**BAB II**

**TINJAUAN PUSTAKA**

1. **Tinjauan Studi**

Beberapa penelitian Android dengan menggunakan Menggunakan Mobile untuk membantu pihak terkait.

1. Penelitian yang dilakukan oleh Deybi Widya Eyrene (2014), yang berjudul Rancang Bangun Aplikasi Pemesanan Tiket Online Kapal Laut Berbasis Anroid, sistem informasi pemesanan tiket merupakan sebuah data bentuk fisik yang di berikan oleh perusahaan kepada pelanggan untuk mendapatkan barang yang tertera di dalamnya. Tiket biasanya berupa kertas yang di dalamnya terdapat penjelasan tertentu yang menunjukan suatu nilai. Tujuan dari penelitian ini adalah untuk merancang aplikasi pemesanan tiket online kapal laut berbasis anroid dengan menggunakan metodologi DAD (Disciplined Agile Delivery) dan memberikan informasi bagi masyarakat yang akan melakukan perjalanan menggunaan trasportasi kapal laut, baik informasi mengenai kapal ,jadwal pelayaran pemesanan tiket ke masing – masing tujuan. Rancang Bangun Pemesanan Tiket *Online* Kapal Laut Berbasis Anroid telah berhasil di bangun dan di inplementasikan dengan fitur aplikasi jadwal kapal, cek tiket, dan boking tiket dan mendaftar sebagai member.
2. Penelitian yang dilakukan oleh Harmaini Wibowo (2010) yang berjudul Analisis Faktor – Faktor Yang Mempengaruhi Waktu Tunggu Kapal Di Pelabuhan Tanjung Emas Semarang. Keterbatasan sarana dan prasana serta keadaan geografis alam yang tidak mendukung di pelabuhan mungkin akan menurunkan kinerja pelabuhan sehingga pelayanan yang di berikan tidak maksimal. Kondisi seperti ini akan menimbulkan masalah yang semakin tingginya waktu tunggu kapal (Waiting) di pelabuhan terutama pada saat kondisi sibuk. Tujuan penelitian ini untuk mengetahui pengaruh permintaan kapal pandu, kesiapan peralatan bongkar muat, produktifitas bongkar muat di dermaga, waktu kedatangan kapal, cuaca dan waktu pengurusan dokumen serta mencari pariabel yang paling dominan yang berpengaruh terhadap waktu tunggu kapal di pelabuhan tanjung emas semarang jumlah sampel penilitian di ambil dengan menggunakan proportional random sampling yaitu dengan mengambil paling banyak 77 kapal yang terdiri dari : 19 kapal penumpang, 31 kapal General Cargo dan 27 kapal petikemas. teknik analisis regresi linier berganda untuk menguji hipotesis di gunakan uji F pada taraf nyata = 0,05, di gunakan mengetahui apakah secara bersama – sama variabel bebas (X) mempunyai pengaruh terhadap variabel terikat (Y). Nilai signifikan F pengujian hipotesis adalah sebesar 0,000 apabilah di bandingkan dengan taraf nyata = 0,05 berarti nilai signifikan F lebih kecil dari tarifnyata = 0,05. Hal ini menunjukan secara bersama – sama variabel bebas permintaan kapal pandu kesiapan peralatan bongkar muat, produktifitas bongkar muat di dermaga, waktu kedatangan kapal, cuaca dan waktu pengurusan dokumen mempunyai pengaruh yang signifikan terhadap variabel terkai yaitu waktu tunggu kapal , berarti hipotesis penilitian ini dapat di terima. Variabel dominan yang mempengaruhi waktu tunggu kapal di tunjukan pada metode stepwise dari regreri yang menyebutkan untuk kapal penumpang variabel yang akan dominan yang mempengaruhi waktu tunggu kapal adalah variabel produktifitas bongkat muat, untuk kapal general cargo yang dominan yang mempengaruhi waktu tunggu kapal adalah waktu pengurusan dokumen, sedangkan kapal petikemas yang mempengaruhi waktu tunggu kapal adalah (permintaan kapal pandu kesiapan peralatan, bongkar muat, kedatangan kapal dan waktu pengurusan dokumen). Koefesien determinasi mendapatkan nilai adjusted untuk kapal penumpang sebesar, 0,527. Dengan demikian 52,7% *Waiting time*  di jelaskan oleh 5 (lima) variabel tersebut, sedangkan sisahnya 47,7% merupakan pengaruh variabel lain yang tidak termaksud di dalam penilitian ini. Koefisien determinasi mendapatkan nilai adjusted untuk kapal general cargo sebesar 0,490 dengan demikian 49,0% *Waiting Time*  di jelaskan 6 (enam) variabel tersebut , sedangkan sisahnya 51% merupakan pengaruh variabel lain yang tidak termaksud di dalam penilitian ini. Koefisien determinasi mendapatkan nilai adjusted untuk kapal petikemas sebesar 0,787 dengan demikian 78,7% *Waiting Time*  di jelaskan oleh 6 (enam) variabel tersebut, sedangkan sisahnya 21,3% merupakan variabel alin yang tidak termaksuk dalam penilitian ini.
3. Penelitian yang dilakukan oleh M.Fikri Shinwani (2016) dengan judul Rancang Bangun Voice Translator Berbasis Android Menggunakan Hidden Markov Model. Bahasa inggri menjadi bahasa yang sangat penting pada saat ini , menjadi ibu bahasa untuk lebih dari 400 juta orang dari seluruh dunia dan akan di gunakan di setiap negara sebagai bahasa komunikasi baik di pakai oleh kepala pemerintahan maupun masyarakat sosial biasa. Namun hal tersebut berbanding terbalik dengan kemampuan berbahasa inggris masyarakat indonesia yang di akui masih sangat minim. Dalam penilitian yang di lakukan EF, indeks kemampuan berbahsa inggris (English Proficiency Index) EPI tahun 2015 masyarakat indonesia masi menduduki 32 dunia di bawah malaysia dan singapura (EF EPI, 2015). Teknik – teknik dalam pengenalan suara telah banyak mengalami perkembangan mulai dari metode ekstraksi fitur bicara terdapat beberapa teknik yang performanya baik di antaranya Linier Predictive Coding (LPC) dan *MelFrequency Cepstral Coefficents* (MFCC) sedangkan untuk pengenalan pola suara digunakan metode *Hidden Markov Model*yang menggunakan pendekatan *stokastik*yang sangat terpopuler karena mampu melakukan pengenalan dengan baik.
4. **Tinjauan Teori**

**2.2.1 Android**

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang buat menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia.

Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau Google Mail Services (GMS), seperti yang terdapat pada smartphone NEXUS keluaran google yang bekerja sama dengan vendor smartphone HTC, dan kedua adalah yang benar–benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai Open Handset Distribution (OHD), jenis yang kedua ini jenis-jenis yang banyak terdapat pada smartphone android di pasaran.

**2.2.1.1 Sejarah Pengembangan Android OS**

android dirilis pertama oleh google pada tanggal 5 november 2007, yang didukung oleh Open Handset Alliance, dan smartphone pertama yang memakai sistem operasi android adalah HTC Dream, yang dirilis pada 22 oktober 2008. pada penghujung tahun ini, android telah merajai penjualan smartphone, mengalah kan NOKIA, Iphone dan Blackberry.

**Android versi 2.0/2.1 code name Eclair ,** diluncurkan pada 3 desember 2009, 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.

**2.2.2. Sistem Informasi Geografis**

**Pengertian Sistem Informasi Geografis (SIG) adalah** sebuah sistem komputer yang memiliki kemampuan untuk mengambil, menyimpan, menganalisa, dan menampilkan informasi dengan referensi geografis (Budianto. 2010.)

Menurut sumber Esri (1990), bahwa sistem informasi geografis adalah kumpulan terorganisasi dari perangkat keras komputer, perangkat lunak, data geografi dan personil yang dirancang secara efisien untuk memperoleh, menyimpan, mengupdate, memanipulasi, menganalisis, dan menampilkan semua bentuk informasi yang bereferensi geografis (Prahasta, Eddy. 2006)

*Definisi Sistem Informasi Geografis adalah* suatu sistem untuk mendayagunakan dan menghasil gunakan pengolahan dan analisis data spasial (keruangan) serta data non- spasial (tabular), dalam memperoleh berbagai informasi yang berkaitan dengan aspek keruangan, baik yang berorientasi ilmiah, komersil, pengelolaan maupun kebijaksanaan. Berikut adalah beberapa keuntungan penggunaan SIG (Hanafi. 2011)

1. SIG mempunyai kemampuan untuk memilih dan mencari detail yang diinginkan, menggabungkan satu kumpulan data dengan kumpulan data lainnya, melakukan perbaikan data dengan lebih cepat dan memodelkan data serta menganalisis suatu keputusan.
2. SIG dengan mudah menghasilkan peta-peta tematik yang dapat digunakan untuk menampilan informasi-informasi tertentu. Peta-peta tematik tersebut dapat dibuat dari peta-peta yang sudah ada sebelumnya, hanya dengan memanipulasi atribut-atributnya.
3. SIG memiliki kemampuan untuk menguraikan unsur-unsur yang terdapat di permukaan bumi menjadi beberapa layer data spasial, dengan layer permukaan  bumi dapat direkonstruksi kembali.

Dengan demikian aplikasi SIG dapat menjawab beberapa pertanyaan berkenaan dengan (Budianto, Eko. 2010.):

1. Lokasi = Ada apa di lokasi tertentu (di lereng gunung, di desa A), apa yang terjadi di lokasi tersebut (rawan banjir, ada deposit emas, curah hujannya tinggi, dan sebagainya).
2. Kondisi = Dimana lokasi jalan yang paling macet, berapa besar potensi tambang yang ada di Kabupaten X dan sebagainya.
3. Kecenderungan/Trend = Seberapa besar tingkat degradasi kawasan hutan lindung di DAS dan sebagainya.
4. Pola = Bagaimana hubungan antara jenis tanah dan produksi gondorukem, bagaimana pola penyebaran penyakit di sekitar kawasan industri kayu dsb.
5. Simulasi/Modeling = Berapa besar menurunnya erosi bila luas hutan di hulu Sungai Jeneberang meningkat sebesar 1.000 hektar.

**2.2.3 Nahkoda Kapal**

Struktur organisasi kapal terdiri dari seorang Nakhoda selaku pimpinan umum di atas kapal dan Anak Buah kapal yang terdiri dari para perwira kapal dan non perwira/bawahan (subordinate crew).

Struktur organisasi kapal diatas bukanlah struktur yang baku, karena tiap kapal bisa berbeda struktur organisaninya tergantung jenis, fungsi dan kondisi kapal tersebut. Selain jabatan-jabatan tersebut dalam contoh struktur organisasi kapal diatas, masih banyak lagi jenis jabatan di kapal, diluar jabatan Nakhoda.  
Misalnya di kapal pesiar ada jabatan-jabatan Bar-tender, cabin-boy, swimming-pool boy, general purpose dan lain sebagainya. Dikapal lain misalnya terdapat jabatan juru listrik (electrician), greaser dan lain sebagainya. Semua orang yang mempunyai jabatan di atas kapal itu disebut Awak kapal, termasuk Nakhoda, tetapi Anak kapal atau Anak Buah Kapal (ABK) adalah semua orang yang mempunyai jabatan diatas kapal kecuali jabatan Nakhoda.

Untuk kapal penangkap ikan masih ada jabatan lain yaitu Fishing master, Boy-boy (pembuang umpan, untuk kapal penangkap pole and Line (cakalang), dlsb.

UU. No.21 Th. 1992 dan juga pasal 341.b KUHD dengan tegas menyatakan bahwa Nakhoda adalah pemimpin kapal, kemudian dengan menelaah pasal 341 KUHD dan pasal 1 ayat 12 UU. No.21 Th.1992, maka definisi dari Nakhoda adalah sebagai berikut:

“ Nakhoda kapal ialah seseorang yang sudah menanda tangani Perjanjian Kerja Laut (PKL) dengan Pengusaha Kapal dimana dinyatakan sebagai Nakhoda, serta memenuhi syarat sebagai Nakhoda dalam arti untuk memimpin kapal sesuai peraturan perundang-undangan yang berlaku “ Pasal 342 KUHD secara ekplisit menyatakan bahwa tanggung jawab atas kapal hanya berada pada tangan Nakhoda, tidak ada yang lain. Jadiapapun yang terjadi diatas kapal menjaditanggung jawab Nakhoda, kecuali perbuatan kriminal.

Misalkan seorang Mualim sedang bertugas dianjungan sewaktu kapal mengalami kekandasan. Meskipun pada saat itu Nakhoda tidak berada di anjungan, akibat kekandasan itu tetap menjadi tanggung jawab Nakhoda. Contoh yang lain seorang Masinis sedang bertugas di Kamar Mesin ketika tiba-tiba terjadi kebakaran dari kamar mesin. Maka akibat yang terjadi karena kebakaran itu tetap menjadi tanggung jawab Nakhoda. Dengan demikian secara ringkas tanggung jawab Nakhoda kapal dapat dirinci antara lain :

1. Memperlengkapi kapalnya dengan sempurna
2. Mengawaki kapalnya secara layak sesuai prosedur/aturan
3. Membuat kapalnya layak laut (seaworthy)
4. Bertanggung jawab atas keselamatan pelayaran
5. Bertanggung jawab atas keselamatan para pelayar yang ada diatas kapalnya
6. Mematuhi perintah Pengusaha kapal selama tidak menyimpang dari peraturan perundang-undangan yang berlaku

**2.2.4 Sistem Mobile Interkoneksi**

Interkoneksi adalah keterhubungan secara fisik dan logika dari jaringan komunikasi umum yang digunakan oleh operator yang sama atau berbeda untuk memungkinkan pengguna dari satu operator untuk berkomunikasi dengan operator yang sama atau operator lainnya, atau untuk mengakses layanan yang disediakan oleh operator lain. Layanan bisa disediakan oleh bagian yang terlibat atau bagian lain yang mendapat akses ke jaringan. (European Commission Directive 12 Juli 2000).

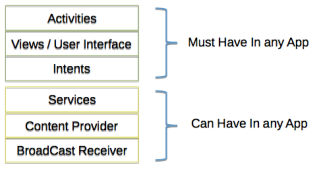
Hakekat interkoneksi antar-jaringan adalah interkoneksi antar Sentral Gerbang (SG) melalui suatu titik interkoneksi (POI – *Point of Interconnection)* yang secara keseluruhan akan membentuk satu kesatuan sistem telekomunikasi end-to-end melalui *switching.* Tiap Penyelenggara jaringan telekomunikasi yang berinterkoneksi wajib menyediakan SG pada sisi masing-masing dan harus memenuhi persyaratan SG.

Dengan perubahan teknologi yang terus berkembang serta iklim kompetisi yang semakin intensif, banyak bentuk-bentuk interkoneksi yang telah dikembangkan. Kesemuanya melibatkan keterhubungan jaringan agar memungkinkan pelanggan dari suatu jaringan untuk berkomunikasi dengan pelanggan-pelanggan lain dari jaringan yang berbeda atau untuk mendapatkan akses terhadap layanan yang ditawarkan oleh operator jaringan lain.

* + 1. **Karakteristik Sistem Mobile**

1. Mendukung penggunaan oleh lebih dari satu user
2. Menjalankan aplikasi yang mampu digunakan oleh lebih dari satu user
3. Stabil (robust), dimana kecil kemungkinan untuk terdapat error pada program.
4. Robustness adalah istilah untuk menunjukkan kemampuan suatu sistem komputer menangani masalah yang terjadi selama digunakan oleh user.
5. Memiliki tingkat keamanan data yang lebih tinggi dari sistem operasi desktop

**2.2.6 Komponen-Komponen Android**



Gambar 2.1 Roll Pada Android

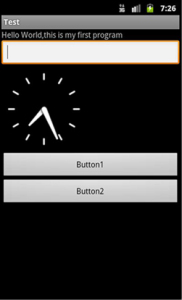
Sudah ada beberapa tutorial yang saya buat tentang android, tapi kurang rasanya kalau saya belum menjelaskan mengenai komponen aplikasi android. Dari awal android muncul hingga hari ini, kebanyakan struktur dasar aplikasi android berubah, bahkan bisa dikatakan sama. Ya memang dari segi fitur dan lain-lain setiap varian android selalu mengalami pengembangan. Secara garis besar, android memiliki empat buah komponen utama untuk aplikasi-aplikasinya, setiap komponen tersebut harus di deklarasikan didalam android manifest agar dikenal oleh OS android dan dapat digunakan.

Setiap komponen-komponen tersebut akan kita tulis dengan bahasa pemograman Java, dan kita harus meng extend sebuah subclass atau super class dari komponen tersebut yang telah disediakan oleh android. Komponen-komponen tersebut adalah ***Activity, Service, Content Provider,****dan****Broadcast Receiver*.**Sejauh ini tutorial yang saya upload baru menyinggung tentang ***activity*,**tapi memang dibandingkan dengan komponen lainnya, biasanya activity akan lebih sering kita gunakan.

Akan saya jelaskan komponen-komponen tersebut berikut cara membuatnya dan pendeklarasiannya di dalam android manifest.

* 1. *Activity*

*Activit*y adalah komponen wajib yang harus ada disetiap aplikasi android, aplikasi android tidak akan memiliki bentuk jika tidak memiliki activity (yaiyalah, mau nampilin apa coba?). *Activity* bertugas menyajikan UI kepada user, activity akan menangani hampir seluruh interaksi user dengan aplikasi tersebut. Gampangnya, jika pembaca sedang membuka sebuah aplikasi android sekarang, itu artinya pembaca sedang melihat sebuah Activity.



Gambar 2.2 *Log Activity*

Activity didalam android didlekarasikan dengan tag **<activity android:name=”.Nama”>….</activity>,**dan sebuah activity dibuat dengan meng extends sebuah subclass activity atau superclassnyya **Activity,**contoh :

* 1. *Service*

Berbeda dengan activity, service tidak memiliki interface, komponen yang satu ini berjalan di background. Biasanya service digunakan untuk melakukan proses-proses tertentu dari aplikasi, namun masih bisa berjalan walaupun aplikasi ditutup, misalnya untuk pemutar audio atau radio, walaupun aplikasi utamanya sudah ditutup tetapi radio atau radio masih bisa kita dengarkan, atau mungkin aplikasi download yang berjalan *dibackgroud*, dll.

Sebuah service bisa berjalan secara independen tanpa terikat denga komponen aplikasi yang lain, jenis service ini kita sebut ***Started* Service,**atau sebuah service berjalan secara terikat dengan komponen aplikasi yang lain, yang kita sebut ***Bound Service.****Bound service* dengan komponen yang terikat padanya bisa saling bertukar informasi, komponen yang terikat pada bound service bisa mengirim request, dan bound service dapat merespone request tersebut.

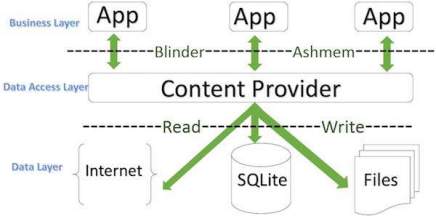
* 1. *Broadcast Receivers*

Jika kita mau membuat sebuah aplikasi yang mampu menangkap event-event tertentu dari OS android, semilah panggilan masuk, panggilan keluar, atau mungkin low battery, maka *Broadcast* Receiver lah jawabannya. Setiap kali sebuah *systemwide event* terjadi pada android OS, maka android OS akan mengirimkan pesan siaran atau broadcast yang bisa ditangkap oleh setiap aplikasi android. Dengan cara seperti ini kita bisa membuat aplikasi untuk memblok panggilan misalnya, dll.

Kita juga bisa membuat pesan broadcast kita sendiri yang bisa ditangkap oleh Broadcast Receiver aplikasi kita atau Broadcast Receiver aplikasi lain, denga cara seperti ini kita bisa memberitahukan aplikasi lain bahwa sebuah event sudah terjadi, misalnya sebuah download sudah selesai, lalu file yang didownload tersebut sudah bisa digunakan oleh aplikasi lain.

* 1. *Content Provider*

Dengan komponen Content Provider, sebuah aplikasi bisa berbagi data dengan aplikasi lainnya, sebuah aplikasi bisa menyimpan data di filesystem, di SQLite, atau di web misalnya. Contoh simplenya ada pada aplikasi sms, atau aplikasi kontak, dimana aplikasi-aplikasi tersebut menggunakan Content Provider untuk dapat berbagai data.



Gambar 2.3 *Content Provider*

Sebuah Content Provider dibuat dengan menjadikan class java kita menjadi subclass dari class ***ContentProvider*,**contoh : Didalam android manifest, content provider dideklarasikan dengan cara membuat tag **<provider android:name”Nama”>….</provider>.**

**2.2.7. Google Maps API**

*Google maps* adalah layanan mapping online yang disediakan oleh google. Layanan ini dapat diakses melalui situs [*http://maps.google.com*](http://maps.google.com). Pada situs tersebut kita dapat melihat informasi geografis pada hamper semua wilayah dibumi. Layanan ini interaktif, karena didalamnya peta dapat digeser sesuai keinginan pengguna, mengubah tingkat *zoom*, serta mengubah tampilan peta.

Bahasa pemograman dari *Google Maps* yang hanya terdiri dari HTML dan Javascript., memungkinkan untuk menampilkan *Google Maps* di *website* lain. Kostumisasi dari aplikasi ini dimungkinkan dengan disediakannnya *Client-side scripts* dan *server-side hooks*.

*Google Maps Application Programming Interface* (API) merupakan suatu fitur aplikasi yang dikeluarkan oleh google untuk memfasilitasi pengguna yang ingin mengintegrasikan *Google Maps* kedalam *website* masing-masing dengan menampilkan data point milik sendiri. Dengan menggunakan *Google Maps API, Google Maps* dapat di-embed pada *website eksternal*. Agar aplikasi *Google Maps* dapat muncul di *website* tertentu, diperlukan adanya *API Key*. *API Key* merupakan kode unik yang digenerasikan oleh google untuk suatu *website* tertentu agar *server google maps* dapat dikenali.

Google Maps API key merupakan bagian yang diberi warna terang. Selain itu juga terdapat bahasa Javascript dengan syntax-syntax Google pada bagian <*head*>. Didalam bagian Javascript dapat ditulis kode untuk mengkostumisasi peta.

*New Gamp2* adalah bagian dimana Google Maps dibentuk sedangkan *map.setCenter* adalah fungsi untuk mengfokuskan titik tertentu pada bagian tengah map. Titik yang dimaksud adalah yang memiliki longitude dan latitude seperti yang dispesifikasikan dengan fungsi new *GlatLng* (37.4419-122.1419) dimana 37.4419 merupakan titik latitude dan -122.1419 merupakan titik longitude. Selain koordinat, tingkat zoom juga dapat ditentukan; pada kode diatas ditujukan dengan angka 13.

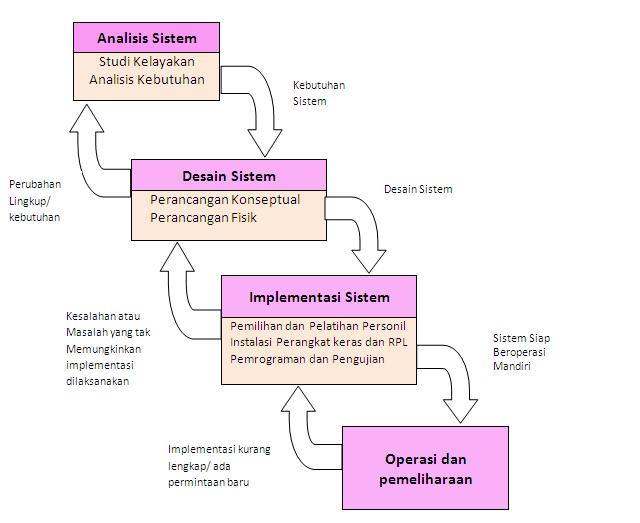
Seiring perkembangannya, terdapat fitur yang sangat mudah bagi para pengguna *Google Maps API*, yaitu geocode alamat, yang memungkinkan pengguna untuk mencari tahu angka suatu koordinat. Hal ini berguna apabila akan memasukan titik koordinat kedalam suatu program.

**2.2.8. Siklus Hidup Pengembangan Sistem**

Pengembangan sistem informasi yang berbasis komputer dapat merupakan tugas kompleks yang membutuhkan banyak sumber daya dan dapat memakan waktu berbulan-bulan bahkan bertahun tahun untuk menyelesaikannya. Proses pengembangan sistem melewati beberapa tahapan dari mulai sistem itu direncanakan sampai dengan sistem tersebut diterapkan, dioperasikan dan dipelihara. Bila operasi sistem yang sudah dikembangkan masih timbul kembali permasalahan-permasalahan yang kritis serta tidak dapat diatasi dalam tahap pemeliharaan sistem, maka perlu dikembangkan kembali suatu sistem untuk mengatasinya dan proses ini kembali ke tahap yang pertama, yaitu tahap perencanaan sistem. Siklus ini disebut dengan siklus hidup suatu sistem (*systems life cycle*). Daur atau siklus hidup dari pengembangan sistem merupakan suatu bentuk yang digunakan untuk menggambarkan tahapan utama dan langkah-langkah didalam tahapan tersebut dalam proses pengembangannya.

Ide dari *systems life cycle* adalah sederhanadan masuk akal. Di *systemslife cycle*, tiap-tiap bagian dari pengembangan sistem dibagi menjadi beberapa tahapan kerja. Tiap-tiap tahapan ini mempunyai karakteristik tersendiri. Tahapan utama siklus hidup pengembangan sistem dapat terdiri dari tahapan perencanaan sistem, analisis sistem, desain sistem, seleksi sistem, implementasi sistem dan perawatan sistem. Tahapan-tahapan seperti ini sebenarnya merupakan tahapan di dalam pengembangan sistem teknik. Pengembangan konstruksi dari gedung, jaringan transmisi tenaga listrik, mesin-mesin dan pabrik-pabrik kimia merupakan contoh dari sistem teknik. Tampak bahwa daur hidup dari sistem tehnik dan sistem informasi dapat sama atau mirip.

Siklus hidup pengembangan sistem dengan langkah-langkah utamanya yang akan digunakan adalah sebagai berikut :



**Gambar 2.1** Siklus Pengembangan Sistem Model Waterfal.

**2.2.9 Analisa Sistem**

Analisa sistem (*system Analisa*) dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh kedalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannaya.

Analisa sistem adalah spesialis yang mempelajari masalah dan kebutuhan sebuah organisasi untuk menentukan bagaimana orang, data, proses, dan teknologi informasi dapat mencapai kemajuan terbaik untuk bisnis.

Analisis sistem adalah *Stakeholder* yang berperan sebagai fasilitator atau pelatih, menjembatani jurang komunikasi yang dapat secara alamiah berkembang antara pemilik dan pengguna *system nonteknis* atau desainer dan perkembangan sistem teknis.

Whitten, et al. (2004 :33) menggungkapkan ”*system analysis* adalah study domain masalah bisnis untuk solusi”.

Impak teknologi objek sangat berarti dalam dunia analisis dan desain sistem. Sebelum ada teknologi objek, kebanyakan bahasa pemrograman didasarkan pada apa yang disebut metode yang terstruktur (*structured method*). Contohnya COBOL (bahsa yang domain O, C, fortan, Pascal dan PL/i. Maka, metode analisis dan desain berorientasi objek telah muncul sebagai pendekatan terpilih untuk membangun kebanyakan sistem informasi saat ini.

Sebagai tambahan keahlian analisis dan desain sistem formal, seorang analis harus mengembangkan atau memiliki keahlian lain, pengetahuan, dan karakter untuk menyelesaikan pekerjaan. Hal ini termasuk :

1. Pengalaman dan keahlian pemograman komputer

Sulit untuk membayangkan bagaimana para analisis sistem dapat dengan cukup mempersiapkan bisnis dan spesifikasi teknis untuk programer jika mereka tidak memiliki pengalaman pemrograman tingkat tinggi.

1. Pengetahuan umum proses dan teknologi bisnis

Analis sistem harus mampu berkomunikasi dengan para ahli bisnis untuk memperoleh pemahaman masalah kebutuhan mereka. Untuk analis, paling tidak sebagian dari pengetahuan ini datang hanya dari pengalaman. Pada saat yang sama analis yang terinspirasi harus mengambil manfaat dari setiap kesempatan untuk menyelesaikan mata kuliah teori bisnis dasar.

Tahap analisis merupakan tahap yg kritis dan sangat penting karena kesalahan didalam tahap ini akan menyebabkan juga kesalahan ditahap selanjutnya. Tahap analisa sistem mencakup studi kelayakan analisis kebutuhan.

1. Studi kelayakan

Studi kelayakan digunakan untuk menentukan kemungkinan keberhasilan solusi yang diusulkan. Tahapan berguna untuk memastikan bahwa solusi yang diusulkan tersebut benar-benar dapat dicapai dengan sumber daya dan dengan memperhatikan kendala yang terdapat pada perusahaan derta dampak terhadap lingkungan sekeliling. Tugas2 yang tercakup dalam studi kelayakan meliputi :

1. Penentuan masalah dan peluang yang dituju sistem.
2. Pembentukan sasaran sistem baru secara keseluruhan.
3. Pengidentifikasian para pemakai sistem.
4. Pembentukan lingkup sistem.

Selain itu, selama dalam tahapan studi kelayakan sistem analisis juga melakukan tugas-tugas sebagai berikut :

1. Pengusulan perangkat lunak dan perangkat keras untuk sistem baru.
2. Pembuatan analisis untuk membuat atau membeli aplikasi.
3. Pembuatan analisis biaya/manfaat
4. Pengkajian terhadap resiko proyek.

Studi kelayakan diukur dengan memperhatikan aspek teknologi, ekonomi, faktor organisasi dan kendala hukum, etika, dan yang lain (Turban, et, al, 1999 dalam Abdul Kadir, 2003:403).

1. Analisis Kebutuhan

Analisis kebutuhan dilakukan untuk menghasilkan spesifikasi kebutuhan (disebut juga spesifikasi fungsional). Spesifikasi kebutuhan adalah spesifikasi yang rinci tentang hal-hal yang akan dilakukan sistem ketika diimplementasikan. Spesifikasi ini sekaligus dipakai untuk membuat kesepakatan antara pengembang sistem, pemakai yang kelak akan menggunakan sistem, manajemen, dan mitra kerja yang lain (misal auditor inernal).

Analisis kebutuhan ini diperlukan untuk menentukan keluaran yang akan dihasilkan sistem, masukan yang diperlukan sistem, lingkup proses yang digunakan untuk mengolah masukan menjadi keluaran, volume data yang akan ditangani sistem, sejumlah pemakaian dan ketegori pemakai, serta kontrol terhadap sistem.

Didalam tahap analsis sistem ini terdapat langkah-langkah yang harus dilakukan oleh analasis sistem, yaitu sebagai berikut :

1. *Identify*, yaitu mengidentifikasi masalah.

Mengidentifikasi (mengenai) masalah merupakan langkah pertama yang dilakukan dalam tahap analisis sistem. Masalah (*problems*) dapat didefinisikan sebagai suatu pertanyaan yang diinginkan untuk dipecahkan. Tahap identifikasi masalah sangat penting karena akan menentukan keberhasilan pada langkah-langkah selanjutnya.

1. *Understand*, yaitu memahami kerja dari sistem yang ada.

Langkah kedua dari tahap analisis sistem adalah memahami kerja dari sistem yang ada. Langkah ini dapat dilakukan dengan mempelajari operasi dari sistem ini diperlukan data yang dapat diperoleh dengan cara melakukan penelitian.

1. *Analyze*, yaitu menganalisis sistem tanpa report.

Langkah ini dilakukan berdasarkan data yang telah diperoleh dari hasil penelitian yang telah dilakukan.

1. *Report*, yaitu membuat laporan hasil analisis.

Tujuan utama dari pembuatan laporan hasil dilakukan :

1. Pelaporan bahwa analisis telah selesai dilakukan
2. Meluruskan kesalah pengertian mengenai apa yang telah ditemukan dan dianalsis oleh analsis sistem tetapi tidak sesuai menurut manajemen.

**2.2.10. Desain Sistem**

Menurut Robert J. Verzello/jhon Reuter III Definisi desain sistem : ”Tahap setelah analisis dari siklus pengembangan sistem : pendefinisian dari kebutuhan-kebutuhan fungsional dan persiapan untuk merancang bangun implementasi, menggambarkan bagaimana suatu sistem dibentuk”.

Hariyanto (2004 : 421) mengungkapkan, perancangan (*design*) sistem adalah strategi tingkat tinggi untuk membuat keputusan-keputusan berikut :

1. Pengorganisasian sistem menjadi subsitem-subsistem.
2. Alokasi subsistem kekomponen-komponen perangkat keras dan perangkat lunak.
3. Keputusan-keputusan konseptual dan kebijakan utama untuk membentuk kerangka kerja rancangan inti.

**2.2.11. Analisis Berorientasi Objek**

Hasil dari analisis berorientasi objek merupakan suatu masukan bagi perancangan sistem/perangkat lunak berorientasi objek. Pada model analisis ini memungkinkan tersedianya informasi detail mengenai skenario penggunaan sistem dan menetapkan landasan untuk model implementasi dan lingkungan dengan mengidentifikasi elemen struktur sistem. Kelas-kelas objek yang meliputi atribut-atribut, metode-metode, dan asosiasi yang didapatkan dari hasil analisis merupakan suatu kerangka kerja saat perancangan dan analisis tersebut harus di rancang untuk implementasi.

1. **Perancangan Sistem**

Perancangan sistem merepresentasikan arsitektur perangkat lunak secara umum yakni :

1. Arsitektur konseptual berkaitan dengan model kelas statik dan koneksi-koneksidiantara komponen-komponen model.
2. Arsitektur model mendeskripsikan pembagian sistem menjadi subsistem-subsistem atau modul-modul ini saling berkomunikasi.
3. Arsitektur kode mendefinisikan bagaimana kode program diorganisasikan menjadi berkas-berkas dan direktori-direktori dan dikelompokkan menjadi pustaka-pustaka.
4. Arsitektur eksekusi berfokus pada aspek dinamis dari sistem dan komunikasi antara komponen-komponen sebagai task dan operasi yang dieksekusi.
5. **Perancangan Objek**

Perancangan objek berfokus pada deskripsi objek dan interaksinya antara satu objek dengan objek yang lainnya secara rinci. Untuk spesifikasi rinci dan struktur data untuk atribut dan rancangan prosedural dari semua operasi selama perancangan objek. Sementara ketampakan diri semua atribut kelas harus didefinisikan dan antarmuka antara objek-objek dielaborasi untuk mendefinisikan rincian-rincian dari model pelawatan peasan secara lengkap.

1. **Perancangan Antar Muka manusia**

Pandangan model pemakai menuntun keproses perancangan antarmuka pemakai, menyediakan skenario yang dielaborasi secara iterasi untuk menjadikannya sebagai sekumpulan kelas-kelas antar muka.

1. **Perancangan Manajemen Data**

Perancangan manajemen data adalah untuk mendukung persistent objek yang menetapkan sekumpulan kelas-kelas dan kolaborasi-kolaborasi yang memungkinkan sistem mengola persistent data (berkas-berkas dan basis data) yang mengimplementasikan persistent objek, dimana pada umumnya manajemen data dirancang secara brlapis. Perancangan manajemen data ini meliputi perancangan atribut-atributdan operasi-operasi yang diperlukan untuk mengola objek-objek.

1. **Perancangan Manajemen Task**

Perancangan Manajemen Task menetapkan infrastruktur yang mengorganisasikan subsistem-subsistem menjadi task-task dan kemudian mengelola konkurensi task. Perancangan (*design*) sistem pada pengembangan sistem/perangkat lunak yang berorientasi objek pada umumnya menggunakan bahasa modeling UML (*Unified Modeling Language*) dan menyajikannya dalam bentuk diagram-diagram yang nantinya akan menjadi acuan bagi pemrogram untuk melakukan coding (pembangunan program).

Adapun diagram-diagram tersebut meliputi seperti yang disajikan pada tabel berikut :

**Tabel 2.2 Daftar Pengelompokkan Diagram Pada UML**

| No | Nama Diagram | Kategori | Deskripsi |
| --- | --- | --- | --- |
| 1 | Class | Struktur | Mengilustrasikan sekumpulan kelas, peket dan hubungan yang merinci satu aspek tertentu dari sistem. |
| 2 | Object | Struktur | Snapshot dari sistem mengilustrasikan hubungan statik yang terjadi diantara objek-objek. |
| 3 | Componen | Struktur | Berkaitan dengan hubungan statik yang ada antara komponen-komponen perangkat lunak yang dipasang. Contohnya file exe, dll, ocx, beans, dan sebagainya. |
| 4 | Deployment | Struktur | Mendeskripsikan topologi fisik sistem. Umumnya termsuk beragam simpul pengolahan, direalisasikan dalam bentuk perangkat (printer, modem) atau pemroses. |
| 5 | Use Case | Perilaku | Menunjukkan sekumpulan use case dan hubungan antara keduanya. Diagram use case berkontribusi ke organisasi model yang efektif juga merupakan pemodelan inti perilaku sistem. |
| 6 | Activity | Perilaku | Memodelkan aliran aktifitas diantara proses-proses. Kebanyakan berguna untuk merinci perilaku use case. |
| 7 | State | Perilaku | Mengilustrasikan perilaku yang berhubungan dengan state dari objek. Transisi-transisi di antara state-state membantu mengidentifikasi dan memvalidasi perilaku kompleks. |
| 8 | Sequence | Perilaku | Tipe diagram interaksi yang mendeskripsikan pesan-pesan berbasis waktu yang dikirim objek-objek. |
| 9 | Collaboration | Perilaku | Tipe diagram interaksi yang mendeskripsikan layout organisasi dari objek-objek yang mengirim dan menerima pesan. |

Sumber : (Hariyanto, 2004:335-336)

Berikut beberapa notasi pengklasifikasi (clasifier) yang digunakan dalam bahasa modeling UML seperti disajikan pada tabel berikut :

Tabel 2.3 Simbol-Simbol Actifity Diagram

|  |  |
| --- | --- |
| Simbol | Keterangan |
|  | *Action/actifity* |
|  | *Object/class* |
|  | *Control/object flow* |
|  | *Initial node* |
|  | *Actifity final node* |
|  | *Decision node* |
|  | *Merge node* |
|  | *Fork node* |
|  | *Join node* |

Sumber: (Alan Dennis, 2010)

Tabel 2.4 Simbol-Simbol Use Case Diagram

|  |  |
| --- | --- |
| Simbol | Keterangan |
|  | *Actor* |
|  | *UseCase* |
|  | *Association relationship* |
|  | *Include relationship* |
|  | *Extend relationship* |
|  | *Generalization relationship* |

*Sumber: (Alan Dennis, 2010)*

Tabel 2.5 Simbol-Simbol Class Diagram

|  |  |
| --- | --- |
| Simbol | Keterangan |
|  | *Class/Object* |
|  | *Interface* |
|  | *Enumeration* |
|  | *Package* |
|  | *Association* |
|  | *Aggregation* |
|  | *Composition* |
|  | *Dependency* |
|  | *Inheritance* |

*Sumber: (Alan Dennis, 2010)*

**2.2.12. Implementasi sistem**

Whitten, et al, (2004 : 34) mengungkapkan: ” *Sistem Implementation* adalah konstruksi, instalasi, pengujian, dan pengeriman sistem kedalam produksi (artinya operasi sehari-hari) ”.

Sistem telah dianalisa dan didesain secara rinci dan teknologi telah diseleksi dan dipilih. Tiba saatnya sekarang sistem untuk diimplementasikan (diterapkan). Tahap implementasi sistem merupakan tahap meletakkan sistem supaya siap untuk dioperasikan. Tahap implementasi sistem dapat terdiri dari langkah-langkah sebagai berikut :

1. Menerapkan rencana implementasi.

Rencana implementasi merupakan kegiatan awal dari tahap implementasi sistem. Rencana implementasi dimaksudkan terutama untuk mengatur biaya dan waktu yang dibutuhkan selama tahap implementasi.

1. Melakukan kegiatan implementasi.

Kegiatan implementasi dilakukan dengan dasar kegiatan yang telah direncanakan dalam rencana implementasi.

Kegiatan-kegiatan yang dapat dilakukan dalam tahap implementasi ini adalah sebagai berikut :

1. Pemilihan dan pelatihan personil.

Telah diketahui bahwa manusia merupakan faktor yang perlu dipertimbangkan dalam sistem informasi. Jika sistem informasi ingin sukses, maka personil-personil yang terlibat harus diberi pengertian dan pengetahuan yang cukup tentang sistem informasi dan posisi serta tugas mereka.

1. Persiapan Tempat dan Instalasi Perangkat Keras dan Perangkat Lunak.

Jika peralatan baru akan dimiliki, maka tempat atau ruangan untuk peralatan ini perlu dipersiapkan terlebih dahulu. Keamanan fisik dari tempat ini perlu juga dipertimbangkan. Sistem komputer yang besar membutuhkan tempat dengan lingkungan yang lebih harus diperhitungkan.

1. Pemograman daan Pengetesan Sistem

Pemograman merupakan kegiatan menulis kode program yang ada akan dieksekusi oleh komputer. Kode program yang ditulis oleh pemrogram harus berdasarkan dokumentasi yang disediakan oleh analis sistem hasil dari desain sistem secara rinci. Sebelum program diterapkan, maka program harus terlebih dahulu bebas dari kesalahan-kesalahan. Oleh sebab itu, program harus diuji untuk menemukan kesalahan-kesalahan yang mungkin dapat terjadi. Program dites untuk tiap-tiap modul dan dilanjutkan dengan pengetesan untuk semua modul yang telah dirangkai.

1. Pengetesan sistem

Pengetesan sistem biasanya dilakukan setelah pengetesan program. Pengetesan program dilakukan untuk memeriksa kekompakan antar komponen sistem yang diimplementasikan. Tujuan utama dari pengetesan sistem ini adalah untuk memastikan bahwa elemen-elemen atau komponen-komponen dari sistem telah berfungsi sesuai dengan yang diharapkan.

**2.2.13. Pemeliharaan Sistem**

”Pemeliharaan sistem adalah proses penguban sistem setelah beroperasi dan digunakan” (Hariyanto 2004 : 603).

”Pemeliharaan sistem adalah tahap dimana kita mulai pengoperasian sistem dan jika diperlukan melakukan perbaikan-perbaikan kecil” (Nugroho 2010 : 7).

Dari kedua pendapat diatas memiliki pandangan sama dimana setelah sistem operasikan dapat saja dilakukan perubaha, apakah itu sebagian kecil dari sistem ataukah secara keseluruhan dengan harapan untuk dapat memenuhi kebutuhan dari pengguna sistem tersebut.

**2.2.14. Teknik Pengujian Sistem**

Pengujian sistem adalah elemen kritis dari jaminan kualitas perangkat lunak dan mempresentasikan kajian pokok dari spesifikasi, desain, dan pengkodean. Tujuan dari pengujian ini adalah diharapkan dengan minimal tenaga dan waktu untuk menemukan berbagai potensi kesalahan dan cacat. Harus didasarkan pada kebutuhan berbagai tahap pengembangan, desain dan dokumen lain atau program yang dirancang untuk menguji struktur internal, dan menggunakan contoh-contoh ini untuk menjalankan program untuk mendeteksi kesalahan.Pengujian sistem informasi harus mencakup pengujian perangkat lunak, pengujian perangkat keras dan pengujian jaringan. Pengujian hardware, jaringan pengujian berdasarkan indikator kinerja spesifik yang akan digunakan di sini pengujian lebih jauh adalah pengujian perangkat lunak.

**2.2.13.1 *WhiteBox***

Pengujian *white-box (glass box)*, adalah metode desain *test case* yang menggunakan struktur kontrol desain prosedural untuk memperoleh *test case*. Dengan menggunakan metode pengujian *white-box*, perekayasa sistem dapat melakukan *test case* untuk memberikan jaminan bahwa :

1. Semua jalur independen pada suatu modul ditelusuri minimal 1 (satu) kali.
2. Semua jalur keputusan logis *True/False* dilalui.
3. Semua *loop* dieksekusi pada batas yang tercantum dan batas operasionalnya.
4. Struktur data internal digunakan agar validitas terjamin.

Pengujian *white-box* bisa dilakukan dengan pengujian *basis path*, metode ini merupakan salah satu teknik pengujian struktur kontrol untuk menjamin semua statemen dalam setiap jalur independen program dieksekusi minimal 1 kali dan tidak menjumpai *error message*. Perhitungan jalur independen dapat dilakukan melalui metrik *Cyclomatic Complexity.* Sebelum menghitung nilai *Cyclomatic Complexity, harus* diterjemahkan desain prosuderal ke grafik alir, kemudian dibuat *flow graphnya*, seperti pada gambar di bawah ini.(Roger S.Pressman, 2002 : 536).

**Gambar 2.2**ContohBagan Alir



**Gambar 2.3**ContohGrafik Alir

Keterangan :

1. *Node* adalah lingkaran yang merepresentasikan satu atau lebih statemen prosedural.
2. *Edge* adalah anak panah pada grafik alir.
3. *Region* adalah area yang membatasi edge dan node.
4. Simpul Predikat adalah simpul atau node yang berisi kondisi yang ditandai dengan dua atau lebih edge yang berasal darinya.

Darigambar *flowgraph* di atas didapat :

*Path* 1 =1– 11

*Path* 2 =1– 2 – 3 – 4 – 5 – 10– 1–11

*Path* 3 =1– 2 – 3 – 6 – 8 – 9 – 10– 1 – 11

*Path* 4 =1– 2 – 3 – 6 – 7 – 9–10–1–11

*Path* 1,2,3,4 yang telah didefinisikan diatas merupakan *basis set*untuk diagram alir.

*Cyclomaticcomplexity* digunakan untuk mencari jumlah *path*dalam satu *flowgraph*.Dapat dipergunakan rumusan sebagaiberikut :

1. Jumlah region grafikalir sesuaidengan*cyclomaticcomplexity*.

2. *Cyclomatixcomplexity*V(G) untukgrafikalir dihitung dengan rumus:

***V(G) =E– N +2*** ………………….(2.3)

Dimana :

E=jumlah*edge*pada grafikalir

N=jumlah *node*pada grafikalir

*Cyclomatixcomplexity*V(G) jugadapatdihitung dengan rumus :

***V(G) =P +1*** …………………. (2.4)

DimanaP =jumlah*predicate node* pada grafikalir

Dari Gambar di atas dapat dihitung*cyclomaticcomplexity*:

1.*Flowgraph*mempunyai 4 region

2.V(G) =11 *edge*– 9 *node* +2 =4

3.V(G) =3 *predicatenode* +1 =4

Jadi*cyclomaticcomplexity*untuk*flowgraph*adalah4.

**2.2.13.2 *BlackBox***

Pengujian *Black-Box* berusaha menemukan kesalahan dalam kategori :

1. Fungsi tidak benar atau hilang.
2. Kesalahan antar muka.
3. Kesalahan pada struktur data (pengaksesan basis data).
4. Kesalahan inisialisasi dan akhir program.
5. Kesalahan performasi.

Pengujian ini berfokus pada persyaratan fungsional perangkat lunak dan merupakan komplemen dari pengujian *White-Box*. Hal ini dapat dicapai melalui :

1. Pengujian *Graph-based*: dimulai dengan membuat grafik sekumpulan node yang mempresentasikan objek (misal *New File*, Layar baru dengan atributnya), link (hubungan antar objek), *node-weight* (misal nilai data tertentu seperti atribut layar, perilaku), dan link-weight (karakteristik suatu link, misal menu select).
2. *Equivalence Partitioning*: membagi domain input untuk pengujian agar diperoleh kelas-kelas kesalahan (misal kelompok data karakter, atau atribut yang lain).
3. Analisis Nilai Batas: pengujian berdasarkan nilai batas domain input.
4. Pengujian Perbandingan: disebut juga pengujian *back-to-back* yang diterapkan pada pada suatu versi perangkat lunak atau perangkat lunak redundan untuk memastikan konsistensinya.
   1. **Perangkat Lunak Pendukung**

Adapun perangkat lunak pendukung yang digunakan oleh penulis dalam membangun sistem ini ada beberapa diantaranya adalah:

**Tabel 2.5**Perangkat Lunak Pendukung

|  |  |  |
| --- | --- | --- |
| **No.** | **Perangkat Lunak Pendukung** | **Kegunaan** |
| 1. | Android Studio 2.1.1 | Bahasa Pemrograman yang digunakan untuk membuat program. |
| 2. | Database MySQL | Sebuah perangkat lunak yang digunakan dalam pengoperasian basis data. |

* 1. **Kerangka Pikir**

1. Bagaimana cara merekayasa sebuah Sistem Iterkonesi Antara Peugas Pelabuahan dengan menggunakan Mobile Sistem?
2. Bagaimana hasil penerapan Androi terhadap Rancang Bangun Aplikasi Android Untuk Interkoneksi Antara Petugas Pelabuhan Dan nahkodah Kapal?

**Masalah**

Petugas Pelabuhan Dan Nahkoda Kapal membutuhkan Suatu Sistem Interkoneksi agar Selalu terhubung Pada Saat barlayar.

**Peluang**

Aplikasi Mobile Untuk Interkoneksi Aktifitas Pelayaran

**Solusi**

1. Sistem Berjalan
2. Sistem Diusulkan

**Analisa Sistem**

1. Desain Model
2. Desain User Interface(Desain Output, Desain Input, Desain Menu Utama)
3. Desain Database
4. Desain Teknologi

**Desain Sistem**

1. Anroid Studio 2.1.1
2. Database MySQL

**Pembangunan Sistem**

1. White Box.
2. Black Box.

**Pengujian**

Kantor KP3 Banggai Laut

**Implementasi**

1. Merancang Bangun Aplikasi Anroid Untuk Interkoneksi Antara Petugas Pelabuahan Dan Nahkodah Kapal.
2. Menerapkan hasil dari Mobile Sistemdalam Melakukan Perancangan Sistem Aplikasi Anroid Untuk Memudahkan Interkonek Antara Petugas Pelabuhan dan Nahkodah Kapal.

**Tujuan**

**Gambar 2.4** Kerangka Pikir