BAB II. LANDASAN TEORI

2.1. Tinjauan Pustaka

Penelitian perancangan aplikasi android untuk sales dengan menggabungkan local based service yang berbasis client-server dengan tujuan mempermudah pekerjaan sales dalam absensi adalah dengan cara aplikasi andoid mengambil titik koordinat yang berupa longitude dan latitude disimpan dalam database untuk rekapan data sales tersebut. Aplikasi android untuk sales dengan menggabungkan local based service berbasis client-server dapat dipergunakan untuk sales ketika mendapatkan proyek dan langsung melakukan input saat itu juga (real time) dengan cara sales input dengan aplikasi android kemudian setelah data di save, data tersebut akan masuk ke dalam server, dan saat itu juga admin dapat melakukan tindakan terhadap data inputan dari sales untuk cetak atau revisi. Hasilnya akan dijadikan laporan untuk manager

Beberapa penelitian yang telah dilakukan terkait Mobile Location Base Services dapat disimpulkan melalui Tabel 2.1 berikut ini :

Tabel 2.1 – Tinjauan pustaka penelitian *Mobile Location Base Services*

Nama Peneliti dan Tahun	Judul	Hasil	Kelebihan dan Kekurangan
(Tullah, Tobing, & Hadi, 2015)	*Sistem Aplikasi Android untuk Sales Dengan Local Based Service (LBS) Berbasis Client - Server*	Aplikasi android yang dapat mencatat absen salesman,mencari dan mendapatkan proyek , input data dan melihat data proyek melalui website	-Aplikasi ini mampu memonitoring absensi sales dan proyek yang didapat serta dapat mencetak laporan kepada manager. -Database nya masih menggunakan database untuk sekala kecil.

(Christmantara, 2015)	*Sistem Penjualan Take Order Berbasis Android Dengan Layanan Berbasis Lokasi Dan Metode Haversine*	aplikasi ini dapat membuat/ efisiensi dan efektifitas proses cek stok dan order barang serta menentukan posisi salesman	-Dapat mengetahui keberadaan salesman melalui pelacakan gps -Untuk daerah yang tidak tercover internet aplikasi tidak dapat berjalan
(Setianna, 2012)	*Perancangan dan Implementasi Aplikasi Pelaporan Penjualan dan Pengecekan Stok berbasis Android*	Pelaporan penjualan dapat di lakukan sebelum penjualan produk berakhir dimana data transaksi dapat langsung di proses oleh admin melalui aplikasi yang ada di smartphone	Dapat melakukan pengecekan stock barang dan laporan penjualan otomatis tanpa menunggu selesai transaksi
(Putra, 2012)	*Aplikasi Ponsel Berbasis Android untuk Penjualan*	Aplikasi ini mampu mencatat pembelian dan penjualan secara cepat dan tepat	aplikasi banyak memakan memori penyimpanan ponsel,belum bisa melakukan pencetakan nota penjualan maupun laporan pembelian dan penjualan, belum bisa melakukan backup data sendiri dan belum bisa melakukan konversi data menjadi file
(Sheng, 2011)	*An Analysis on Mobile Location based Services*	Navigasi GPS yang paling banyak di gunakan dan aplikasi yang berhubungan dengan menghibur seperti game dan komunitas dengan dunia luar	Menurut lin Mobile lokasi berbasis layanan di kategorikan layanan hiburan, informasi dan perdagangan.
(Lin, 2011)	*Mobile Location-based Servi* (Wang, 2008)*ces: An Empirical Study of User*	kategori MBLS adalah sebagai berikut: navigasi (0,21434), hiburan (0,20744), keamanan dan	Ada dua aplikasi yang yang terkenal yaitu apple store dan android market. Navigasi dan kategori yang berhubungan

	Preferences	pelacakan (0,20164), informasi (0,19891), dan perdagangan (0,1777).	dengan perjalanan adalah paling populer di apple stores sedangkan di android market perjalanan dan kategori komunitas yang paling populer
(Wang, 2008)	*Location-based Services: A Roadmap for New Zealand*	lokasi panggilan darurat memainkan peran utama sebagai penggerak teknologi lokasi dan pembangunan aplikasi komersial layanan berbasis lokasi	untuk navigasi dan routing layanan dari sektor seperti pariwisata, permintaan keselamatan publik untuk panggilan lokasi darurat. Ada kekurangan yang signifikan dari LBS kurang cukup akurasi dengan teknologi posisi
(Carey, 2013)	*Smartphone location based services in the social, mobile, and surveillance practices of everyday life*	Control pengguna kehilangan kontrol atas privasi mereka dari kurangnya memahami bagaimana bekerja teknologi yang semakin kompleks, navigasi, hubungan sosial, representasi diri, dan privasi.	LBSS terutama digunakan untuk navigasi karena mereka mudah digunakan, nyaman, berguna dan menghemat waktu. menyebabkan masalah ketergantungan dan gangguan.
(Domingues, 2008)	*Improving Sales Force Performance Through Mobile Applications*	meningkatkan kinerja Sales Force melalui aplikasi mobile	proyek ini memungkinkan untuk mengidentifikasi semua konsep Sales Force Automation yang utama, fitur dan solusi yang dapat di lakukan
(Malaka, 2000)	*Developing Location Based Services for Tourism - The Service Providers View*	Layanan gateway mengintegrasikan aplikasi lokasi di atas layanan terminal mobile, platform nirkabel, Internet Protocol (IP)	Dapat menampilkan rute yang terbaik atau rute terpendek antara dua titik tertentu di jalan jaringan. Agen rute menyediakan fungsionalitas routing

		sistem platform, dan atau penentuan posisi mobile. layanan ini beroperasi antara sistem nirkabel IP dan server aplikasi lokasi.	terkenal. mencoba untuk menyarankan Melihat dari wisata individu sesuai dengan minat wisatawan
(Global Navigation Satelite system agency, 2015)	*Location-Based Services (LBS)*	pasar LBS terus tumbuh dengan perangkat high end memanfaatkan multi-konstelasi dan posisi hybrid.	aplikasi yang memanfaatkan informasi lokasi hampir setengah dari total kemudian kategori permainan dan hiburan
(Smith, 2000)	*LOCATION BASED SERVICES – THE UNDERLYING TECHNOLOGY*	interkoneksi antara komunikasi nirkabel, positioning teknologi dan domain SDI sebagai daerah kontribusi yang diperlukan untuk pengembangan LBS	LBS memberi manfaat keselamatan pribadi seperti layanan tambahan yang disediakan oleh E-911, LBS menawarkan berbagai aplikasi untuk pengguna mencari tempat parkir atau kemampuan untuk melacak bingkisan. Layanan sehari-hari ini akan berkontribusi kepada perekembangan teknologi masyarakat.
(Dilmener, 2013)	*LOCATION BASED SALES PROMOTION STRATEGIES*	promosi yang memungkinkan pelanggan untuk mewujudkan penghematan besar jika mereka check in ke toko yang berpartisipasi menggunakan layanan berbasis lokasi	Menggunakan geolocation untuk membuat tampilan virtual toko sehingga pengguna hanya dapat membeli produk edisi terbatas ketika mereka berdiri di depan sebuah iklan adalah pengalaman

(Tullah, Tobing, & Hadi, 2015) *Location Based Services* (LBS) merupakan suatu layanan yang beraksi aktif terhadap perubahan entitas posisi sehingga mampu mendeteksi letak objek dan memberikan layanan sesuai dengan letak objek yang telah di ketahui tersebut. Pada teknologi LBS berbasis jaringan seluler, penentuan posisi sebuah tempat atau lokasi di tentukan berdasarkan posisi yang relatif. Dalam menentukan posisi dan layanan IP nirkabel yang menggunakan informasi geografis untuk memberikan layanan informasi lokasi kepada pengguna. Aplikasi android untuk sales dengan menggabungkan *local based service* berbasis *client-server* dapat dipergunakan untuk sales ketika mendapatkan proyek dan langsung melakukan input saat itu juga (*real time*) dengan cara sales input dengan aplikasi android kemudian setelah data di *save*, data tersebut akan masuk ke dalam *server*, dan saat itu juga admin dapat melakukan tindakan terhadap data inputan dari sales untuk cetak atau revisi. Hasilnya akan dijadikan laporan untuk manager.

(Christmantara, 2015) Layanan berbasis lokasi adalah layanan informasi yang mengutilisasi kemampuan untuk menggunakan informasi lokasi dari perangkat bergerak dan dapat diakses dengan perangkat bergerak melalui jaringan telekomunikasi bergerak Fokus objek permasalahan pada penelitian ini adalah efisiensi dan efektifitas proses cek stok dan *order* barang, menentukan posisi *salesman* sehingga dapat mengetahui toko-toko serta *taylor* di sekitarnya menggunakan *mobile device* Android. Untuk dapat membangun sistem dan aplikasi yang diinginkan, maka data yang dibutuhkan adalah data barang, data stok di gudang, dan data toko. Sistem dapat mengefisienkan proses penjualan

dengan menggantikan proses order yang selama ini dilakukan secara manual, yaitu dengan cara menelpon ke gudang di Surabaya (*by phone*).

(Setianna, 2012) Kondisi salesman yang tersebar dari beberapa perusahaan masih melakukan penawaran yang bersifat manual dengan pembukuan yang sederhana bahkan ada yang tidak terkomputerisasi. Hal ini biasanya di sebababkan kurangnya pengetahuan tentang teknologi baru yang muncul atau teknologi yang sudah ada belum di gunakan secara maksimal. Alat komunikasi yang di gunakan oleh salesman masih menggunakan metode telepon secara langsung untuk menanyakan stok barang yang masih ada. Pelaporan hasil penjualan juga di lakukan setelah Jam kerja selesai. Hal tersebut membuat perusahaan tersebut harus melakukan pendataan akhir pekerjaan atau di bahkan di lakukan esok hari nya sehingga tidak efisien mengingat akses data dan informasi sangat penting untuk menentukan kebijakan di dalam bisnis. Dengan menggunakan aplikasi berbasis android pelaporan penjualan dan pengecekan stok cukup cepat dan di akses secara mudah. Pelaporan penjualan dapat di lakukan secara langsung sehingga monitor penjualan dapat di lakukan tanpa menunggu akhir proses pemasaran.

(Putra, 2012) Dalam kegiatan transaksinya, pemilik dari Q-Mono Flower masih menggunakan pulpen dan buku catatan khusus untuk mencatat tiap transaksi yang terjadi, sehingga bisa diketahui total pendapatan tiap hari dan tiap bulannya. Kemudahan yang dihasilkan oleh ponsel berbasis Android memungkinkan dibuatnya sebuah aplikasi untuk penjualan pada kios eceran yang berguna untuk mempermudah penjual dalam melakukan transaksi dengan konsumen serta dalam mengetahui pendapatan perbulannya. Dengan penggunaan

aplikasi ponsel berbasis Android untuk penjualan pada kios eceran ini mampu menghemat biaya untuk membeli buku catatan, nota, pulpen, pensil, *tipp-ex*, penghapus. Selain itu tidak perlu memakai komputer desktop yang digunakan untuk transaksi penjualan seperti yang ada pada toko-toko grosir, minimarket, atau supermatket karena tidak cocok untuk kios eceran dan membutuhkan biaya yang mahal serta memakan tempat. Hasil informasi yang bisa diperoleh dari aplikasi tersebut adalah jumlah sisa stok barang, jumlah pembelian tiap hari, tiap bulan, atau rentang waktu yang ditentukan oleh user, jumlah penjualan tiap hari, tiap bulan, atau rentang waktu yang ditentukan oleh user, dan total tagihan pembayaran yang harus dibayar oleh pelanggan tiap kali transaksi.

(Sheng, 2011) Ponsel memiliki karakteristik di mana-mana, tepat waktu, pelacakan lokasi, interaksi dan personalisasi. Kemampuan mobile data operator seluler semakin baik untuk pengembangan nilai data baru untuk memenuhi permintaan pelanggan tentang mobile lokasi berbasis layanan yang paling populer pada perangkat mobile. Dengan metode Analytic Hierarchy Process (AHP) untuk menemukan pengguna mobile lokasi berbasis layanan.

(Lin, 2011) meneliti untuk mengetahui pengguna Mobile Location Base Services (Mlbs). ponsel lokasi berbasis layanan yang ditinjau dan diklasifikasikan lebih lanjut menjadi sebuah hirarki. hirarki diaplikasikan untuk menyelidiki proses analisis hirarki preferensi pengguna sehingga dapat di kembangkan untuk memperbaiki dan dapat memperkirakan akan meningkat dan hambatan pengembangan lebih sedikit.

(Wang, 2008) Kemajuan di mobile, nirkabel dan teknologi lokasi telah memungkinkan aplikasi layanan rute dan bantuan panggilan darurat. Pada tahun

1966, undang-undang baru diperkenalkan bertujuan meningkatkan layanan panggilan darurat yang dikenal sebagai 911, Operator seluler diwajibkan untuk menyediakan informasi secara otomatis lokasi pemanggil situasi darurat dalam panggilan 911.

(Carey, 2013) ada Empat focus utama penelitian location based service (LBSS) yaitu pemetaan, manajemen informasi lokasi, jaringan sosial, dan geotagging melalui media sosial. Penelitian menekankan dampak teknologi pada bagaimana pengguna menampilkan diri kepada orang lain, menavigasi ruang fisik, dan memediasi privasi mereka.

(Domingues, 2008) Mobile Sales Force Automation aplikasi (MSFA) merupakan bagian dari informasi yang lebih luas Sistem yang dimiliki oleh organisasi penjualan, sistem Sales Force Automation (SFA). Sistem ini termasuk desktop dan aplikasi mobile data dengan sinkronisasi data dengan jaringan perusahaan. MSFA ditargetkan untuk perwakilan penjualan di lapangan dan manajer penjualan karena aktivitas kerja mereka yang paling mungkin dapat meningkatkan efisiensi dan efektivitas melalui aplikasi mobile. fitur Mobile aplikasi Sales Force Automation menyediakan: manajemen konten, Agenda, saluran komunikasi, Services, Reporting and Settings.

(Malaka, 2000) fokus penelitian ini penciptaan User-friendly layanan mobile pribadi untuk Pariwisata menerapkan, memvalidasi, dan uji coba layanan yang berkaitan dengan pariwisata nilai tambah bagi pengguna nomadic di mobile dan jaringan tetap. Khususnya penggunaan teknologi agen akan dievaluasi dalam hal user-akseptabilitas, kinerja dan praktek terbaik sebagai pendekatan yang cocok untuk layanan diakses cepat. potensi layanan berbasis lokasi (LBS) untuk

pariwisata dengan menunjukkan beberapa contoh aplikasi. LBS adalah layanan untuk pengguna ponsel yang mengambi posisi saat pengguna ke account user ketika melakukan tugas mereka. menjelajah sistem informasi dapat menggunakan roaming di ruang fisik dengan teknologi lokalisasi untuk beradaptasi dengan model konteks yang lebih kaya dari pengguna sebelumnya dan evaluasi Posisi gerakan.

(Global Navigation Satelite system agency, 2015) aplikasi GNSS didukung oleh beberapa kategori perangkat, terutama smartphone dan tablet, juga peralatan tertentu seperti perangkat pelacakan, kamera digital, portable komputer dan peralatan kebugaran. Rata-rata, lebih dari 70 aplikasi per perangkat di-download oleh pengguna,meskipun 50% dari pengguna tidak pernah membayar lebih dari $ 1 untuk sebuah aplikasi. Download aplikasi yang mengandalkan data posisi akan mencapai 7,5 miliar pada 2019, naik dari 2,8 miliar pada tahun 2014. Perangkat Smartphone dan tablet, perangkat pelacakan, digital kamera, perangkat kebugaran, laptop, smartphone di seluruh dunia pasar saham 2014. Samsung 24,5%, appel 14,8%, Lenovo 5,4%, Huawei 5,7%, LG 4,6% , Lainnya 45%. Jumlah aplikasi di toko - Desember 2014 Google Play 1,43 juta aplikasi, Apple App Store 1,21 juta aplikasi, Windows Phone Store 300 ribu aplikasi, Amazon Appstore 293 ribu aplikasi dan Blackberry Dunia 130 ribu aplikasi. Berdasarkan GNSS total handset tahun 2014 3.1 miliar dan tahun 2017 naik 5,2 Miliar banyak pengguna memiliki lebih dari satu perangkat.

(Smith, 2000) kecepatan jaringan data nirkabel tinggi meningkatkan faktor bentuk ponsel dan teknologi lokasi akurasi tinggi. Telepon selular telah memiliki dampak yang dramatis pada masyarakat. Metode yang menggambarkan rute dan

area ruangan mempunyai fungsi sumber daya, hari ini sumber daya itu ponsel yang menyediakan orang dengan rincian informasi yang relevan yang paling di butuhkan. waktu kritis dan lokasi layanan informasi telah diidentifikasi. LBS memperkuat geografi dan rasa tempat untuk pelanggan mobile. Penyediaan informasi yang relevan dengan posisi saat pengguna perlahan-lahan melalui LBS memasuki konsumen pasar, dan dapat memberikan layanan seperti arah mengemudi dan menemukan toko terdekat. Ini adalah integrasi komputasi jaringan, teknologi positioning dan komunikasi nirkabel yang telah menyebabkan perkembangan LBS. diharapkan yang LBS akan mendapatkan keuntungan dari perkembangan di setiap daerah dan karenanya memberikan peningkatan layanan kepada konsumen.

(Dilmener, 2013) Bisnis menjadi lebih dekat dengan pelanggan dan dapat menyajikan proposal kepada pelanggan dengan teknologi baru. Lokasi berdasarkan promosi penjualan (LBSP) dapat didefinisikan sebagai proses mencapai pelanggan yang tepat, di lokasi yang tepat, pada waktu yang tepat dengan tawaran promosi penjualan disesuaikan atau massal dengan alat berbasis teknologi lokasi. LBM menargetkan pengguna ponsel dalam area geophone tertentu atau lokasi melalui alat pemasaran mobile. LBM menargetkan pengguna langsung dan sebagian besar waktu, konsumen menerima pesan pada perangkat mobile mereka yang berisi panggilan untuk bertindak seperti memasuki kompetisi, kunjungi situs web atau memesan produk dan insentif, seperti kupon LBM menjembatani kesenjangan antara semua bentuk media pemasaran, termasuk media sosial dan internet. LBM meliputi pemanfaatan atau integrasi semua media untuk terlibat dan pasar kepada orang-orang di tempat-tempat

tertentu dengan penawaran khusus. LBM menggunakan layanan berbasis lokasi untuk mencapai pelanggan berdasarkan di mana mereka berada. real-time diskon dikirimkan ke perangkat mobile juga akan mempercepat dengan layanan berbasis lokasi

2.2. Landasan Teori

2.2.1. *Sejarah dan Perkembangan Android*

Nama android adalah nama nick dari Andy Rubin yang di berikan rekan kerjanya karena kecintaan Andy Rubin terhadap robot, Pembuat logo android adalah Irina Blok, logo android di rancang untuk google pada tahun 2007 oleh desainer grafis Irina Blok. Inspirasi logo android didapatkan dari simbol manusia yang terdapat pada pintu toilet, dan memodifikasi bentuknya menjadi bentuk robot. Android,Inc didirikan di Palo Alto, California, pada bulan oktober 2003 oleh Andy Rubin, Rich Miner, Nick Sears, dan Chris White untuk mengembangkan perangkat telepon pintar. Tujuan awal pengembangan android adalah untuk mengembangkan sebuah sistem operasi canggih yang diperuntukan untuk kamera digital, namun kemudian disadari bahwa pasar untuk perangkat tersebut tidak cukup besar, dan pengembangan Android lalu dialihkan ke pasar telepon pintar untuk menyaingin Symbian dan Windows Phone. Meskipun para pengembang Android adalah pakar-pakar teknologi yang berpengalaman, Android Inc. Dioperasikan secara diam-diam, hanya diungkapkan bahwa para pengembang sedang menciptakan sebuah perangkat lunak yang diperuntukan bagi telepon seluler.

Google mengakuisisi Android Inc. pada tanggal 17 agustus 2005, menjadikannya sebagai anak perusahaan yang sepenuhnya dimiliki oleh Google. Pendiri Android Inc. masih tetap bekerja digoogle setelah diakuisisi. Setelah itu, tidak banyak yang diketahui tentang perkembangan Andorid Inc.,namun banyak yang beranggapan bahwa Google berencana untuk memasuki pasar telpon seluler dengan tindakan ini. Google dengan tim yang dipimpin oleh Rubin mulai mengembangkan platform perangkat seluler dengan mengunakan Kernel Linux. Google memasarkan platform tersebut kepada produsen perangkat seluler dan operator nirkabel, dengan janji bahwa mereka menyediakan sistem yang fleksibel dan bisa diperbaruhi. Google telah memilih beberapa mitra perusahaan perangkat lunak dan perangkat keras, serta mengisyaratkan kepada operator seluler bahwa kerjasama ini terbuka bagi siapapun yang ingin berpartisipasi.

Pada saat perilisan perdana Android, 5 Novemnber 2007, Android bersama Open Headset Allience menyatakan mendukung perkembangan open source pada perangkat seluler. Di lain pihak, Google melirik kode-kode Android dibawah linseni Apache, sebuah lisensi perangkat lunak dan open platform perangkat seluler. Sejak tahun 2008, Andoid secara bertahap telah melakukan sejumlah pembaruan untuk meningkatkan kinerja sistem operasi, menambahkan fitur, dan memperbaiki Bug yang terdapat pada versi sebelumnya. Setiap versi utama yang dirilis dinamakan secara alfabetis berdasarkan nama-nama makanan pencuci mulut atau cemilan bergula, misalnya , versi 1.5 bernama Cupcake , yang kemudian diikuti dengan versi 1.6 Donut. Versi terbaru adalah 4.3 Jelly Bean. Pada tahun 2010 , Google merilis seri Nexus, perangkat telpon pintar dan tablet

dengan sistem operasi Android yang diproduksi oleh mitra perusahaan telepon seluler seperti HTC, LG , dan Samsung.

2.2.2. *Platform Android*

Android merupakan generasi baru platform mobile, platform yang memberikan pengembangan untuk melakukan pengembangan sesuai dengan yang diharapkan. Sistem operasi yang mendasari Android dilisensikan di bawah GNU, General Public Lisensi versi 2 (GPLv2), yang sering dikenal dengan istilah ”copyleft” lisensi di mana setiap perbaikan pihak ketiga harus terus jatuh dibawah terms. Android didistribusikan dibawah lisensi Apache Software (ASL.Apache2), yang memungkinkan untuk distribusi kedua dan seterusnya. Komersialisasi pengembangan dapat memilih untuk meningkatkan Platform tanpa harus memberikan perbaikan mereka ke masyarakat Open Source. Sebaliknya pengembang dapat keuntungan dari perangkat tambahan seperti perbaikan dan mendistribusikan ulang pekerjaan mereka dibawah lisensi apapun yang mereka inginkan. Pengembang aplikasi Android diperbolehkan untuk mendistribusikan aplikasi mereka dibawah lisensi apapun yang mereka inginkan.

Beberapa *Platform* masa depan Android diantranya :

1. *Complete Plaform*

   Para desainer dapat melakukan pendekatan yang komperhensif ketika mereka sedang mengembangkan *Platform* Android. Android merupakan sistem operasi yang aman dan banyak menyediakan *tools* dalam membangun software dan memungkinkan untuk peluang pengembangan aplikasi.

2. *Open Source Platform*

   Platform Android disediakan melalui lisensi *Open Source*. Pengembangan dapat dengan bebas untuk mengembangkan aplikasi. Android sendiri menggunakan Linux Kernel 2.6

3. *Free Platform*

   Android adalah *Platform*/aplikasi yang bebas untuk *develope*. Tidak ada lisensi atau biaya royalti untuk dikembangkan pada *Platform* Android. Tidak ada biaya keanggotaan diperlukan. Tidak diperlukan biaya pengujian. Tidak ada kontrak yang diperlukan. Android dapat didistribusikan dan diperdagangkan dalam bentuk apapun.

2.2.2.1. *Android 5.1.1 Lolipop*

Android Lollipop adalah versi dari sistem operasi mobile Android yang dikembangkan oleh Google, yang mencakup versi antara 5.0 dan 5.1.1. Yang di rilis preview pada 25 Juni, 2014, tersedia secara resmi melalui resmi over-the -air (OTA) update pada 12 November 2014. Kode sumbernya dibuat tersedia pada 3 November 2014. Salah satu perubahan yang paling menonjol dalam rilis Lollipop adalah user interface didesain ulang dibangun di sekitar bahasa desain yang dikenal sebagai "material design". Perubahan lain termasuk perbaikan notifications, yang dapat diakses dari lockscreen dan ditampilkan dalam aplikasi banner di bagian atas layar. Google juga membuat perubahan internal untuk platform, dengan Android Runtime (ART) secara resmi menggantikan Dalvik untuk meningkatkan kinerja aplikasi, dan dengan perubahan dimaksudkan untuk meningkatkan dan mengoptimalkan penggunaan baterai.

Pada Juni 2015, statistik yang dikeluarkan oleh Google menunjukkan bahwa 12,4% dari semua perangkat Android yang mengakses Google Play telah menjalankan Android Lollipop.

Android 5.1 Lollipop telah mendukung fitur dual-SIM dan panggilan suara HD. serta meningkatkan performa dan stabilitas sistem operasi. Android 5.1 juga membawa kontrol nirkabel seperti Wi-Fi dan Bluetooth ke menu pengaturan cepat atau quick setting. Fitur yang lain adalah device protection yang merupakan fitur keamanan terbaru Google. Jika sudah diaktifkan, ponsel Android tak akan bisa diaktifkan, sekalipun sudah dilakukan factory reset, sebelum memasukkan password akun Google yang sebelumnya dipakai. Fitur semacam ini sebelumnya diperkenalkan oleh Apple di IOS 7 dengan nama Activation Lock. Fitur tersebut dikatakan bisa menurunkan tingkat pencurian Phone di San Francisco sebesar 40%. Setelah Android Lolipop, Google dikabarkan menyiapkan Android versi mendatang yang akan mendukung virtual reality. Meski begitu, belum ada informasi lebih lanjut yang dapat digali dari rencana OS baru tersebut hingga saat ini. fitur yang diusung oleh Android Lollipop memungkinkan pengguna mengontrol pairing Bluetooth langsung melalui fasilitas Quick Settings. Berikut ini adalah daftar fitur yang diusung oleh update OS Android Lollipop:

a. Silent mode ditambahkan setelah menghilang di Android 5.0 Lollipop
b. Perbaikan umum dalam stabilitas sistem
c. Peningkatan manajemen RAM
d. Perbaikan untuk aplikasi yang menutup mendadak *(force close)*
e. Peningkatan manajemen baterai

f. Perbaikan konsumsi berlebihan pada perangkat ketika menggunakan jaringan Wi-Fi
g. Perbaikan masalah dengan koneksi nirkabel
h. Perbaikan masalah dengan fungsi ‘OK Google‘
i. Perbaikan masalah notifikasi
j. Perbaikan masalah sound di beberapa perangkat tertentu
k. Perbaikan dan perubahan lainnya
l. Perubahan dalam palet warna Material Design
m. Dukungan HD Voice
n. Device Protection
o. Dukungan dual SIM

2.2.3. UML

Unified Modelling Language (UML) adalah standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem.

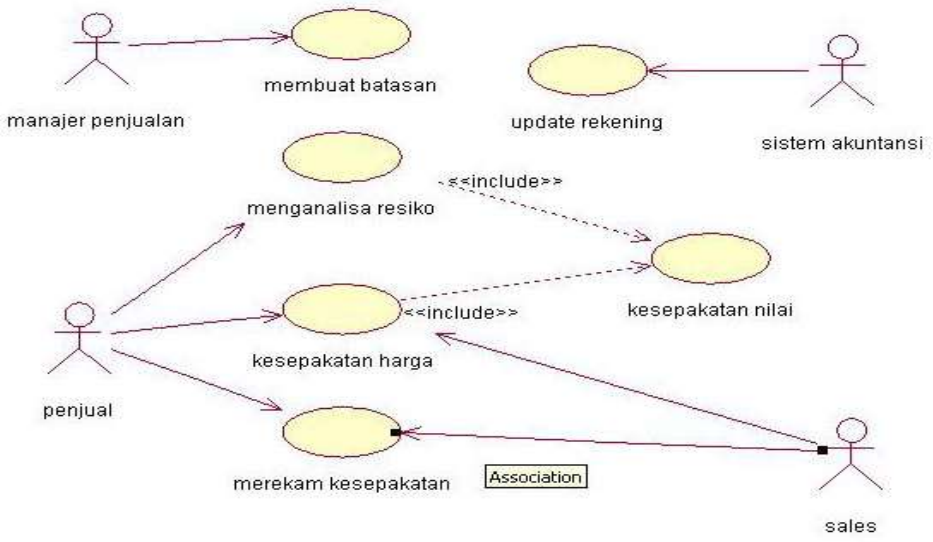
Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET.

2.2.3.1. Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu.

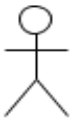
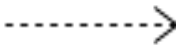
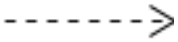
Sebuah use case juga dapat meng-extend use case lain dengan behaviour-nya sendiri. Sementara hubungan generalisasi antar use case menunjukkan bahwa use case yang satu merupakan spesialisasi dari yang lain.

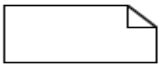
Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Sebuah *Use case* mempersentasikan sebuah interaksi antara aktor dengan sistem contoh proses login ke sistem. Use case diagram menjelaskan manfaat sistem dilihat menurut pandangan orang yang berbeda diluar sistem(actor). Contoh dari *use case diagram* adalah sebagai berikut :



Gambar 2.1 Contoh *Use Case Diagram* (Fowler, 2005)

Berikut ini simbol Use Case Diagram, yang umum digunakan :

SIMBOL	NAMA	FUNGSI
	AKTOR	menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan use case
	DEPEDENCY	hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	GENERALIZATION	hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
	INCLUDE	menspesifikasikan bahwa *use case* sumber secara *eksplisit*.
	EXTEND	menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan
	ASSOCIATION	menghubungkan antara objek satu dengan objek lainnya.

	SYSTEM	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	USE CASE	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.
	NOTE	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.
	COLLABORATION	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).

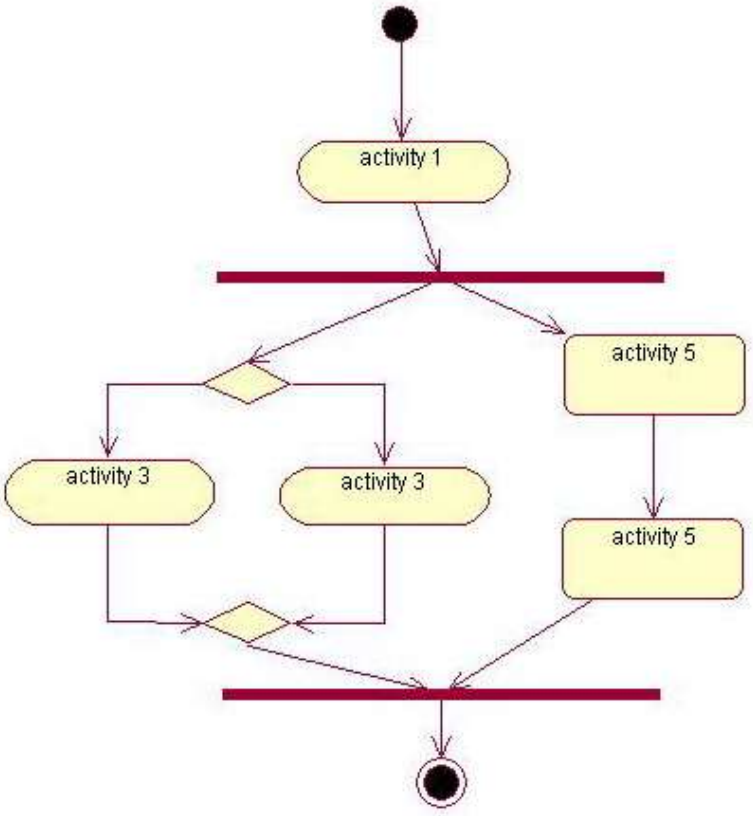
2.2.3.2.Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan *state* diagram khusus, dimana sebagian besar *state* adalah action dan sebagian besar transisi di-*trigger* oleh selesainya *state*

sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behavior* internal sebuah sistem dan interkasi antar subsistem secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Sebuah aktivitas dapat direalisasikan oleh satu *use case* atau lebih. Aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas.

Contoh *activity diagram* sebagai berikut

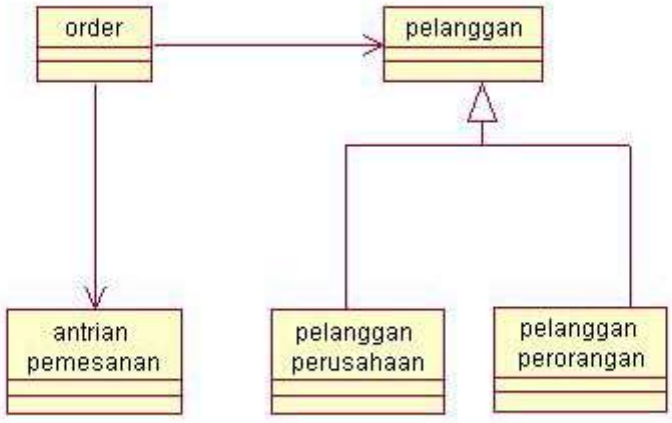


Gambar 2.2 *Activity diagram* (Fowler, 2005)

2.2.3.3.Class Diagram

Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan objek dan merupakan inti dari pengembangan dan desain

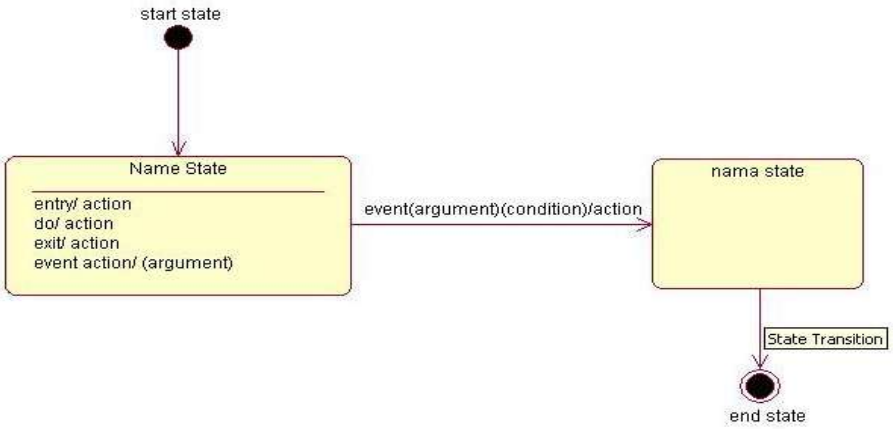
berorientasi objek. *Class* menggambarkan kaadaan suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut. Contoh *class diagram* adalah sebagai berikut



Gambar 2.3 Contoh *Class diagram* sederhana (Fowler, 2005)

a. Statechart Diagram

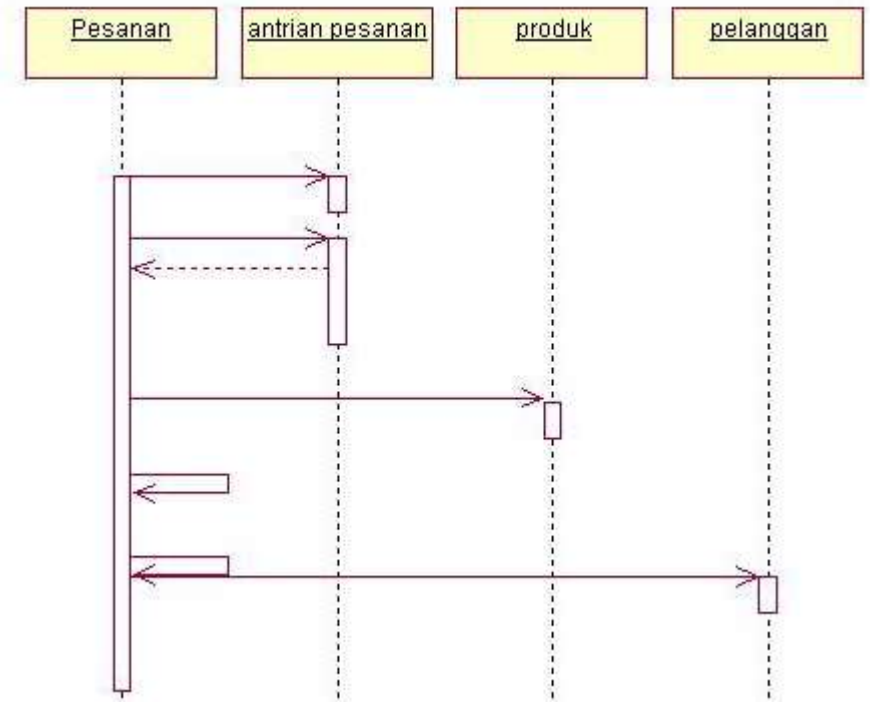
Statechart diagram menggambarkan transisi dan perubahan keadan dari satu *state* ke *state* yang lain. Pada umumnya *statechart diagram* menggambarkan *class* tertentu. contoh gambar statechart sebagai berikut :



Gambar 2.4 *State diagram* (Fowler, 2005)

e. Sequence Diagram

Sequence diagram menggambarkan interaksi antara objek di dalam dan di sekitar sistem dan digunakan untuk menggambarkan sekenario, tipe diagram ini memperlihatkan tahapan-tahapan yang seharusnya terjadi untuk menghasilkan *use case* contoh dari *sequence diagram* :

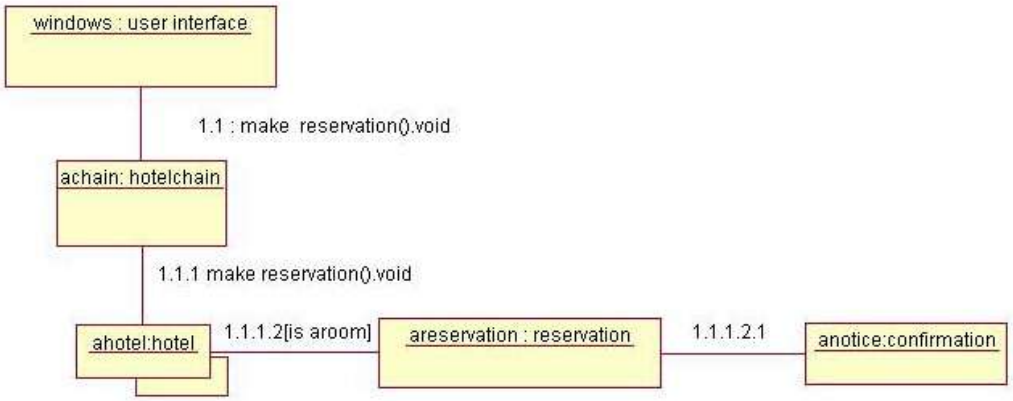


Gambar 2.5 *Squence d iagram* (Fowler, 2005)

f. Collaboration Diagram

Collaboration diagram menggambarkan interaksi antar objek seperti *sequence diagram,* tetapi lebih menekankan kepada peran masing-masing objek dan bukan pada waktu penyampaian message.

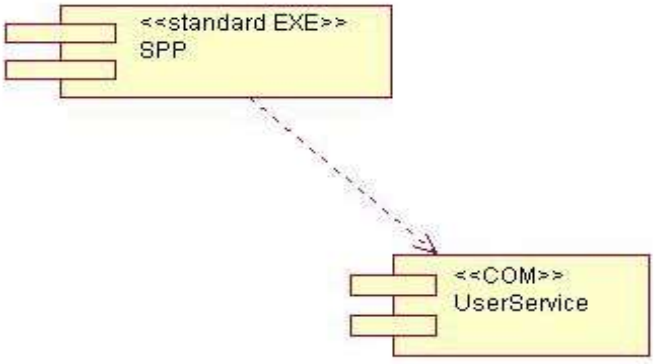
Contoh *collaboration diagram* adalah sebagai berikut :



Gambar 2.6 *Collaboration diagram* (Fowler, 2005)

g. Component Diagram

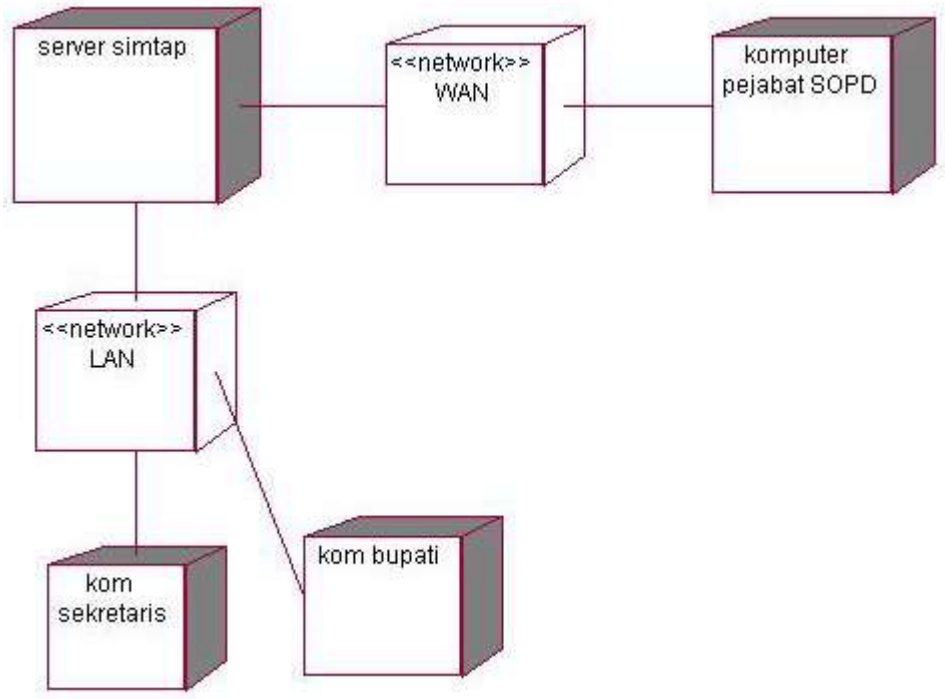
Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan diantaranya. Komponen piranti lunak adalah modul berisi *code*, baik berisi *source code* maupun *binary codeI*, baik *library* maupun *execubate.* Adapun contoh dari *component diagram* adalah sebagai berikut :



Gambar 2.7 *Component Diagram* (A. Suhendar, 2002)

h. Deployment Diagram

Deployment/physical diagram menunjukan susunan fisik sebuah sistem, menunjukan bagian perangkat lunak mana yang berjalan pada perangkat keras mana. Hal utama dalam diagram ini adalah pusat-pusat yang dihubungkan oleh jalur komunikasi. Sebuah pusat adalah sebuah adalah sebuah titik yang dapat mengumpulkan perangkat keras. Contoh dari *Deployment Diagram* seperti dibawah ini :



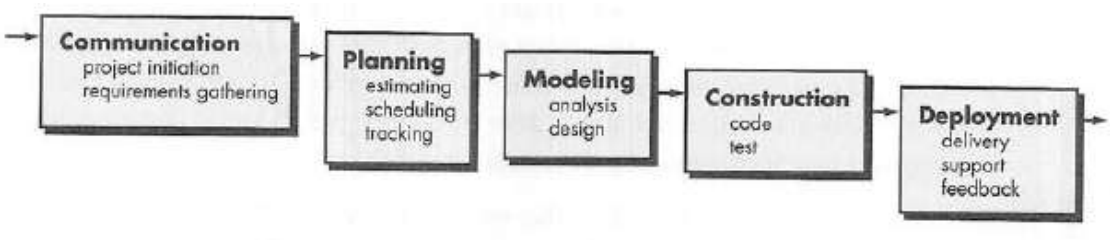
Gambar 2.8 *Deployment Diagram* (A. Suhendar, 2002)

2.2.4.Metode Pengembangan Sistem

2.2.4.1.Metode Waterfall

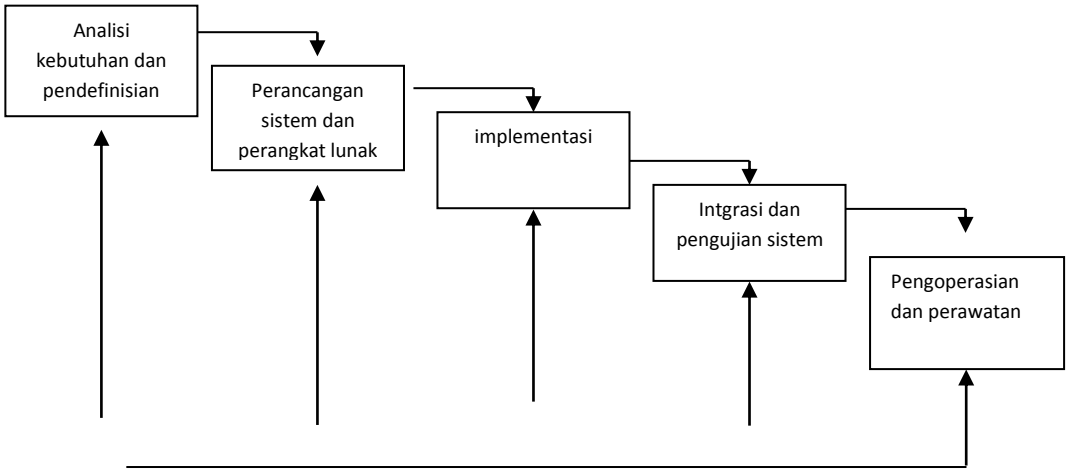
Menurut (Pressman, 2010) , p39 model proses *Waterfall*, yang sering disebut *classic life cycle*, menawarkan pendekatan yang sistematik dan

sekuensial dalam pengembangan *software*. Model proses *Waterfall* diawali dengan pengumpulan kebutuhan pengguna, proses perencanaan, pemodelan, konstruksi, dan penyebaran, berpuncak pada dukungan yang berkelanjutan pada *software* yang telah jadi.



Gambar 2.9 Model Proses *Waterfall*

Waterfall adalah model proses prangkat lunak yang memisahkan antara spesifikasi dan pengembangan hal ini menyebabkan sulitnya merespon perubahan kebutuhan pengguna maka metode ini lebih cocok jika spesifikasi kebutuhan pengguna sudah dipahami dengan baik. Model *waterfall* ini dapat digambarkan sebagai berikut :



Gambar 2.10 metode *waterfall* (Nugroho, 2002)

Dari gambar fase model proses *waterfall* tadi setiap state harus selesai terlebih dahulu dan bila ada perubahan maka harus kembali ke state sebelumnya sehingga membutuhkan waktu yang cukup lama.

2.2.4.2.Metode Prototipe

prototipe dibuat saat pengguna tidak tahu pasti apa yang mereka inginkan baik rincian masukannya, rincian proses dan rincian keluaran yang diinginkan untuk itu dibuatlah prototipe kepada pengguna. Kemudian pengguna menyarankan perbaikan-perbaikan jika terdapat kekurangan sistem yang perlu diperbaiki. Adapun tahapan-tahapannya adalah sebagai berikut :

1. Interaksi dengan pengguna

   Pada tahapan ini penyusun menganalisis apa yang ingin pengguna dapatkan dari sistem/perangkat lunak itu. Sehingga aplikasi yang dihasilkan sesuai dengan kebutuhan pengguna dan sistem.

2. Membuat Prototipe

   Pada tahapan ini akan dibuat sebuah prototipe aplikasi berbasis WAP berdasarkan atas kebutuhan pengguna dan sistem pada tahap interaksi dengan pengguna.

3. Menguji Prototipe

   Tahapan ini adalah proses penilaian terhadap prototipe yang telah dibuat apakah sesuai dengan kebutuhan atau tidak jika tidak maka prototipe akan diperbaiki.

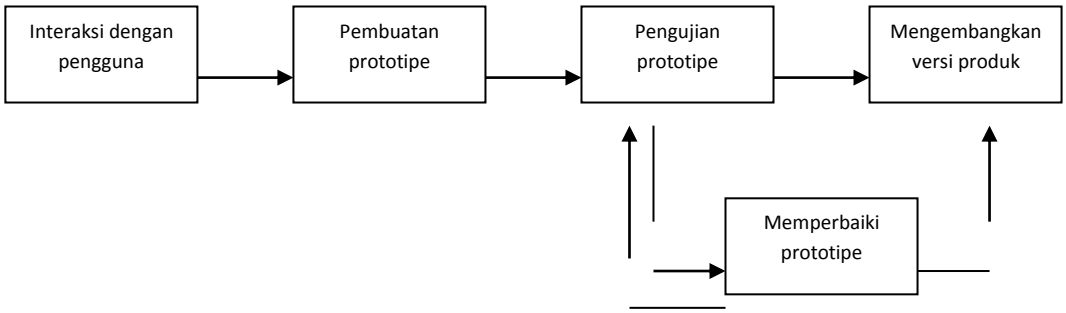
4. Memperbaiki prototipe

   Setelah ditemukan letak kesalahan dari prototipe yang dirancang pada tahapan ini penyusun akan membuat atau memperbaiki prototipe yang ada setelah itu akan di uji lagi sehingga prototipe sesuai dengan keinginan pengguna.

5. mengembangkan versi produk

   setelah aplikasi dapat berjalan dan memenuhi kebutuhan sistem maka aplikasi ini siap dipakai

metode prototipe ini dapat digambarkan sebagai berikut :

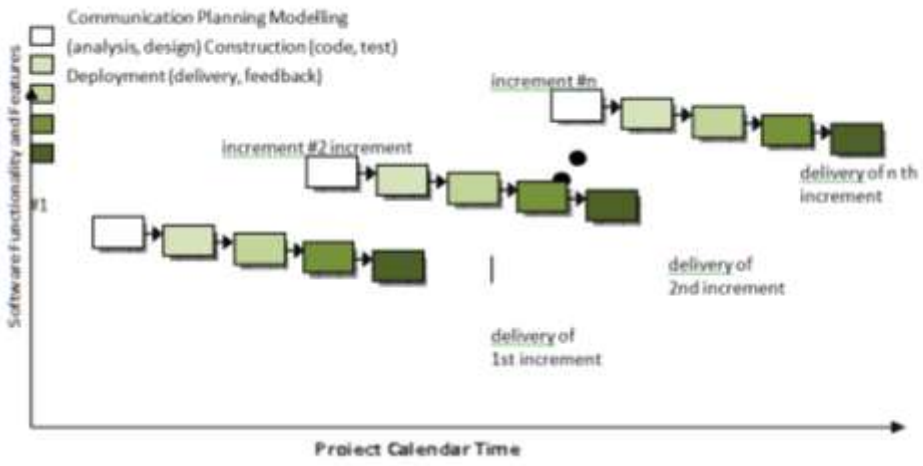


Gambar 2.11 Metode Prototipe (Nugroho, 2002)

2.2.4.3.Metode Proses Incremental

Menurut (Pressman, 2010), model proses *Incremental* mengkombinasikan elemen-elemen aliran proses linier dan paralel. Proses model *Incremental* menerapkan urutan linier seperti waktu yang berlangsung pada kalender. Setiap urutan linier menghasilkan penyampaian "penambahan" dari *software* dengan cara yang mirip dengan

peningkatan yang dihasilkan oleh aliran proses evolusi.

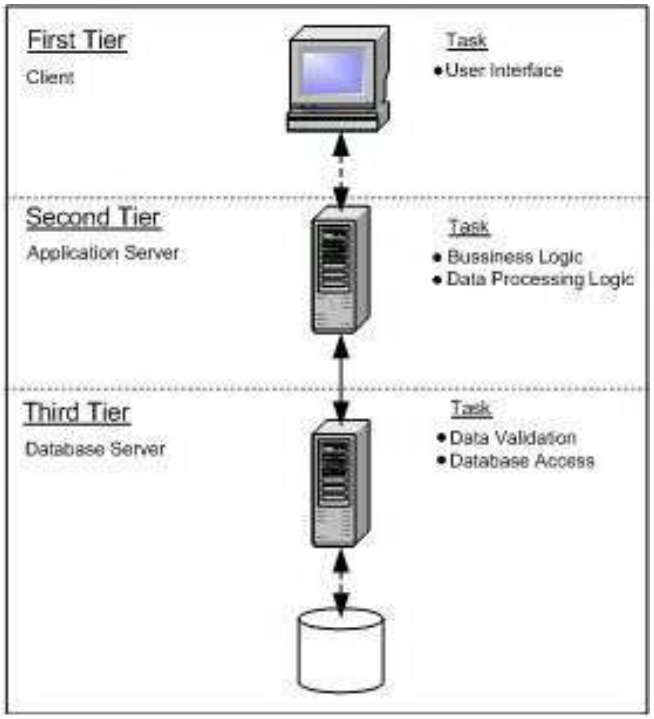


Gambar .12 Model Proses *Incremental*

2.2.5.Arsitektur Three-Tier

Arsitektur *Three-Tier Client-Server* mengajukan 3 lapisan, dimana masing-masing berpotensi dijalankan pada *platform* yang berbeda-beda :

1. Lapisan antarmuka pengguna, yang berjalan pada komputer pengguna (klien).
2. Lapisan *bussiness logic* dan proses. Tingkatan menengah ini dapat berjalan pada *server* dan sering disebut *application server*.
3. Sebuah DBMS yang menyimpan data yang dibutuhkan oleh tingkatan menengah. Tingkatan ini dapat berjalan pada *server* yang terpisah yang disebut *database server*. (Connolly, 2005), p61



Gambar 2.13 Arsitektur *Three-Tier Client-Server*

2.2.6.OBJECT-ORIENTED PROGRAMMING (OOP)

Menurut (Deitel, 2012) *Object-Oriented Programming* meningkatkan produktivitas sehingga mengurangi biaya pengembangan *software*. Pengembang *software* menemukan bahwa menggunakan sebuah desain berorientasi objek modular dan pendekatan implementasi dapat membuat kelompok pengembangan *software* jauh lebih produktif daripada dengan teknik sebelumnya seperti "*structured programming*" — *object-oriented programming* sering lebih mudah untuk dipahami, diperbaiki dan dimodifikasi. Berikut ini adalah beberapa kunci mengenai konsep *object-oriented programming*.

a.*Method* dan *Class*

Pada sebuah program, menjalankan suatu tugas membutuhkan sebuah *method. Method* ini merupakan tempat statemen program yang

melakukan tugasnya. *Method* ini menyembunyikan statemen dari pengguna. Pada Java, dibuat sebuah unit program yang disebut dengan *class* untuk menjadi tempat sekumpulan *method* yang melakukan tugas dari *class* tersebut. (Deitel, 2012), p11

b.*Instantiation*

Object dari *class* harus dibangun sebelum program dapat melakukan tugas yang didefinisikan oleh *method* pada *class*. Proses pembangunan ini disebut dengan *instantiation*. Sebuah *object* kemudian disebut sebagai instansi dari *class*-nya. (Deitel, 2012), p12

c.*Reuse*

Class dapat dipakai berkali-kali untuk membangun *object-object*. Menggunakan kembali *class* yang sudah ada ketika membuat *class- class* dan *program-program* menghemat waktu dan tenaga. Penggunaan kembali juga membantu pembangunan sistem yang lebih handal dan efektif, karena *class* dan komponen yang ada telah melalui pengujian, *debug* dan kinerja *tuning*. (Deitel, 2012), p12

d.*Messages* dan *Methods Calls*

Message dikirimkan ke *object*. Setiap *message* diimplementasikan sebagai sebuah panggilan *method* yang memberitahukan sebuah *method* pada *object* untuk melakukan tugas. (Deitel, 2012), p13

e.*Attributes* dan *Instance Variables*

Sebuah *object* memiliki atribut yang dibawanya bersama sepanjang penggunaan *object* tersebut dalam sebuah program. Atribut ini ditetapkan sebagai bagian dari *class object*. Atribut ditentukan oleh *instance variables class*. (Deitel, 2012), p12

f.*Encapsulation*

Class mengenkapsulasi atribut dan *method* ke dalam *object* – sebuah atribut *object* dan *method* yang berhubungan erat. *Object* dapat berkomunikasi satu sama lain, tetapi biasanya suatu *object* tidak mengetahui bagaimana *object* lainnya diimplementasikan – rincian implementasi tersembunyi dalam *object* itu sendiri. (Deitel, 2012), p13

g.*Inheritance*

Sebuah *class object* baru dapat dibuat dengan cepat dan mudah dengan *inheritance* – *class* baru akan menyerap karakteristik *class* yang sudah ada, memungkinkan pengubahan dan penambahan karakteristik unik sendiri. (Deitel, 2012), p13

Class yang sudah ada disebut dengan *superclass*, dan *class* yang baru disebut dengan *subclass*. Setiap *subclass* dapat menjadi *superclass* untuk *subclass* berikutnya. (Deitel, 2012), p360

Sebuah anggota *public* pada *class* dapat diakses di manapun program ini memiliki referensi ke sebuah *object* dari *class* itu atau salah satu *subclass*-nya. Sebuah anggota *private* pada *class* dapat diakses hanya oleh anggota *class* itu sendiri. Sebuah anggota *protected* pada *superclass* dapat diakses oleh anggota dari *superclass*-nya, anggota *subclass*-nya dan anggota dari *class* lain

yang berada pada *package* yang sama — anggota *protected* mempunyai akses ke *package*. (Deitel, 2012), p360

h.*Polymorphism*

Polymorphism memungkinkan untuk membuat "program di umum" daripada "program dalam spesifik". Secara khusus, *polymorphism* memungkinkan untuk menulis program yang proses *object*-nya berbagi *superclass* yang sama (baik secara langsung atau tidak langsung) seolah-olah mereka semua *object* dari *superclass*. Hal ini dapat menyederhanakan pemrograman. (Deitel, 2012), p397

2.2.7.DELAPAN ATURAN EMAS PERANCANGAN USER INTERFACE

Menurut (Shneiderman's, 2010), pp88-89 terdapat delapan aturan yang digunakan sebagai petunjuk dasar yang baik untuk merancang suatu tampilan antarmuka pengguna. Delapan aturan yang disebut juga dengan *Eight Golden Rules of Interface Design*, yaitu :

a.Berusaha untuk konsisten

Bentuk-bentuk dari konsistensi adalah konsisten untuk aksi yang sejenis atau mirip, konsistensi warna, jenis tulisan, ukuran, dan tata letak dan harus diterapkan pada seluruh aplikasi.

b.Menyediakan *usability universal*

Penerapan *usability universal* adalah dengan menggunakan singkatan dan tombol-tombol *shortcut* yang sudah umum digunakan, sehingga familiar dengan pengguna.

c.Memberikan umpan balik yang informatif

Adanya umpan balik dari sistem untuk setiap aksi yang dilakukan oleh pengguna. Untuk aksi yang sering dilakukan dan tindakan kecil, dapat menggunakan respon yang sederhana. Sedangkan untuk aksi yang jarang dilakukan dan tindakan besar, maka harus menggunakan respon yang lebih substansial.

d.Merancang dialog untuk menghasilkan suatu penutupan

Urutan tindakan sebaiknya diatur dalam pengelompokkan menjadi bagian awal, tengah, dan akhir. Umpan balik yang informatif pada akhir tindakan harus dapat memberikan indikasi bahwa proses yang dilalui sudah benar.

e.Memberikan penanganan kesalahan yang sederhana

Rancangan sistem yang dibuat harus bisa meminimalisir kemungkinan pengguna melakukan kesalahan fatal. Walaupun pengguna melakukan kesalahan, maka sistem harus dapat mendeteksi dan memberikan instruksi sederhana namun jelas agar pengguna dapat memperbaiki kesalahan.

f.Mudah kembali ke tindakan sebelumnya

Dengan mengetahui bahwa aksi yang sudah dilakukan dapat dibatalkan, maka kecemasan pengguna dapat dihilangkan. Sehingga pengguna tidak takut untuk mengekplorasi pilihan-pilihan lain yang belum digunakan.

g.Mendukung *internal locus of control*

Sistem yang baik adalah sistem yang memberikan kendali kontrol kepada pengguna, sehingga pengguna tidak merasa sistem yang mengontrol pengguna.

h. Mengurangi beban ingatan jangka pendek

Ingatan manusia memiliki keterbatasan untuk mengolah informasi, oleh

karena itu sistem tidak boleh membebani ingatan pengguna dalam jangka panjang

2.2.8. STRUCTURED QUERY LANGUAGE (SQL)

Berikut ini beberapa teori yang menyangkut SQL, meliputi pengertian dan kelompok *statement* dalam SQL.

a. Pengertian SQL

Standard Query Language (SQL, dibaca S-Q-L atau juga Sequel) adalah bahasa *query* paling umum yang sekarang baik secara formal dan de facto merupakan bahasa standar untuk *database management system*. (Connolly, 2005), p16

Sebuah *statement* dalam SQL terdiri dari *reserved words* dan *user-defined words*. *Reserved words* merupakan bagian tetap dari bahasa SQL dan memiliki makna yang tetap. *Reserved words* harus dieja dengan tepat seperti yang diharuskan dan tidak dapat dipisahkan. *User-defined words* dibuat oleh pengguna dan merepresentasikan nama dari variasi objek *database* seperti, tabel, kolom, dan sebagainya. (Connolly, 2005), p116

b.Kelompok *Statement* dalam SQL

Statement dalam SQL dikelompokkan menjadi lima, yaitu *Data Defintion Language*, *Data Manipulation Language*, *Data Control Language, Transaction Control Language*, dan Pengendali Programatik.

1)*Data Definition Language* (DDL)

Data Definition Language adalah bahasa yang memperbolehkan seorang *database administrator* (DBA) atau pengguna untuk mendeskripsikan dan menamai suatu entitas, atribut, dan hubungan yang dibutuhkan oleh aplikasi,

bersamaan dengan integritas data dan keamanan datanya. (Connolly, 2005), p40

2)*Data Manipulation Language* (DML)

Data Manipulation Language merupakan bahasa yang menyediakan seperangkat operasi untuk mendukung operasi manipulasi data yang berada dalam *database*. Pengoperasian manipulasi data diantaranya memasukkan data baru, memodifikasi data, mengambil data, maupun menghapus data dari basis-data. (Connolly, 2005), p41

3)*Data Control Language* (DCL)

Data Control Language merupakan bahasa yang digunakan untuk mengendalikan pengaksesan data. Pengendalian dapat dilakukan pada setiap pengguna, tabel, kolom, ataupun operasi yang dapat dilakukan. Perintah-perintah ini berkaitan dengan pengaturan keamanan basis-data. Contoh dari bahasa DCL adalah *GRANT*, *REVOKE*, dan *LOCK*.

4)*Transaction Control Language* (TCL)

Transaction Control Language merupakan bahasa yang digunakan untuk mengontrol pengeksekusian dari sebuah transaksi. Contoh dari perintah TCL adalah *COMMIT*, dan *ROLLBACK*.

5)Pengendali Programatik

Dalam kelompok *statement* ini, perintah-perintah yang termasuk pengendali programatik merupakan pernyataan-pernyataan yang berhubungan dengan pemanfaatan SQL dalam bahasa lain. Pernyataan-pernyataan terdapat pada bahasa konvensional generasi ketiga (3-GL), seperti COBOL. Perintah-perintah yang termasuk dalam kategori ini diantaranya *CLOSE*, *DECLARE*, *FETCH*, dan *OPEN*.