BAB 2
LANDASAN TEORI

Bagian ini berisi tentang beberapa teori-teori yang mendukung dalam proses analisis dan implementasi berdasarkan masalah yang diangkat dalam pembangunan Aplikasi Panduan Perjalanan Di Kota Cirebon Dengan Menggunakan Angkutan Umum.

2.1. Sistem Informasi

Sistem informasi adalah sekumpulan elemen yang saling bekerja sama baik secara manual atau berbasis komputer yang didalamnya ada pengumpulan, pengolahan, pemprosesan dan penyimpanan data untuk mengasilkan informasi yang berguna untuk proses pengambilan keputusan [4].

2.2. Android

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan sistem operasi yang terbuka bagi para pengembang untuk menciptakan aplikasi Android diluncurkan untuk umum pada musim gugur di tahun 2008. Android sangat berkembang pesat di industri karena dua aspek utama yaitu bersifat *opensource* dan model arsitekturya. Sebagai sebuah proyek yang bersifat *opensource*, memungkinkan Android untuk sepenuhnya dipahami dan dianalisis mengenai fitur, penyelesaian pada *bug program* hingga *hardware* [5].

Aplikasi Android terdiri dari komponen yang bebas digabungkan, diikat menggunakan suatu project manifest yang menggambarkan masing-masing komponen dan bagaimana komponen-komponen itu saling berhubungan. Ada enam komponen dalam Android untuk membangun suatu aplikasi yaitu [5]:

1. *Activities*

   Activity merupakan bagian yang paling penting dalam sebuah aplikasi, karena *activity* menyajikan tampilan visual program yang sedang dijalankan oleh *user*. Setiap *activity* dideklarasikan dalam sebuah *class* yang bertugas untuk menampilkan antarmuka yang terdiri dari *views* dan *respons* terhadap *event*.

Setiap aplikasi mempunyai satu buah *activity* atau lebih. Biasanya pasti akan ada *activity* yang pertama kali tampil ketika aplikasi sedang dijalankan. Perpindahan antara *activity* dengan *activity* lainnya diatur melalui sistem, dengan memanfaatkan *activity stack*. Keadaan suatu *activity* ditentukan oleh posisinya pada *activity stack*. Bila suatu *activity* baru dimulai, *activity* yang sebelumnya digunakan maka akan dipindahkan ke tumpukan paling atas. Ketika *activity* dipanggil dan disimpan dalam *activity stack* terdapat 4 kemungkinan kondisi transisi yang akan terjadi [5]:

a. *Active*

   Setiap *activity* yang berada di tumpukan paling atas, maka *activity* tersebut akan terlihat dan menerima masukan dari *user*. Android akan membuat *activity* ini tetap hidup dengan menghentikan *activity* yang berada dibawah tumpukan jika diperlukan. Ketika *activity* sedang aktif, maka yang lainnya akan dihentikan sementara.

b. *Paused*

   Beberapa *activity* akan terlihat tapi tidak terfokus pada kondisi ini disebut *paused*. Keadaan ini terjadi jika *activity* transparan dan tidak *fullscreen* pada layar. Ketika *activity* dalam keadaan *paused*, dia terlihat aktif namun tidak bisa diakses oleh *user*.

c. *Stopped*

   Ketika sebuah *activity* tidak terlihat, maka itulah yang disebut *stopped*. *Activity* akan tetap berada dalam memori dengan semua keadaan dan informasi yang ada. Namum akan di eksekusi oleh sistem jika sistem membutuhkan sumberdaya lebih.

d. *Inactive*

   Kondisi ini ketika *activity* telah dihentikan atau sebelum dijalankan. *Inactive activity* telah ditiadakan dari *activity stack* sehinga perlu *restart* ulang agar dapat tampil dan digunakan kembali.

2. *Services*

   Suatu *services* tidak memiliki tampilan antarmuka, melainkan berjalan di *background* untuk waktu yang tidak terbatas. Komponen *service* diproses tidak terlihat, memperbaharui sumber data dan menampilkan notifikasi.

3. *Content Providers*

   Content providers digunakan untuk mengelola dan berbagi *database*. Data dapat disimpan dalam *file system*, dalam *database* SQLite, atau dengan cara lain yang pada prinsipnya sama. Dengan adanya *content provider* memungkinkan antar aplikasi untuk saling berbagi data.

4. *Intens*

   Intens merupakan sebuah mekanisme untuk menggambarkan tindakan tertentu, seperti memilih foto, menampilkan halaman web, dan lain sebagainya. *Intens* tidak selalu dimulai dengan menjalankan aplikasi, namun juga digunakan oleh sistem untuk memberitahukan ke aplikasi bila terjadi sesuatu, semisal pesan masuk.

5. *Broadcast Receivers*

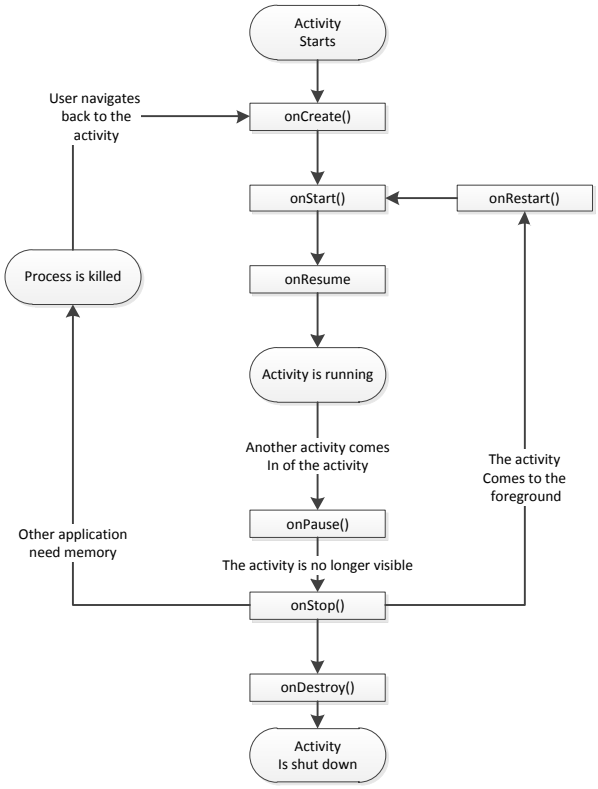
   Broadcast receivers merupakan komponen yang sebenarnya tidak melakukan apa-apa kecuali menerima dan bereaksi menyampaikan pemberitahuan. Sebagian besar *broadcast* berasal dari sistem contohnya baterai sudah hampir habis, informasi zona waktu sudah berubah, atau user telah mengganti bahasa *default* pada perangkat Android. Sama halnya dengan *service*, *broadcast receivers* tidak menampilakan antarmuka *user*. Namun, *broadcast receivers* dapat menggunakan *notification* manager untuk memberitahukan sesuatu kepada *user*.

6. *Notifications*

   Notifications merupakan kerangka pemberitahuan untuk *user*. *Notifications* memberikan isyarat atau peringatan tanpa mengganggu aktivitas yang sedang dilakukan oleh user.

2.2.1 Siklus Hidup Aplikasi Andoid

Siklus hidup aplikasi Android dikelola oleh sistem, berdasarkan kebutuhan user dan sumber daya yang tersedia. Sistem sangat berperan dalam menentukan apakah aplikasi dijalankan, dihentikan sementara, atau dihentikan sama sekali. Jika *user* sedang menjalankan sebuah *activity*, maka sistem akan memprioritaskan aplikasi tersebut. Sebaliknya, jika suatu *activity* tidak terlihat dan sistem membutuhkan sumber daya yang lebih, maka *activity* yang prioritas rendah akan ditutup[9]. Berikut adalah siklus hidup *activity* Android pada gambar 2.1:



Gambar 2.1 Siklus Hidup *Activity* Android [6]

Pengguna mengakses *method* ini pada *activity class*, dan Android akan memanggilnya menurut waktu yang tepat [6].

1. *onCreate( )*

   Dipanggil ketika *activity* pertama kali dibuat, disini *user* melakukan pengaturan statis secara normal, misal membuat *view*, memasukan data dan sebagainya. Selalu diikuti dengan *onStart( )*.

2. *onStart( )*

   Dipanggil sebelum *activity* terlihat oleh pengguna. Selalu diikuti dengan *onResume( ).*

3. *onResume( )*

   Dipanggil sebelum *activity* mulai berinteraksi dengan *user*. Pada titik ini, *activity* berada pada *stack activity* yang paling atas. Selalu diikuti dengan *onPause( ).*

4. *onPause( )*

   Dipanggil ketika sistem akan memulai *activity* lain. *Method* ini biasa digunakan untuk melakukan perubahan yang tidak disimpan kedalam data yang tetap, menghentikan animasi, dan hal lain yang memakai CPU. Diikuti *onResume( )* jika *activity* tetap berada dilayar atau *onStop( )* jika *activity* menjadi tidak terlihat oleh *user*.

5. *onStop( )*

   Dipanggil ketika *activity* tidak lagi terlihat oleh *user*. Ini mungkin terjadi karena *activity* kemudian dihancurkan atau jika *activity* lain yang sudah ada atau yang baru telah dilanjutkan. Diikuti dengan *onRestart( )* jika *activity* kembali berinteraksi dengan *user* atau dengan *onDestroy( )* jika *activity* akan ditutup.

6. *onRestart( )*

   Dipanggil setelah *activity* dihentikan, dan merupakan prioritas untuk dijalankan kembali dan selalu diikuti *onStart( ).*

7. *onDestroy( )*

   Dipanggil sebelum *activity* dihancurkan, ini merupakan panggilan terakhir yang diterima *activity.* Bisa dikatakan *activity* diakhiri (*finish( )*), atau karena sistem secara *temporary* menghancurkan *instant activity* untuk menghemat ruang.

8. *onSaveInstanceState( )*

   Android akan memanggil *method* ini untuk mengizinkan *activity* menyimpan setiap contoh keadaan, seperti posisi kursor dalam sebuah *text field*.

9. *onRestoreInstanceState( )*

   Dipanggil ketika *activity* di inisialisasi ulang sejak keadaan sebelum disimpan dari *onSaveInstanceState( ).*

Android menjalankan setiap aplikasi dalam proses secara terpisah, yang masing-masing memiliki mesin *virtual* pengolah sendiri, dengan melindungi penggunaan memori pada aplikasi. Selain itu juga Android dapat mengontrol aplikasi mana yang layak menjadi prioritas utama. Karena itu Android menjadi sangat sensitif dengan siklus hidup aplikasi dan komponen-komponennya.

2.2.2 Karakteristik Android

Andorid merupakan subset perangkat lunak untuk perangkat mobile yang meliputi sistem operasi, middleware, dan aplikasi init yuang di-release oleh Google. Pada tulisan sebelumnya, kita mengenal SDK (Software Development Kit). SDK adalah suatu tools dan API yang diperlukan untuk mengembangkan aplikasi pada platform atau linkungan Android. Pengembangan aplikasi Android menggunaka bahasa pemrograman Java. Seperti kita ketahui, SDK ini dikembangkan oleh OHA (Open Handsate Alliance). Organisasi OHA ini terdiri atas Googl, Intel, Motorola, Qualcomm, T-Mobile, dan NVIDIA. SDK dapat kita gunakan pada beberapa IDE (Integrated Development and Environment) – software untuk membuat suatu program – akan tetapi, pada tulisan selanjutnya, saya akan menggunakan Eclipse karena OHA secara resmi membuat plugin untuk IDE Eclipse [5]

2.2.3 Perkembangan Android

Dikutip dari http://id.wikipedia.org/wiki/Android_(sistem_operasi) Android telah banyak melakukan perkembangan, berikut adalah perkembangan dari android :

a. Android versi 1.1

Pada 9 Maret 2009, Google merilis Android versi 1.1. Android versi ini dilengkapi dengan pembaruan estetis pada aplikasi, jam alarm, voice search (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan email.

b. Android versi 1.5 (Cupcake)

Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (Software Development Kit) dengan versi 1.5 (Cupcake). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung secara otomatis ke headset Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem.

c. Android versi 1.6 (Donut)

Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang dintegrasikan; CDMA / EVDO, 802.1x, VPN, Gestures, dan Text-to-speech engine; kemampuan dial kontak; teknologi text to change speech (tidak tersedia pada semua ponsel; pengadaan resolusi VWGA.

d. Android versi 2.0/2.1 (Eclair)

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (Eclair), perubahan yang dilakukan adalah pengoptimalan hardware, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan flash untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1.

Untuk bergerak cepat dalam persaingan perangkat generasi berikut, Google melakukan investasi dengan mengadakan kompetisi aplikasi mobile terbaik (killer apps - aplikasi unggulan). Kompetisi ini berhadiah $25,000 bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik.

Dengan semakin berkembangnya dan semakin bertambahnya jumlah handset Android, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi Android. Aplikasi terkenal yang diubah ke dalam sistem operasi Android adalah Shazam, Backgrounds, dan WeatherBug. Sistem operasi Android dalam situs Internet juga dianggap penting untuk menciptakan aplikasi Android asli, contohnya oleh MySpace dan Facebook.

e. Android versi 2.2 (Froyo: Frozen Yoghurt)

Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, intergrasi V8 JavaScript engine yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan WiFi Hotspot portabel, dan kemampuan auto update dalam aplikasi Android Market.

f. Android versi 2.3 (Gingerbread)

Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (gaming), peningkatan fungsi copy paste, layar antar muka (User Interface) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (reverb, equalization, headphone virtualization, dan bass boost), dukungan kemampuan Near Field Communication (NFC), dan dukungan jumlah kamera yang lebih dari satu.

g. Android versi 3.0/3.1 (Honeycomb)

Android Honeycomb dirancang khusus untuk tablet. Android versi ini mendukung ukuran layar yang lebih besar. User Interface pada Honeycomb juga berbeda karena sudah didesain untuk tablet. Honeycomb juga mendukung multi prosesor dan juga akselerasi perangkat keras

(hardware) untuk grafis. Tablet pertama yang dibuat dengan menjalankan Honeycomb adalah Motorola Xoom. Perangkat tablet dengan platform Android 3.0 akan segera hadir di Indonesia. Perangkat tersebut bernama Eee Pad Transformer produksi dari Asus. Rencana masuk pasar Indonesia pada Mei 2011.

h. Android versi 4.0 (ICS :Ice Cream Sandwich)

Diumumkan pada tanggal 19 Oktober 2011, membawa fitur Honeycomb untuk smartphone dan menambahkan fitur baru termasuk membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara offline, dan berbagi informasi dengan menggunakan NFC.

i. Android Versi 4.1 (Jelly Bean)

Diumumkan ada tanggal 27 Juni 2012 meruupakan pembaruan yang meningkatkan fungsi dan kinerja antarmuka pengguna (UI). Pembaruan ini diwujudkan dalam "Proyek Butter", perbaikan ini termasuk antisipasi sentuh, triple buffering, perpanjangan waktu vsync, dan peningkatan frame rate hingga 60 fps untuk menciptakan UI yang lebih halus

j. Android Versi 4.4 KitKat

Google mengumumkan Android 4.4 KitKat dengan pembaruan antarmuka bar status dan navigasi transparan pada layar depan, dengan optimasi kinerja perangkat dengan spesifikasi yang lebih rendah, perluasan fungsionalitas bagi layanan pendengar notifikasi dengan menambahkan dukungan Bluetooth Message Access Profil (MAP).

2.2.4 Kelebihan Android

Sudah banyak platform untuk perangkat selular saat ini, termasuk didalamnya Symbian, iPhone, Windows Mobile, BlackBerry, Java Mobile Edition, Linux Mobile (LiM), dan banyak lagi. Namun ada beberapa hal yang

menjadi kelebihan Android. Walaupun beberapa fitur-fitur yang ada telah muncul sebelumnya pada platform lain, Android adalah yang pertama menggabungkan hal seperti berikut :

1. Keterbukaan, Bebas pengembangan tanpa dikenakan biaya terhadap system karena berbasiskan Linux dan open source. Pembuat perangkat menyukai hal ini karena dapat membangun platform yang sesuai yang diinginkan tanpa harus membayar royality. Sementara pengembang software menyukai karena android dapat digunakan diperangkat manapun dan tanpa terikat oleh vendor manapun.
2. Arsitektur komponen dasar android terinspirasi dari teknologi internet Mashup. Bagian dalam sebuah aplikasi dapat digunakan oleh aplikasi lainnya, bahkan dapat diganti dengan komponen lain yang sesuai dengan aplikasi yang dikembangkan.
3. Banyak dukungan service, kemudahan dalam menggunakan berbagai macam layanan pada aplikasi seperti penggunaan layanan pencarian lokasi, database SQL, browser dan penggunaan peta. Semua itu sudah tertanam pada android sehingga memudahkan dalam pengembangan aplikasi.
4. Siklus hidup aplikasi diatur secara otomatis, setiap program terjaga antara satu sama lain oleh berbagai lapisan keamanan, sehingga kerja system menjadi lebih stabil. Pengguna tak perlu kawatir dalam menggunakan aplikasi pada perangkat yang memorinya terbatas.
5. Dukungan grafis dan suarat terbaik, dengan adanya dukungan 2D grafis dan animasi yang diilhami oleh Flash menyatu dalam 3D menggunakan OpenGL memungkinkan membuat aplikasi maupun game yang berbeda.
6. Portabilitas aplikasi, aplikasi dapat digunakan pada perangkat yang ada saat ini maupun yang akan datang. Semua program ditulis dengan menggunakan bahas pemrograman Java dan dieksekusi oleh mesin virtual Dalvik, sehingga kode program portabel antara ARM, X86, dan arsitektur lainnya. Sama halnya dengan dukungan masukan seperti

penggunaan Keyboard, layar sentuh, trackball dan resolusi layar semua dapat disesuaikan dengan program.

2.2.5 Kekurangan Android

Tak bisa dipungkiri disamping kelebihan tentu saja ada kekurangan dari sistem operasi ini. 1. Koneksi Internet yang terus menerus, kebanyakan ponsel berbasis system ini memerlukan koneksi internet yang simultan alias terus menerus aktif.Koneksi internet GPRS selalu aktif setiap waktu, itu artinya Anda harus siap berlangganan paket GPRS yang sesuai dengan kebutuhan. 2. Iklan – Aplikasi di Ponsel Android memang bisa didapatkan dengan mudah dan gratis, namun konsekuensinya di setiap Aplikasi tersebut, akan selalu Iklan yang terpampang, entah itu bagian atas atau bawah aplikasi.

2.2.6 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 di release 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 member anda kesempatan unutk membuat aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawaan Hadphone/Smartphone. Beberapa fitur-fitur android yang paling penting adalah :

a. Framework : aplikasi yang mendukung pengganti komponen dan reusable.
b. Dalvik Virtual Machine dioptimalkan untuk perangkat mobile
c. Integrated Browser verdasarkan engine open source WebKit.
d. Grafis yang dioptimalkan dan didukung oleh libraries grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1,0 (Opsional Ekselerasi hardware)
e. SQLite untuk penyimpanan data.

f. Media Support yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, AMR, JPG, PING, GIF), GSM Telephony (tergantung Hardware)
g. Bluetooth, EDGE, 3G, dan WiFi (tergantung hardware)
h. Kamera, GPS, Kompas, dan Accelerometer (tergantung hardware)
i. Lingkungan Development yang lengkap dan termasuk pernagkat emulator, tools untuk debugging, profil dan kinerja memori, dan plugin untuk IDE Eclipse.

Untuk source SDK Android ini dapat dilihat dan didownload langsung di situs resmi pengembang SDK Android di http://www.developer.android.com [7]

2.3. JSON (*JavaScript Object Notation*)

JSON (*JavaScript Object Nation*) adalah format pertukaran data ringan, mudah dibaca dan ditulis oleh manusi a, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer. Format ini dibuat berdasarkan bagian dari bahasa pemograman JavaScript, Standar ECMA-262 Edisi ke-3-Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemograman apapuun karena menggunakan gaya bahasa yang umum digunakan oleh *programmer* keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Phyton dll. Oleh karena sifat-sifat tersebut , menjadikan JSON ideal sebagai bahasa pertukaran data[8].

2.4. API (*Application Programming Interface*)

Application programming interface (API) adalah sekumpulan perintah, fungsi, komponen, dan protokol yang disediakan oleh sistem operasi ataupun bahasa pemograman tertentu yang dapat digunakan oleh *programmer* saat membangun perangkat lunak.

Dalam suatu pemograman dibutuhkan setidaknya ribuan *system call* per detik oleh karena itu banyak *programmer* yang menggunakan API. Didalam API terdapat fungsi-fungsi/perintah untuk menggantikan bahasa yang digunakan dalam s*ystem calls* dengan bahasa yang lebih mudah dimengerti oleh *programmer*. Fungsi yang dibuat dengan menggunakan API tersebut kemudian akan memanggil

system call sesuai dengan sistem operasinya. Tidak tertutup kemungkinan nama dari *system call* sama dengan nama API.

Kelebihan pemograman menggunakan API adalah [9]:

1. Portabilitas

   Programmer yang menggunakan API dapat menjalankan programnya dengan sistem operasi mana saja asalkan sudah terinstal API tersebut.

2. Mudah Dimengerti

   API menggunakan bahasa yang lebih terstruktur dan mudah dimengerti daripada bahasa *system call.*

2.5. LBS (*Location Based Services*)

Location based service adalah *service* yang berfungsi untuk mencari dengan teknologi *Global Positioning Service* (GPS) dan *Google's Cell-based Location*. *Map* dan layanan berbasis lokasi menggunakan garis lintang dan garis bujur untuk menentukan letak geografis, namun sebagai *user* kita membutuhkan alamat atau posisi real time kita bukan nilai garis lintang dan garis bujur. *Android* menyediakan *geocoder* yang mendukung *forward* dan *reverse geocoding*. Menggunakan *geocoder*, anda dapat mengkorversi nilai garis lintang dan garis bujur menjadi alamat dunia nyata atau sebaliknya [9].

2.6. Global Positioning System (GPS) dan Assisted Global Positioning System (A-GPS)

Global Positioning System atau yang biasa kita kenal dengan GPS adalah suatu sistem yang dapat membantu mengetahui posisi keberadaan kita saat ini [9]. Dengan mentransmisikan sinyal dari satelit ke perangkat GPS, maka didapatlah data yang akurat. Cara kerja GPS yaitu dengan menerima sinyal dari satelit. Perangkat GPS menentukan lokasi dari minimal 3 satelit yang membentuk kawasan segitiga dengan mencari logitude, latitude, dan data lainnya yang diperlukan.

perbedaan cara kerja GPS dan A-GPS. Dimana GPS receiver yang biasanya digunakan oleh kapal, mobil langsung memperoleh data dari satelit GPS nya langsung, sedangkan A-GPS seperti yang terdapat pada Handphone, data

yang diberikan itu mengambil dari server yang sebelumnya sudah menyimpan data. Oleh karena itu, GPS (perangkat Khusus) membutuhkan waktu yang cukup lama yaitu kurang lebih 12 menit dan sangat berbeda jauh dengan A-GPS yang membutuhkan waktu hanya 20 detik saja.

Untuk keakuratan data. Tentunya sangatlah berbeda jauh. GPS memiliki keakuratan data yang sangat tinggi, dan bisa bekerja dalam keadaan cuaca apapun, dan lokasi manapun. GPS dapat menampilkan keakuratan data yang sangat tinggi walaupun cuaca malam, dan siang, baik di hutan, laut, ataupun berada pada lingkungan yang memiliki gedung-gedung yang tinggi yang bisa mencapai 15 meter atau bahkan dengan teknologi Wide Area Augmentation System (WAAS), keakuratannya bisa mecapai 3 meter.

Sedangkan A-GPS (Assisted Global Positioning System), didesain agar perangkat dapat terhubung dengan satelit secara cepat dan lebih dapat diandalkan daripada menggunakan GPS tunggal, dikarenakan data yang diambil telah disimpan di server GPS

Untuk keakuratan data, A-GPS memiliki keterbatasan yaitu dipengaruhi oleh halangan dari gedung-gedung yang tinggi, kondisi cuaca, dan lokasi pengguna.

A-GPS menggunakan jaringan seluler 2G dan 3G serta koneksi paket data GPRS atau EDGE. Proses mengunduh data memerlukan transmisi melalui provider layan jaringan yang dipakai, oleh karena itu akan dikenakan biaya pada penggunaan A-GPS.

2.7. Best-Path Planning

Dalam tugas akhir ini menggunakan algortima best path planning yang mana algoritma ini berfungsi untuk mencari rute yang tersedia dari suatu tempat ke tempat lainnya dengan jumlah transfer rute sesedikit mungkin [13]. Karena data yang sedemikian besar dan resource perangkat yang dimiliki terbatas, maka pada aplikasi ini jumlah transfer terbatas.

Algoritma

Algorthm

Let o and d denote the numbers assigned to the origin and destination, respectively.

1. *Trivial cases: if o = d, show an appropriate message and return a null plan.*
2. *Direct: if Qo,d = 1, return any service in DService(o, d).*
3. *One transfer: if Qo,d = 2, there must be a location m such that Qo,m = 1 and Qm,d = 1. Combine any route in DService(o;m) and any route in DService(m, d) to obtain a one-transfer plan.*
4. *Two transfers: if Qo,d = 3, there must be different locations m1 and m2 such that Qo;m1 = 1, Qm1,m2 = 1, and Qm2,d = 1. Combine one route from each of DService(o,m1), DService(m1,m2), and DService(m2, d) to obtain a two transfer plan.*

Gambar 2.2 Algoritma Best-Path Planning

Gambar 2.5 adalah algoritma best-path planning yang dipakai sebagai acuan dalam pembuatan aplikasi [13]. Algoritma ini pun lansung diambil dari paper Best-Path Planning for Public Transportation System. Algoritma ini mengecek tahapan demi tahapan apakan perjalanan bisa dilakukan hanya dengan menggunakan 1 angkot saja. Apabila tidak, maka dilakukan pengecekan ke tahapan selanjutnya yaitu dengan satu kali transfer saja, dan begitu pula selanjutnya sampai maksimal dapat dilakukan dengan 3 angkutan umum saja atau dengan 2 kali transfer saja.

Langkah-langkah dari algoritma ini tentu saja harus di indikasikan bahwa rute asal dan rute tujuan tidak sama, seperti yang terlihat di nomor pertama kemudia jika rute asal dan tujuan sudah tidak sama, maka dilakukan pengecekan kembali. Apakah rute tersebut bisa dilakukan dengan menggunakan rute langsung saja. Bisa dilihat pada algoritma pada nomor 2 bahwa apabila $Q_{o,d}$ =1, maka harus kembali kepada Dservice (o,d), kemudian apabila ternyata rute yang dimaksud tidak bisa dilakukan dengan rute langsung, maka kemungkinan yang ada harus melalui 1 kali transfer rute. Caranya adalah dengan menemukan titik pertemuan antara kedua rute asal dan tujuan. Misal $Q_{o,d}$ = 2, maka harus ada lokasi

m yang menemukan kedua rute o dan d seperti $Q_{o,m}$ = 1, dan $Q_{m,d}$ =1. Dimana $Q_{o,m}$ = 1, dan $Q_{m,d}$ =1 adalah rute langsung [10].

Untuk menemukan rute 2 kali transfer dapat dilakukan dengan mencari 2 titik temu dari rute asal dan tujuan. Misal $Q_{o,d}$ =3, maka harus ada 2 lokasi berbeda m1 dan m2 yang menghubungkan 2 rute langsung dari rute yang dicari, seperti $Q_{o,m1}$ =1, $Q_{m1,m2}$ =1, dan $Q_{m2,d}$ =1. Dimana m1 adalah titik temu antara rute langsung dari asal dan rute transfer yang pertama, dan m2 adalah titik temu antara rute transfer yang pertama dan rute tujuan akhir [10]. Tentunya untuk mencari kemungkinan transfer rute semuanya berpusat pada matriks transisi yang telah ada. Karena matriks transisi itu berfungsi sebagai penentu dari transfer rute-transfer rute yang mungkin.

2.8. Matriks Transisi

Pembuatan aplikasi ini memanfaatkan matriks transisi yang digunakan untuk mengetahui jumlah transfer rute yang dibutuhkan dari tempat yang satu ke tempat lainnya, Anggap jumlah tempat yang dilalui oleh angkutan umum adalah n. Maka digunakan matriks transisi dengan ukuran n x n. Selanjutnya memberi nilai pada cell $T_{i,j}$ dari matriks transisi T adalah jumlah rute langsung yang mungkin bagi seseorang untuk berpergian dari tempat i ke tempat j. Otomatis nilai $T_{i,i}$ adalah 0.

Para ahli telah membuktikan bahwa pangkat n dari T adalah jumlah rute yang dapat ditempuh dari lokasi i ke j dengan (n - 1) kali transfer, untuk n ≥ 2.

Definisi standar pangkat n dar T ialah :

$$T_{i,j}^{n} = \sum_k T_{i,j}^{n-1} T_{k,j} \qquad \ldots (1)$$

Dapat diasumsikan bahwa $T_{i,j}^{n-1}$ adalah jumlah rute yang dapat dilakukan untuk perjalanan dari i ke j dengan (n – 2) transfer, yang mana ini berlaku untuk kasus n =2. Kemudian pernyataan pertama dari peersamaan diatas $T_{i,j}^{n-1}$, adalah jumlah rute yang mungkin dapat dilakukan untuk perjalanan dari i ke lokasi penghubung

k dengan (n – 2) kali transfer, dan pernyataan kedua adalah rute langsung yang mungkin dapat dilakukan untuk perjalanan dari k ke j. Dengan mengalihkan dua hasil tersebut akan menghasilkan dua hasil tersebut akan menghasilkan jumlah total rute untuk berpergiaan dari i ke j dan transfer pada k tertentu dengan (n – 1) kali transfer. Dengan demikian, menjumlahkan semua k yang mungkin akan menghasilkan jumlah total rute untuk berpergian dari i ke j dengan (n – 1) kali transfer [10].

2.9. *Object Oriented Analysis and Design*

Konsep OOAD mencakup analisis dan desain sebuah sistem dengan pendekatan objek, yaitu analisis berorientasi objek (OOA) dan desain berorientasi objek (OOD). OOA adalah metode analisis yang memerika requirement (syarat/keperluan) yang harus dipenuhi sebuah sistem) dari sudut pandang kelas-kelas dan objek-objek yang ditemui dalam ruang lingkup sistem. Sedangkan OOD adalah metode untuk mengarahkan arsitektur *software* yang didasarkan pada manipulasi objek-objek sistem atau subsistem [11].

2.9.1. Unfied Modeling Language (UML)

Unified Modeling Language (UML) adalah termasuk ke dalam rumpun jenis pemodelan notasi grafis yang didukung oleh meta-model tunggal. Pemodelan ini berguna untuk membantu dalam menjelaskan dan merancang perangkat lunak yang dibangun dengan *object-oriented* (OO). UML merupakan standar terbuka yang dikelola oleh *Open Management Group* (OMG) yang berada dibawah naungan perusahaan-perusahaan konsorsium terbuka[11]. UML merupakan bahasa pemodelan yang terdiri banyak model diantaranya adalah :

1. *Use case Diagram*

   Use case adalah teknik untuk menggambarkan fungsional sebuah sistem. *Use case* diagram memperlihatkan hubungan diantara *actor* dan *use case*. Aktor mempresentasikan seorang *user* atau subsistem lain yang akan berinteraksi dengan sistem. Sedangkan *use case* merupakan urutan kejadian yang

menggambarkan interaksi antara user dengan sistem. Fungsionalitas sistem didefinisikan ke dalam *use case* dari sudut eksternal sistem yang berguna untuk diuji kelayakan sistem[11].

2. *Sequence Diagram*

   Sequence diagram menggambarkan interaksi diagram yang menunjukan bagaimana kelompok-kelompok objek saling berkolaborasi dalam beberapa behavior, Sebuah *Sequence diagram* secara khusus menjabarkan behavior sebuah skenario tunggal. Diagram tersebut menunjukan sejumlah objek contoh dan pesan-pesan yang melewati objek ini dalam *use case*. *Sequence diagram* adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, display, dan sebagainya berupa pesan/message. *Sequence diagram* digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/even untuk menghasilkan *output* tertentu [11].

3. *Class diagram*

   Class diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai macam hubungan statis yang terdapat diantara mereka. Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendifinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas, atribut mendeskripsikan properti dengan sebaris teks di dalam kotak kelas tersebut. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas [11].

2.9.2. Componenet Diagram

Componen Diagram dapat diartikan sebagai berikut :

1. Component diagram menggambarkan struktur dan hubungan antar komponen peranti lunak, termasuk ketergantungan (dependency) diantaranya.

2. Komponen peranti lunak adalah modul berisi code, baik berisi source code maupun binary code, baik library maupun executable, baik yang muncul pada compile time, link time maupun run time.
3. Pada umumnya komponen terbentuk dari bebrapa class dan/atau package, tapi dapat juga dari komponen-komponen yang lebih kecil.
4. Komponen dapat juga berupa interface, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.

   Tipe-tipe Component :

   Bentuk-bentuk component ada 3 yaitu:

   1. Deployment Component: Yang menjadi basis dari executable system. Contoh deployment component diantaranya: (Dynamic Library Link) file exe, Active X Control, Java Bean dan lain-lain
   2. Work Product Component: Yaitu file-file yang dibutuhkan untuk pembuatan deployment component. Contoh untuk component kedua ini diantaranya file data, file source code dan lain-lain.
   3. Execution Component: Yang dibuat sebagai hasil dari sistem yang akan dijalankan.

Menurut Fowler (2004) hal penting pada component adalah component mewakili potongan-potongan yang independen yang bisa dipesan dan diperbaharui sewaktu-waktu. Dengan demikian, pembagian sistem kedalam component-component lebih banyak didorong oleh kepentingan marketing dari pada kepentingan teknis. Meskipun demikian harus juga diingat bahwa terlalu banyak component juga kurang bagus, karena susah mengatur dan memeliharanya khususnya yang menyangkut masalah versioning [11].

2.10. Pengembangan Perangkat Lunak

2.10.1. Personal Home Page (PHP)

PHP adalah kombinasi antara bahasa pemograman dan aplikasi server PHP merupakan server-side scripting yang digunakan untuk membuat dan menjalankan

aplikasi web server yang dinamis, interaktif dan mempunyai performasi tinggi. Aplikasi server adalah program yang terdiri atas teknik-teknik dalam satu paket yang meliputi :

1. Ketangguhan bahasa pemograman
2. Pengaksesan basis data ke media penyimpanan yang permanent.
3. Mendukung internet protocol, khususnya HTTP dan e-mail.

PHP mampu berhubungan dengan basis data dan dapat diintegrasikan dengan HTML. PHP dapat berjalan dalam web server yang berbeda dalam sistem operasi yang berbeda pula. PHP ditulis dalam bahasa C, sehingga sebagian besar sintak PHP mirip dengan C dan perl. Beberapa keuntungan menggunakan PHP adalah sebagai berikut :

1. Open source, semua source PHP tersedia.
2. PHP diterbitkan secara gratis, tidak ada biaya dalam mengimplementasikan.
3. Cross-platform, PHP dapat berjalan dalam sistem operasi Windows 98, Windows NT dan Macintosh.
4. PHP juga dapat berjalan pada server Microsoft Personal Web Server, IIS dan Aphace.
5. Embedded-html, karena itu PHP mudah dipelajari.[12].

2.10.2. Database Management System (DBMS)

DBMS (Database Management System) adalah suatu system aplikasi yang digunakan untuk menyimpan, mengelola dan menampilkan data. suatu system aplikasi disebut DBMS jika memenuhi persyaratan minimal sebagai berikut:

a. Menyediakan fasilitas untuk mengelola akses data
b. Mampu menangani integritas data
c. Mampu menangani akses data secara bersamaan
d. Mampu menangani backup data

DBMS sudah mulai berkembang sejak tahun 1960an. Kemudian sekitar tahun 1970an mulai berkembang teknologi Relational DBMS yaitu DBMS berbasis relasional model. Relasional model pertama kali dikembangkan oleh

Edgar J. Codd pada tahun 1970. Secara sederhana relasional dapat dipahami sebagai suatu model yang memandang data sebagai sekumpulan tabel yang saling terkait. Hampir semua DBMS komersial dan open source saat ini berbasis Relastional DBMS. Pada tahun 1980an mulai berkembang Object Oriented DBMS (OODBMS). OODBMS berkembang seiring dengan perkembangan teknologi pemrograman berorientasi objek. Saat ini OODBMS juga cukup berkembang namun belum menggeser kepopuleran RDBMS.MySQL adalah perangkat lunak DBMS atau server basis data yang dapat melayani banyak pengguna, dan dapat melakukan proses dalam waktu yang bersamaan (multi threaded). MySQL juga sangat cepat dan fleksibel dan dapat menyimpan data yang berkapasitas besar. Dalam perancangannya, MySQL mengutamakan kecepatan proses, ketangguhan dan kemudahan bagi penggunanya. Pada saat ini MySQL adalah server basis data yang paling banyak diminati oleh banyak perusahaan di dunia. MySQL menyediakan lingkungan yang mudah dan efisien untuk menyimpan data dan mengakses basis data berkapasitas besar lebih cepat dari pada server basis data lainnya. Keuntungan lainnya adalah bahwa MySQL server tidak membutuhkan spesifikasi hardware yang mahal. MySQL server memungkinkan membuat aplikasi client untuk mengakses dan memanipulasi data di server basis data melalui basis data provider. Keuntungan utama menggunakan MySQL server adalah mudah digunakan, berkecepatan tinggi dan membutuhkan sedikit memori serta tidak membutuhkan spesifikasi hardware yang mahal [12].

2.11. Kuesioner

2.11.1. Peranan Kuesioner

Rancangan Kuesioner adalah salah satu pondasi dasar riset, kuesioner merupakan alat untuk mewawancarai seorang, sebuah kuesioner memberikan suatu keterangan dimana pewawancara dapat mencatat jawaban, tanpa kuesioner wawancara tidaka akan teratur. Bagian pengolahan data menggunakan kuesioner yang telah diisi untuk memmbuat analisis jawaban.

2.11.2. Tujuan Kuesioner

1. Tujuan utama kuesioner adalah untuk mengolah informasi akurat dari responden. Periset berusaha memperoleh gambaran paling dekat tentang kondisi penelitian. Informasi yang akurat diperoleh dengan mengajukan pertanyaan yang tepat kepada orang yang tepat.
2. Kuesioner Memberikan struktur pada wawancara sehingga wawancara dapat berjalan lancar. Hal yang penting dalam suatu survei adaalh bahwa semua responden diberi pertanyaan yang sama. Tanpa struktur ini akan ada kekacauan dan tidak mungkin membangun gambaran keseluruhan. Kuesioner berfungsi sebagai alat pengingat pewawancara agar tidak keluar jalur. Bagi responden, kuesioner memberikan urutan pertanyaan yang logis, mengarahkan ke suatu pokok berikutnya.

2.11.3. Skala Pengukuran Variabel

Skala pengukuran variabel dalam penelitian ini mengacu pada Skala Liket (Liket Scale), dimana masing-masing dibuat dengan menggunakan skala 1-5 kategori jawaban, yang masing-masing jawaban diberi score atau bobot banyaknya score antara 1 sampai 5 dengan rincian sebagai berikut :

1. Jawaban SS sangat setuju diberi score 5.
2. Jawaban S setuju diberi score 4.
3. Jawaban BS biasa saja diberi score 3
4. Jwaban KS kurang setuju diberi Score 2
5. Jawana TS tidak setuju diberi Score 1

2.11.4. Teknik Pengolahan Data

Data yang telah terkumpul selanjutnya diolah semua data yang terkumpul kemudian disajikan dalam susunan yang baik dan rapi. Yang termasuk dalam pengolahan data adlaha menghitung frekuensi mengenai pengaruh desain interior terhadap minat berkunjung masyarakat berdasarkan data hasil kuesioner kemudian

diolah untuk mendapatkan nilai presentase. Tahap-tahap pengolahan data tersebut adalah:

1. Penyuntingan

   Semua daftar pertanyaan wawancara, data kuesioner yang berhasil dikumpulkan selanjutnya diperiksa terlebih dahulu dan dikelompokkan.
2. Penyusunan dan Perhitungan Data

   Penyusunan dan perhitungan data dilakukan secara manual dengan menggunakan alat bantu berupa komputer.
3. Tabulasi

   Data yang telah disusun dan dihitung selanjutnya disajikan dalam bentuk tabel. Pembuatan tabel tersebut dilakukan dengan cara tabulasi langsung karena data langsung dipindahkan dari kerangka tabel yang telah disipakan tanpa proses perantara lainnya. (S ingarimbun, 1994: 248) [13].

2.11.5 Analisis data

Tahap-tahap pengolahan data hasil penelitian ini adlah sebagai berikut :

1. Pemeriksaan akan kelengkapan jawaban.

   Pada tahap ini data yang diperoleh diperiksa kembali untuk mencari jawaban dari kuesioner yang tidak lengkap.
2. Tally, yaitu menghitung jumlah atau frekuensi dari masing-masing jawaban dalam kuesioner.
3. Menghitung presentase jawaban responden dalam bentuk tabel tunggal melalui distribusi frekuensi dan presentase. Dengan menggunakan rumus :

$$\mathbf{P=fNx100\%}$$

P : Presentase

f : Frekuensi Data

N : Jumlah sampel yang diolah (Warsito, 1992:59) [14].