Safaat Harahap, 2012) merupakan sistem operasi untuk telepon seluler yang berbasis *Linux*. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam piranti bergerak. *Android* umum digunakan di *smartphone* dan juga tablet PC. Fungsinya sama seperti sistem operasi Symbian di Nokia, iOS di Apple, dan BlackBerry OS.

2.6.1 Sejarah Android

Awalnya perusahaan *search engine* terbesar saat ini, yaitu Google Inc, membeli Android Inc, pendatang baru yang membuat perangkat lunak untuk ponsel. Android Inc. Didirikan oleh Andy Rubin, Rich Milner, Nick Sears dan Chris White pada tahun 2003. Pada Agustus 2005 Google membeli Android Inc.

Dimulai pada tahun 2005, Android Inc. Dibawah naungan Google Inc. Berusaha membuat sebuah operating system mobile baru. Sejak itulah mulai beredar rumor bahwa Google akan melakukan ekspansi bisnis ke industri seluler. Akhirnya pada bulan September 2007 Google mengajukan hak paten atas produknya yang dinamai *Nexus One*.

Kemudian pada akhir tahun 2008, dibentuk sebuah tim kerja sama yang dinamai *Open Handset Alliance* (OHA). OHA ini terdiri dari beberapa produsen perangkat telekomunikasi ternama dunia, antara lain ASUS, Toshiba, Sony Erickson (sekarang Sony), Garmin, Vodafone, dan Softbank. OHA bekerja sama untuk mengembangkan sebuah kernel *Linux* yang akan dijadikan sebuah program untuk perangkat seluler. Hingga akhirnya OHA berhasil dan mengumumkan produk operating system mobile yang diberi nama android. Ponsel yang mendapat kehormatan yang mencoba pertama kali sistem operasi *Android* adalah *HTC Dream*. *HTC Dream* dirilis pada bulan Oktober tahun 2008. Sejak saat itu banyak perusahaan perangkat seluler di dunia ikut menggunakan *Android* sebagai *operating system* ponsel mereka.

2.6.2 Komponen Software *Android*

Android SDK (Software Development Kit) merupakan sebuah *tools* yang diperlukan untuk mengembangkan aplikasi berbasis *Android* menggunakan bahasa pemrograman Java. Pada saat ini *Android SDK* telah menjadi alat bantu dan *API (Application Programming Interface)* untuk mengembangkan aplikasi berbasis *android*. *Android SDK* dapat anda lihat dan unduh pada situs resminya, yaitu <http://developer.android.com/>, *Android SDK* bersifat gratis dan bebas anda distribusikan karena *Android* bersifat *open source*.

Berikut adalah garis besar pada arsitektur yang *Android* miliki:

1. *Applications dan widget*
2. *Application Frameworks*
3. *Libraries*
4. *Android Run Time*
5. *Linux Kernel*

2.6.3 Jenis dan Versi *Android*

a. *Android* versi 1.0.1.1

Sistem *android* versi 1.0 ini pertama kali hadir di tahun 2008, tepatnya pada Oktober 2008. Ponsel pertama yang menggunakan sistem *Android* adalah HTC. Pada bulan februari 2009 rilis update pertama *android* versi 1.1 yang masih belum memiliki nama. Dan untuk *android* versi selanjutnya google memutuskan untuk memberi nama versi *android* dengan nama makanan ringan, tujuannya adalah agar mudah diingat oleh para pengguna dan pecinta *android*.

b. Cupcake (*Android* versi 1.2 – 1.5)

Nama Cupcake diambil dari sebuah makanan ringan berupa kue kecil yang dikemas dalam sebuah wadah yang biasanya di sajikan dengan *frosting* diatasnya. Dengan penamaan Cupcake ini maka penamaan dari versi *Android* dimulai.

c. Donut (Android versi 1.6)

Versi selanjutnya adalah Donut, versi ini dirilis pada bulan September 2009 tentu donut sudah tak asing lagi bagi telinga orang Indonesia. Donut adalah sebuah makanan ringan berbentuk bulat yang terdapat lubang di tengahnya (berbentuk seperti cincin). Versi *android* ini memiliki fitur foto dan video dan kamera antarmuka dan integrasi pencarian yang lebih baik. Selain itu ditambahkan juga dukungan untuk ukuran layar yang lebih besar, dan diberi versi awal fitur navigasi *turn-by-turn* besutan Google.

d. Eclair (Android versi 2.0-2.1)

Eclair adalah makanan penutup yakni kue yang biasanya berbentuk persegi panjang yang dibuat dengan krim di tengah dan lapisan cokelat di atasnya. Di rilis pada bulan Oktober 2009, dengan rilisnya android versi ini google menambahkan fitur Bluetooth 2.1, Flash dan kamera dengan digital zoom, multi-touch, libe walpaper dan lainnya.

e. Froyo (Android versi 2.2-2.2.3)

Nama Froyo diambil dari kependekan dari Frozen Yoghurt yaitu yoghurt yang telah mengalami proses pendinginan sehingga terlihat seperti es krim. Versi ini dirilis pada bulan Mei 2010 dengan update memperbaiki segi kecepatan dan pengadopsian JavaScript dari browser Google Chrome dengan fitur-fiturnya.

f. Gingerbread (Android versi 2.3-2.4)

Nama Gingerbread diambil dari jahe atau cookie dengan rasa khas jahe yang biasanya berbentuk boneka mirip manusia. Versi ini dirilis pada bulan Desember 2010. Smartphone pertama yang memakai versi android ini adalah Nexus S yang dikeluarkan oleh produsen Samsung Android 2.3 Gingerbread ini merupakan OS Android yang paling lama berkuasa bahkan sampai sekarangpun beberapa vendor masih mengeluarkan ponsel dengan versi ini. Adapun perbaikan di versi Gingerbread ini adalah tambahan fitur dukungan untuk SIP internet calling, kemampuan nirkabel NFC, dukungan untuk dual kamera, dukungan untuk sensor giroskop dan sensor lainnya, fitur download manager, sejumlah tweak untuk penggunaan di Tablet, dan lainnya.

g. Honeycomb (Android versi 3.0-3.2)

Nama Honeycomb diambil dari nama sereal manis yang terbuat dari jagung rasa madu yang berbentuk sarang lebah, makanan ini yang populer sejak tahun 1965. Untuk versi ini merupakan versi yang ditujukan untuk gadget Tablet. *Android* Honeycomb rilis pada Februari 2011, kemudian upgrade ke versi 3.1 dan 3.2.

h. Ice Cream Sandwich (Android versi 4.0)

Android Ice Cream Sandwich atau biasa dikenal dengan nama *Android* 4.0 ICS adalah versi terbaru *Android* yang sangat mendukung baik untuk *smartphone*, *tablet*, dan lainnya. *Android* ini rilis pada 19 Oktober 2011. Yang baru dalam *android* ini adalah perubahan interface dari *android* sebelumnya, antara lain pengoptimalan multitasking, variasi layar beranda yang bisa disesuaikan dan interaktivitas mendalam serta cara baru yang ampuh untuk berkomunikasi dan berbagi konten. Nama Ice Cream Sandwich diambil dari nama makanan dimana lapisan es krim yang biasanya berupa vanila yang terjepit antara dua cookies coklat dan biasanya berbentuk persegi panjang.

i. Jelly Bean (Android versi 4.1-4.3)

Android 4.1 Jelly Bean diumumkan pada 27 Juni 2012 pada konferensi Google I/O yang secara resmi dikenalkan ke publik sekitar Oktober 2012. Versi ini adalah yang tercepat dan terhalus dari semua versi Android. Fitur baru yang terdapat di versi ini adalah meningkatkan kemudahan dan keindahan tampilan dari Ice Cream Sandwich dan memperkenalkan pengalaman pencarian Google yang baru di Android. Android 4.2 Jelly Bean juga menawarkan peningkatan kecepatan dan kemudahan Android 4.1 serta mencakup semua fitur baru seperti *Photo Sphere* dan desain baru aplikasi kamera, *keyboard Gesture Typing*, *Google Now* dan lainnya. Untuk nama Jelly Bean pasti semua tahu kan, ya nama ini diambil dari sejenis permen yang juga populer disebut kacang jeli.

j. KitKat (Android versi 4.4)

Awalnya android versi ini diisukan bernama Key Lime Pie. Namun pada tanggal Oktober 2013 google merilis kitkat sebagai generasi android berikutnya. Android versi ini memiliki banyak fitur dan semakin menjakan para pengguna android. Diantaranya : *Immersive mode*. Akses kontak langsung dari aplikasi telepon, *google now launcher*, dan pastinya memiliki Interface UI yang baru

k. Lollipop (Android versi 5.0)

Android versi 5.0 merupakan versi paling baru dari sistem operasi android. Android 5.0 sendiri dianggap membawa update yang fantastis, banyak perubahan yang disertakan Google di dalamnya.

2.6.4 Kelebihan Android

Kelebihan dari sistem operasi android adalah

2.6.4.1 User friendly

Kalimat ini sangat melekat pada sistem operasi *windows* miliknya *microsoft*, ibaratnya kita dapat sangat mudah mengoperasikan komputer hanya dengan belajar beberapa hari bahkan beberapa jam saja, dan ini juga melekat ada *android* yang berjalan pada *smartphone*.

2.6.4.2 Notifications

Dapat dengan mudah mendapat notifikasi dari *smartphone android* dengan mengatur beberapa akun *email*, *sms*, *voice dial*, *update*, dan lain sebagainya

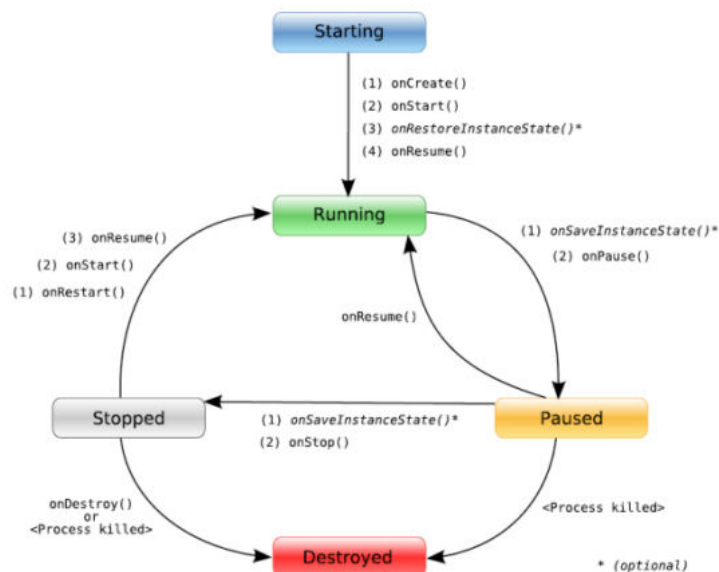
2.6.4.3 Open Source

Operating system ini memang dibuat *open source* oleh penciptanya, karena memang berbasis *kernel linux*.

2.7 Konsep Android

(Novtiananda , M. Imam, M. Reza,. & Febriani, 2014) mendefinisikan “*android* merupakan sebuah sistem operasi telepon seluler dan komputer tablet layar sentuh (*touchscreen*) yang berbasis *linux*”. Sedangkan menurut Murya di dalam Novtiananda et al. mendefinisikan “*android* adalah sistem operasi berbasis

linux yang digunakan untuk telepon seluler (*mobile*) seperti telepon pintar (*smartphone*) dan komputer tablet (PDA)” (p.2). Perangkat berbasis *android* hanya mempunyai satu layar *foreground*. Normalnya saat menghidupkan *android*, yang pertama di lihat adalah *home*. Kemudian bila menjalankan sebuah aplikasi catur, *User Interfacenya* akan menumpuk diatas layar sebelumnya (*home*). Kemudian bila melihat *help*-nya catur, maka UI *help* akan menimpa UI sebelumnya (catur), begitu seterusnya. Semua proses diatas direkam di *application stack* oleh sistem *Activity Manager*. Menekan tombol *back* hanya kembali ke halaman sebelumnya, analoginya mirip dengan *browser* dimana ketika mengklik tombol *back browser* akan kembali menampilkan halaman sebelumnya. Setiap *user interface* diwakili oleh kelas *Activity* (*Activity Class*). Setiap *Activity* mempunyai siklus, dapat dilihat di gambar 2.2. Setiap aplikasi dapat terdiri dari satu atau lebih *Activity* yang diproses dalam *Linux*.



Gambar 2.1 Siklus Activity

Selama siklus ini berjalan, *Activity* bisa mempunyai lebih dari dua status seperti yang terlihat pada gambar 2.2. Kita tidak bisa mengontrol setiap status karena semuanya sudah ditangani oleh sistem. Namun kita akan mendapat pesan saat terjadi perubahan status melalui method onXX(). Berikut ini penjelasan setiap status.

Table 2.1 Status Android

NO	NAMA STATUS	KETERANGAN
1	onCreate(Bundle)	Dipanggil saat pertama kali aplikasi dijalankan. Kita dapat menggunakan ini untuk deklarasi variable atau membuat <i>userinterface</i> .
2	onStart()	Mengindikasikan <i>Activity</i> yang ditampilkan ke pengguna (<i>user</i>).
3	onResume()	Dipanggil saat aplikasi kita mulai berinteraksi dengan pengguna. Disini sangat cocok untuk meletakkan animasi ataupun musik
4	onPause	Dipanggil saat aplikasi yang kita jalankan kembali ke halaman sebelumnya atau biasanya karena ada <i>Activity</i> baru yang dijalankan. Disini cocok untuk meletakkan algoritma penyimpanan (<i>save</i>).
6	onStop	Dipanggil saat aplikasi kita berjalan di belakang layar dalam waktu cukup lama
7	onRestart()	<i>Activity</i> kembali menampilkan <i>user interface</i> setelah status <i>stop</i>
8	onDestroy()	Dipanggil saat aplikasi benar-benar berhenti
9	onSaveInstanceState(Bundle)	Method ini mengizinkan <i>Activity</i> untuk menyimpan setiap status instance. Misal dalam mengedit teks, kursor bergerak dari kiri ke kanan

2.7.1 Komponen Aplikasi *Android*

Berikut ini adalah beberapa komponen *android* yang harus diketahui

1. *Activity*

Normalnya setiap *Activity* menampilkan satu buah user interface kepada pengguna. Misalnya sebuah *Activity* menampilkan daftar menu minuman, kemudian dapat memilih satu jenis minuman. Contoh lainnya pada aplikasi sms, dimana satu *Activity* digunakan untuk menulis pesan, *Activity* berikutnya untuk menampilkan nomor kontak tujuan, atau *Activity* lainnya digunakan untuk menampilkan pesan lama. Meskipun *activity-Activity* tersebut terdapat dalam satu aplikasi sms, namun masing-masing *Activity* berdiri sendiri. Untuk pindah dari satu *Activity* ke *Activity* lainnya dapat melakukan satu *event* misalnya tombol diklik.

2. *Service*

Service tidak memiliki *user interface*, namun berjalan di belakang layar. Misalnya *music player*, sebuah *Activity* digunakan untuk memilih lagu kemudian di mainkan. Agar *music player* bisa berjalan dibelakang aplikasi lain maka harus menggunakan *service*.

3. *Intens*

Intens adalah mekanisme untuk menggambarkan sebuah action secara detail seperti bagaimana cara mengambil sebuah photo.

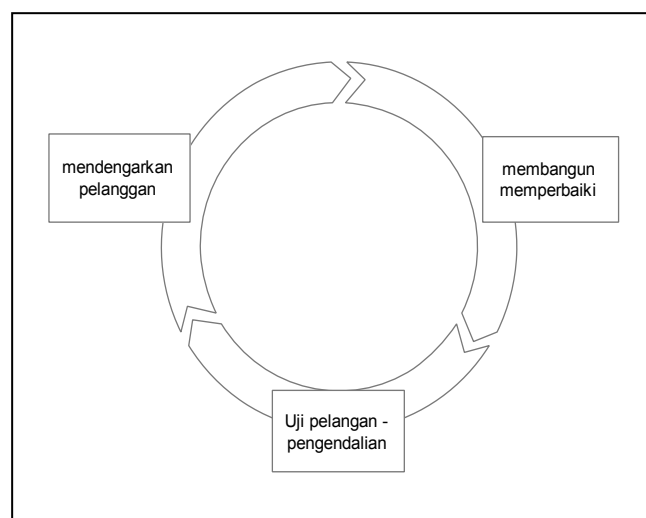
4. *Content Providers*

Menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *Activity*, misalnya kita menggunakan aplikasi berbasis peta (MAP). *Activity* membutuhkan cara untuk mengakses data kontak untuk prosedur navigasi. Disinilah peran *content providers*.

2.8. Metode Pengembangan Perangkat Lunak Dengan Metode Prototype

(Pressman, 2002) menguraikan bahwa *prototype* merupakan metode yang efektif dalam merancang perangkat lunak. *Prototype* dimulai dengan mengumpulkan kebutuhan. Pengembang dan pelanggan bertemu dan mendefinisikan objek keseluruhan dari perangkat lunak, mengidentifikasi segala

kebutuhan yang diketahui dan kemudian melakukan perancangan kilat. Perancangan kilat berfokus pada penyajian dari aspek-aspek perangkat lunak tersebut yang akan nampak bagi pelanggan atau pemakai (contohnya pendekatan input dan format output). Perancangan kilat membawa kepada konstruksi sebuah *prototype*. *Prototype* tersebut dievaluasi oleh pelanggan dan dipakai untuk menyaring kebutuhan pengembangan perangkat lunak. Metode *prototype* dapat dilihat pada gambar 2.2 berikut:



Gambar 2.2 *Prototipe* paradigma

Prototype juga dapat didefinisikan sebagai proses pengembangan suatu prototipe secara cepat untuk digunakan terlebih dahulu dan ditingkatkan terus menerus sampai didapatkan sistem yang utuh. Merupakan proses yang digunakan untuk membantu pengembang perangkat lunak dalam membentuk *prototype* dari perangkat lunak yang harus dibuat. Proses pada model *prototyping* dapat dijelaskan sebagai berikut :

- 1) Pengumpulan kebutuhan : *Developer* dan klien bertemu dan menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya.
- 2) Perancangan : Perancangan dilakukan cepat dan rancangan mewakili semua aspek perangkat lunak yang diketahui, dan rancangan ini menjadi dasar pembuatan *prototype*.

3) Evaluasi *prototype* : Klien mengevaluasi *prototype* yang dibuat dan digunakan untuk memperjelas kebutuhan perangkat lunak.

Perulangan ketiga proses ini terus berlangsung hingga semua kebutuhan terpenuhi. *Prototype-prototype* dibuat untuk memuaskan kebutuhan klien dan untuk membangun perangkat lunak lebih cepat, namun tidak semua *prototype* bisa dimanfaatkan. Demi kebutuhan klien lebih baik *prototype* yang dibuat diusahakan dapat dimanfaatkan.

2.9 Unified Modeling Language (UML)

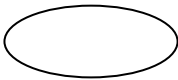
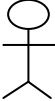

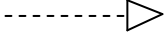
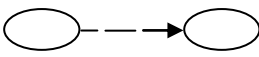
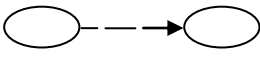
(A.S Rosa dan Salahuddin, 2011) menjelaskan bahwa *unified modeling language* (UML) menguraikan salah satu *standar* bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat aplikasi dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek.

Kesimpulan metode *UML (Unified Modeling Language)* adalah merupakan sebuah metode atau sebuah bahasa yang digunakan dalam menterjemahkan, menjelaskan, memodelkan, mendefinisikan suatu sistem dengan bentuk simbol-simbol tertentu yang bertujuan untuk memberikan penjelasan-penjelasan detail dari sebuah sistem.

2.9.1 Use Case Diagram

(A.S Rosa dan Salahuddin, 2011) menguraikan bahwa *use case* diagram merupakan pemodelan untuk kelakuan sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Simbol-simbol yang digunakan untuk pembuatan *use case* diagram dapat dilihat pada tabel 2.2 berikut:

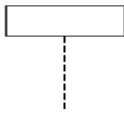
Tabel 2.2 Simbol *Use Case Diagram*

Simbol	Keterangan
<i>Use Case</i> 	Menggambarkan bagaimana seseorang akan menggunakan atau memanfaatkan sistem.
Aktor 	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri.
Asosiasi 	Komunikasi antara <i>use case</i> dan aktor yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
Generalisasi 	Sebagai penghubung antara aktor- <i>use case</i> atau <i>use case-use case</i> .
<<Include>> 	<i>Include Relationship</i> (relasi cakupan) : Memungkinkan suatu <i>use case</i> untuk menggunakan fungsionalitas yang disediakan oleh <i>use case</i> yang lainnya.
<<Extend>> 	<i>Extend Relationship</i> : Memungkinkan relasi <i>use case</i> memiliki kemungkinan untuk memperluas fungsionalitas yang disediakan oleh <i>use case</i> yang lainnya.


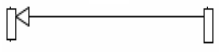
2.9.2 Sequence Diagram

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa pesan yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Simbol-simbol yang digunakan untuk pembuatan *class diagram* dapat dilihat pada tabel 2.4 berikut :

Tabel 2.3 Simbol *Sequence Diagram*

GAMBAR	NAMA	KETERANGAN
	<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.



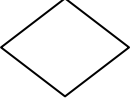


Tabel 2.3 (Lanjutan)

	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas.

2.9.3 Activity Diagram

Activity Diagram menggambarkan aliran kerja atau aktivitas dari sebuah sistem atau proses bisnis bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Simbol-simbol yang digunakan untuk pembuatan *activity diagram* dapat dilihat pada tabel 2.5 di bawah ini :

Tabel 2.4 Simbol Activity Diagram

Simbol	Keterangan
Status Awal 	Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan 	Asosiasi percabangan dimana ada pilihan aktivitas lebih dari satu.
Penggabungan 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status Akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

Tabel 2.4 (Lanjutan)

<p><i>Swimlane</i></p> <div style="border: 1px solid black; width: 100px; height: 20px; margin: 5px auto; text-align: center;">nama swimlane</div>	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas.
--	---

2.10 Penelitian Terdahulu

Table 2.6 berikut merupakan jurnal penelitian terdahulu :

Tabel 2.5 Penelitian Terdahulu

Nama	Judul	Terbit/ Tahun	Abstrack
Vinno Christmantara	SISTEM PENJUALAN TAKE ORDER BERBASIS ANDROID DENGAN LAYANAN BERBASIS LOKASI DAN METODE HAVERSINE (STUDI KASUS PADA DATA TOKO/ TAYLOR PEMBELI PRODUK TEKSTIL PT. MAXISTAR INTERMODA INDONESIA UNTUK AREA MALANG)	Jurnal Teknik Informatika/ 2015	Sistem penjualan yang digunakan oleh PT. Maxistar Intermoda Indonesia adalah <i>take order</i> , dan pemesanan dilakukan secara manual. <i>Salesman</i> tidak dapat mengetahui jumlah stok sebenarnya yang ada di gudang, sementara bagian gudang tidak menerima informasi <i>order</i> secara cepat. Penelitian ini bertujuan untuk membangun sistem penjualan yang menggunakan <i>mobile device</i> berbasis Android yang mempermudah proses pengecekan stok dan pengiriman order, dilengkapi dengan layanan berbasis lokasi yang akan mempermudah <i>salesman</i> untuk menemukan toko di sekitarnya (<i>Point of Interest</i>), dan memilih toko melalui fitur <i>check-in</i> . Metode yang digunakan untuk fitur terakhir adalah Metode Haversine.
Afriana Fita	ANALISIS DAN PERANCANGAN SISTEM MOBILE DELIVERY ORDER BERBASIS ANDROID DI SHANIA CAKE SHOP YOGYAKARTA	SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER AMIKOM YOGYA	Shania Cake Shop merupakan sebuah toko online yang bergerak dalam bidang penjualan dan pembuatan kue sesuai permintaan dari pelanggan. Pelanggan bisa melihat contoh daftar kue di blog shania cake, lalu pelanggan dapat memesan melalui Facebook, email maupun SMS. Lalu pihak shania baru akan menghubungi pelanggan tentang kisaran harganya.

Tabel 2.5 (lanjutan)

		KARTA 2012	<p>Hasil analisis masalah ditemukan beberapa hal yang masih diabaikan seperti,</p> <p>tidak adanya daftar harga, data pelanggan tidak dapat terkumpul menjadi satu karena cara pemesanannya ada banyak sehingga dapat menimbulkan perulangan jika tidak teliti dan jika pelanggan ingin melihat contoh kue harus membuka beratus-ratus halaman postingan di blog dan dapat menimbulkan ketidakefisiensian waktu.</p> <p>Oleh karena itu diharapkan media pemesanan dalam bentuk mobile aplikasi dapat menjadi solusi untuk mengatasi kekurangan kekurangan dan mampu untuk menengahi baik proses pemesanan maupun pengiriman produk dari Shania Cake yang dapat menguntungkan baik pihak toko maupun pihak pelanggan.</p>
--	--	-----------------------	---