

## **BAB II**

### **LANDASAN TEORI**

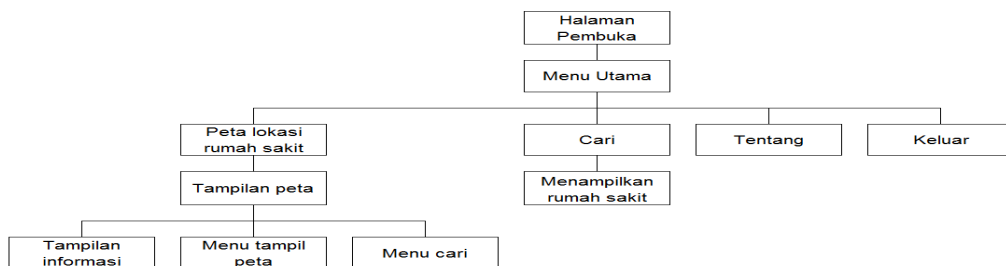
#### **2.1 Tinjauan Pustaka**

Saat ini dengan adanya kemajuan dan perkembangan zaman seperti saat ini pemanfaatan teknologi dibutuhkan untuk dapat memberikan informasi dengan cepat, dimana pemanfaatan teknologi dapat diakses kapan dan dimanapun penggunaannya berada. Perkembangan teknologi sangat berkembang, Salah satunya telepon seluler dimana saat ini tidak hanya dapat digunakan untuk melakukan komunikasi seperti telepon dan SMS (*Short Message Service*) saja tetapi dapat juga digunakan untuk mengakses internet. Dengan fungsi mengakses internet ini penggunaan telepon seluler dapat di kembangkan menjadi sangat luas, dapat digunakan untuk membaca *email*, *browsing*, *chatting*, membaca berita, melihat peta *online* seperti *Google Maps*. Oleh sebab itu, ketepatan dalam memilih jenis rumah sakit sesuai kebutuhan masyarakat sangat penting, terutama jika dalam keadaan darurat.

Teknologi informasi yang semakin lama semakin maju dan berkembang setiap saat, mempunyai peranan yang sangat penting dalam segala aspek kehidupan. Salah satu aspek teknologi yang sedang berkembang adalah teknologi *mobile* pada perangkat telepon pintar (*Smartphone*). Salah satu teknologi *Smartphone* yang sedang ramai diperbincangkan adalah Android. Android merupakan sebuah sistem operasi untuk perangkat *Smartphone*. Sistem operasi ini dibuat berbasis *kernel* Linux yang telah dimodifikasi sehingga cocok untuk digunakan di ponsel atau perangkat genggam lain. Saat ini semakin banyak orang yang menggunakan sistem operasi Android, hampir berbagai kalangan masyarakat menggunakannya.

## 2.2 Kerangka Pemikiran

Rumah sakit merupakan salah satu dari sarana kesehatan tempat menyelenggarakan upaya kesehatan, tetapi tidak semua rumah sakit diketahui oleh masyarakat baik yang berasal dari Surakarta maupun yang berasal dari luar daerah Surakarta karena kurangnya informasi tentang rumah sakit di Surakarta. Rumah sakit adalah institusi pelayanan kesehatan yang menyediakan pelayanan rawat inap, rawat jalan, dan gawat darurat, serta dibantu oleh dokter, perawat dan tenaga ahli kesehatan lainnya. Sebagian besar masyarakat mengunjungi rumah sakit untuk melakukan pemeriksaan rutin atau hanya untuk chek up. Daerah se-Soloraya, khususnya kota Surakarta merupakan salah satu daerah yang memiliki cukup banyak rumah sakit. Tidak semua rumah sakit diketahui oleh masyarakat apalagi oleh pendatang baru. Untuk mengatasi masalah tersebut dibutuhkan layanan lokasi guna memberikan informasi fasilitas layanan kesehatan yang bisa diakses dengan mudah kapan pun dan dimanapun oleh pengguna. Atas dasar tersebut penelitian ini bertujuan untuk mengembangkan Go Hospital berbasis Android dengan menggunakan teknologi *Location Based Service* (LBS) dan petunjuk rute untuk menuju ke rumah sakit memanfaatkan Google Map API yang dapat diakses secara *online*. Tahapan penelitian meliputi analisis kebutuhan, perancangan sistem, implementasi dan pengujian. Aplikasi Go Hospital diuji menggunakan metode *Black Box* dengan hasil pengujian menyatakan bahwa fungsi-fungsi yang ada dalam Go Hospital berjalan dengan lancar. Dengan adanya Go Hospital ini informasi kesehatan khususnya di kota Surakarta dapat terpenuhi. Dengan demikian, diperlukan suatu aplikasi yang dapat menyediakan informasi rumah sakit di suatu wilayah tertentu, dalam konteks ini rumah sakit yang ada Kota Surakarta. (Gambar 2.1).



Gambar 2.1 Kerangka Aplikasi Go Hospital

## 2.3 Landasan Teori

### 2.3.1 Informasi

Informasi merupakan salah satu hal yang sangat dibutuhkan dalam suatu organisasi terutama kebutuhan bagi manajemen di dalam pengambilan suatu keputusan. Selain itu informasi dapat diartikan juga sebagai data yang telah yang telah mengalami suatu proses yang menjadi berguna bagi yang menerimanya. Adapun data dapat diartikan sebagai kumpulan suatu kejadian yang diangkat dari suatu kenyataan. Maka informasi dapat diartikan sebagai suatu hasil pengolahan data yang dapat digunakan sebagai dasar pengambilan suatu keputusan (Witarto, 2004).

### 2.3.2 Aplikasi Perangkat Lunak

Perangkat lunak aplikasi yaitu perangkat lunak yang digunakan untuk membantu pemakai komputer untuk melaksanakan pekerjaannya. Jika ingin mengembangkan program aplikasi sendiri, maka untuk menulis program aplikasi tersebut, dibutuhkan suatu bahasa pemrograman, yaitu *Language software* yang dapat berbentuk *assembler*, *compiler* ataupun *interpreter*. Jadi *Language software* merupakan bahasanya dan program yang ditulis merupakan program aplikasinya.

Perangkat lunak menjadi bagian sistem komputer yang tidak berwujud. Perangkat lunak adalah istilah umum untuk data yang diformat dan disimpan secara digital, termasuk program komputer, dokumentasi, dan berbagai informasi yang bisa dibaca dan ditulis oleh komputer. Istilah ini menonjolkan perbedaan dengan perangkat keras komputer. Perkembangan perangkat keras (*hardware*) pada akhirnya akan berimbas pula kepada perkembangan dari perangkat lunak, ini dibuktikan dengan semakin majunya bidang teknologi informasi baik yang berupa perangkat keras maupun perangkat lunak.

3. Memenuhi sejumlah kriteria *implisit*, yaitu jika salah satu kriteria *implisit* tersebut tidak dapat dipenuhi, maka perangkat lunak yang bersangkutan tidak memiliki kualitas yang baik.

Selanjutnya perangkat lunak tersebut dibagi ke dalam 4 kelompok, yaitu sebagai berikut (Sutanta, 2005) :

1. Sistem Operasi, adalah perangkat lunak yang digunakan untuk mengendalikan *resources* selama proses berlangsung.
2. Bahasa Pemrograman, merupakan perangkat lunak yang digunakan sebagai alat untuk pengembangan program aplikasi.
3. Bahasa *Query*, merupakan perangkat lunak bahasa tingkat sangat tinggi yang dapat digunakan untuk menampilkan informasi–informasi yang diinginkan hanya dengan menuliskan sedikit perintah saja.
4. Aplikasi, merupakan perangkat lunak yang dikembangkan untuk digunakan pada aplikasi tertentu.

#### 2.3.2.1 Android

Saat ini kemajuan teknologi informasi telah mengubah cara kerja seseorang dalam mendapatkan informasi, hal ini memungkinkan orang mengontrol suatu informasi tanpa tergantung pada lokasi yang disebut metode kerja bergerak (*mobile*). Sistem operasi untuk aplikasi bergerak yang mengalami perkembangan cukup pesat yaitu Android. Android merupakan sistem operasi bergerak (*mobile*) yang menggunakan versi modifikasi dari *kernel* Linux. Sistem ini memiliki berbagai keunggulan sebagai *software* berbasis kode komputer yang bisa didistribusikan secara terbuka (*open source*), sehingga *programmer* bisa membuat aplikasi baru di dalamnya. Adanya AndroidMarket dalam sistem operasi Android, menghadirkan ribuan aplikasi baik yang gratis maupun yang berbayar. Dari arsitektur sistem, Android merupakan sekumpulan *framework* dan *virtual machine* yang berjalan di atas *kernel* Linux. *Virtual machine* Android bernama *Dalvik Virtual machine* (DVM), *engine* ini berfungsi untuk menginterpretasikan dan menghubungkan seluruh kode mesin yang digunakan oleh aplikasi dengan *kernel* Linux.

Dengan antarmuka pengguna berdasarkan manipulasi langsung, Android dirancang terutama untuk perangkat *mobile touchscreen* seperti *Smartphone* dan komputer tablet, dengan antarmuka pengguna khusus untuk mobil (*Android Auto*), televisi (*TV Android*), hingga pada jam tangan (*Android Wear*). OS ini

menggunakan *input* sentuh yang mudah sesuai dengan tindakan di dunia nyata seperti menyapu, menekan, mencolek, dan mencubit untuk memanipulasi objek di layar dan keyboard *virtual*. Meskipun dirancang untuk *input touchscreen*, juga telah digunakan dalam konsol game, kamera digital, PC biasa, dan elektronik lainnya.

Android adalah sebuah sistem operasi (OS)-bersifat *open source* (terbuka) yang dimiliki oleh Google Inc. Pada awal peluncuran, Android hanya digunakan untuk perangkat *mobile* pada telepon seluler. Entah dengan maksud apa, sistem operasi Android yang dimulai dari versi 1.5 selalu memakai nama-nama makanan/kue sebagai nama alias. Misalnya Android 3.0 (Honeycomb) diluncurkan, sistem operasi Android resmi digunakan dalam komputer tablet. OS tersebut hingga versi Android 4.1 Jelly Bean dianggap paling cocok diaplikasikan pada komputer tersebut. Beberapa perkembangan Android adalah sebagai berikut.

1. Android versi 1.1

Android versi ini dirilis pada tanggal 9 Maret 2009. Muncul pertama kali dengan fitur pembaruan terhadap estetika/*behaviour* pada aplikasi, jam alarm, *Voice search* (pencarian suara), pengiriman pesan dengan Gmail, dan pemberitahuan (notifikasi) *email*.

2. Android versi 1.5 (Cupcake)

Cupcake atau dalam terjemahan bahasa Inggris berarti kue cawan, adalah nama alias dari sistem operasi Android versi 1.5. Sistem itu secara resmi diperkenalkan pada pertengahan Mei 2009. Di dalam versi ini, ada beberapa fitur yang diperbarui, dan juga ada penambahan beberapa fitur yaitu untuk merekam dan menonton video pada modus kamera, meng*upload* video ke YouTube dan gambar ke aplikasi Picasa secara langsung, kemampuan dukungan *Bluetooth* A2DP, animasi pada layar, serta keyboard pada layar.

3. Android versi 1.6 (Donut)

Donut atau yang kita kenal dengan kue berlubang dipakai sebagai nama alias dari versi Android 1.6. Versi ini dirilis September 2009, dilengkapi dengan kemampuan proses pencarian, penggunaan baterai indikator, dan *Virtual Private Network* (VPN) yang lebih baik dibanding sebelumnya.

4. Android versi 2.0/2.1 (Eclair)

Pada 3 Desember 2009 lagi-lagi sistem operasi Android merilis versi baru yaitu versi 2.0/2.1 yang diberi nama alias Eclair. Eclair adalah sebuah kue sus yang berbentuk panjang dengan topping cokelat di atasnya. Beberapa perubahan pada versi ini yaitu adanya optimalisasi pada *hardware*, peningkatan Google *Maps* 3.1.2, perubahan *user interface* (UI) dengan *browser* baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3.2 MP, digital zoom, serta *Bluetooth* 2.1.

5. Android versi 2.2 (Froyo)

Froyo (Frozen yoghurt) adalah yoghurt (susu yang dibuat melalui fermentasi bakteri) yang dibekukan mirip seperti *ice cream*. Froyo dipakai sebagai nama alias dari sistem operasi Android versi 2.3. Versi ini meluncur pada Mei 2010 dengan peningkatan pada *performance* dan memori, integrasi dengan Google Chrome, USB *tethering*, WI-FI *Hotspot*, *Quick switching*, *Voice dialing*, serta mendukung Adobe *Flash* 10.1.

6. Android versi 2.3 (Gingerbread)

Gingerbread adalah kue yang terbuat dari jahe, biasanya berbentuk boneka dan sering disajikan sebagai teman untuk minum kopi. Gingerbread dipakai sebagai nama alias dari sistem operasi Android versi 2.3. Versi 2.3 dari Android ini merupakan versi terakhir dari sistem operasi Android, khususnya untuk perangkat telepon seluler.

7. Android versi 3.0 (Honeycomb)

Honeycomb atau sarang madu adalah bagian dari hasil produksi lebah. Honeycomb dipakai sebagai nama alias dari sistem Android versi 3.0. Melihat perkembangannya, versi Android 3.0 ini lebih ditujukan untuk mengoptimalkan sistem operasi tablet, atau lebih tepatnya, Android versi 3.0 adalah sistem operasi Android untuk tablet PC.

8. Android versi 4.0 (*Ice cream* Sandwich)

*Ice cream* Sandwich, tentu saja kita tahu ini adalah minuman atau tepatnya makanan yang sangat disukai terutama oleh anak kecil. *Ice cream* Sandwich dipakai sebagai nama alias dari Android versi 4.0. Platform terbuka dari

Google ini ditujukan untuk perangkat seluler dan tablet. Android *Ice cream Sandwich* dilengkapi dengan fitur *Near Field Communication* (NFC) yang sudah dikenal dengan sistem pembayaran masa depan.

9. Android versi 4.1 (Jelly Bean)

Setelah sukses meluncurkan OS Android *Ice cream Sandwich*, Android dengan bangga meluncurkan OS Jelly Bean sebagai OS terbarunya. Terdapat tujuh fitur utama yang dapat dirasakan saat menggunakan OS Jelly Bean, yaitu Photo Sphere dapat membuat *multishoot* dengan *image* yang besar, seperti pada Google Street View.

10. Android 4.4 (KitKat)

Android 4.4 KitKat pertama kali dirilis pada Oktober 2013 dan menggunakan nama *brand* cemilan terkenal. Versi Android ini bisa dikatakan menjadi favorit dari hampir keseluruhan *user smartphone* di dunia. Hal ini dikarenakan Android 4.4 KitKat dapat memberikan optimalisasi yang baik, termasuk pada perangkat *smartphone* yang memiliki spesifikasi minimalis.

11. Android 5.0 & 5.1 (Lollipop)

Mulai beberapa versi ke belakang, Android dan Google pun mulai secara rutin memperbarui sistem operasi mereka dalam selang waktu setahun, yaitu Android 5.0 - 5.1 Lollipop yang dirilis dan diresmikan pada Juni 2014. Bisa dibayangkan Android 5.0 - 5.1 Lollipop menjadi *pionir* dibuatnya *smartphone flagship* dengan spesifikasi cukup mumpuni. Versi Android ini sudah mendukung arsitektur 64-bit yang sudah memberi peluang pada penggunaan RAM di atas 3GB. Salah satunya ASUS Zenfone 2 yang sudah mengukung RAM 4GB saat itu.

12. Android 6.0 (Marshmallow)

Android 6.0 Marshmallow menjadi suksesor dari versi Android sebelumnya dan pertama kali diperkenalkan pada Mei 2015 serta mulai dirilis pada Oktober 2015 silam. Sistem operasi ini secara jelas memberikan peningkatan pada sistem keamanan dengan dihadirkannya sensor *fingerprint* sebagai sistem keamanan *biometric* yang digunakan. Selain digunakan untuk

mengunci layar, sensor *fingerprint* ini dapat digunakan untuk otentikasi Google Play Store dan pembelian dengan menggunakan Android Pay.

#### 13. Android 7.0 & 7.1 (Nougat)

Android 7.0 - 7.1 Nougat diperkenalkan pada Juni 2016 dengan menampilkan ikon robot Android dengan batangan Nougat. Sistem operasi Android 7.0 - 7.1 Nougat mengalami perubahan dari segi tampilan antarmuka. Selain itu ada juga fitur *splitscreen* untuk membagi tampilan layar untuk dua aplikasi sekaligus. Untuk saat ini, sistem operasi Android ini masih digunakan beberapa *smartphone* yang baru dirilis belakangan ini.

#### 14. Android 8.0 & 8.1 (Oreo)

Android 8.0 - 8.1 Oreo menjadi sistem operasi Android yang dirilis secara stabil mulai Agustus 2017 serta mengalami pembaruan lewat versi Android 8.1 Oreo. Android 8.0 - 8.1 Oreo menjadi versi Android kedua yang menggunakan makanan manis dari nama *brand* setelah Android 4.4 KitKat. Selain itu ada juga *Project Treble* yang memberi peluang pada *user* mendapat pembaruan lebih cepat. Sistem operasi ini juga menawarkan pengalaman *multitasking* yang makin mumpuni dibanding versi sebelumnya.

#### 15. Android 9.0 (Pie)

Android 9.0 Pie yang secara resmi diperkenalkan pada Agustus 2018. Sistem operasi Android ini memberi banyak perubahan, terutama untuk HP dengan desain baru. Misalnya Pie memberikan navigasi berupa *gesture* yang menggantikan tombol fisik *Home*, *Back* dan *Recent Apps*. Fitur lainnya adalah sistem notifikasi, pengatur kecerahan hingga sistem *screenshoot* terbaru yang lebih mudah.

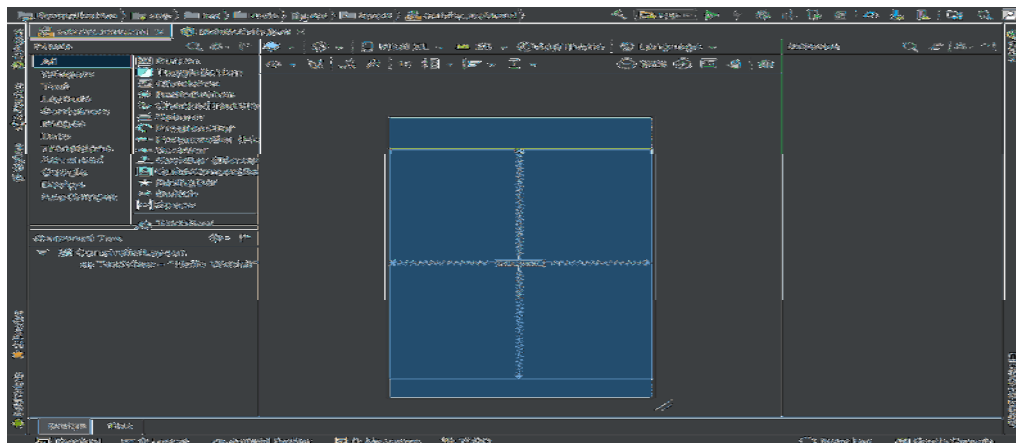
#### 2.3.2.2. Android Studio

Android Studio adalah *Integrated Development Environment* (IDE) untuk mengembangkan pada *platform* Android. Diumumkan pada 16 Mei 2013 di Google I/O konferensi dengan Product Manager Google, Ellie Powers. Android Studio tersedia secara bebas di bawah Apache License 2.0. Android Studio berada di awal tahap *preview* mulai dari versi 0.1 Mei 2013, kemudian memasuki tahap



beta mulai dari versi 0.8 yang dirilis pada bulan Juni 2014. Sedangkan versi pertama yang stabil dirilis pada bulan Desember 2014, mulai dari versi 1.0. Android Studio berdasarkan JetBrains IntelliJ IDEA *software* telah dirancang khusus untuk pengembangan Android dan telah tersedia untuk diunduh pada Windows, Mac OS X dan Linux. Sedangkan menurut Jubilee Enterprise (2015:10) Android Studio merupakan IDE (*Integrated Development Environment*) *official* berbasis IntelliJ IDEA untuk membuat aplikasi Android. Langkah-langkah untuk melakukan konfigurasi Android Studio adalah sebagai berikut.

1. Jalankan Android Studio.
2. Tunggu sampai proses *download* selesai.
3. Setelah proses *download* selesai, klik *Finish*.
4. Jendela Welcome to Android Studio akan di tampilkan.
5. Pilih *Configure–Project Defaults–Project Structure*.
6. Pilih lokasi SDK kemudian klik tombol OK.
7. Setelah memilih SDK, Android Studio sudah dapat digunakan. (gambar 2.1)



Gambar 2.1. Antarmuka Android Studio

### 2.3.2.3. Android *software* Development Kit (SDK)

Android SDK adalah *tools* API (*Application Programming Interface*) yang diperlukan untuk memulai pengembangan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java. Pada Android SDK ini terdiri dari *debugger*, *libraries*, *handset emulator*, dokumentasi dan *tutorial*. *Software* Development Kit memungkinkan pengembang untuk membuat aplikasi untuk

*platform* Android. SDK Android mencakup proyek sampel dengan kode sumber, perangkat pengembangan, *emulator* dan perpustakaan yang diperlukan untuk membangun aplikasi Android. Aplikasi yang ditulis menggunakan bahasa pemrograman Java dan berjalan di *Dalvik*, mesin *virtual* yang dirancang khusus untuk penggunaan *embedded* yang berjalan di atas *kernel* Linux (Safaat, 2011).

#### 2.3.2.4. Android *Virtual Device* (AVD)

Android *Virtual Device* (AVD) yang merupakan *emulator* untuk menjalankan program aplikasi Android yang kita buat (Safaat, 2011). AVD ini yang selanjutnya digunakan sebagai tempat untuk *Test* dan menjalankan aplikasi Android yang telah dibuat. Dengan AVD ini, *developer* bisa mengembangkan dan mencoba aplikasi Android tanpa harus menggunakan perangkat Android yang sebenarnya. Sebelum menggunakan AVD harus menentukan karakteristiknya, misalkan dalam menentukan versi Android, jenis dan ukuran layar dan besarnya memori. AVD bisa dibuat sebanyak yang kita inginkan.

#### 2.3.2.5 Java

Java merupakan salah satu bahasa pemrograman yang bersifat *multiplatform* dengan slogan dari para pengembangnya adalah '*Write once run everywhere*' sehingga aplikasi yang dikembangkan menggunakan bahasa Java akan dapat dijalankan pada berbagai macam *platform* atau sistem operasi. Hal ini menjadi salah satu solusi dari berbagai macam bahasa pemrograman yang ada di dunia IT saat ini, yang biasanya hanya dapat dijalankan pada satu sistem operasi saja dan tidak dapat dijalankan di sistem operasi yang lain. Bahasa pemrograman Java pertama kali dikembangkan oleh James Gosling di Sun Microsystems, yang sekarang menjadi bagian dari Oracle Corporation. Dirilis pada tahun 1995 sebagai bagian dari *platform Java* oleh *Sun Microsystems*. Bahasa ini telah mengembangkan banyak sintaks dari C dan C ++. Aplikasi Java biasanya dikompilasi ke *Byte Code* (kelas file) yang dapat berjalan pada *Java Virtual machine* (JVM). Pada saat ini, Java dinyatakan sebagai salah satu bahasa pemrograman paling populer digunakan dengan memiliki sekitar 10 juta pengguna di seluruh dunia. Java dapat berjalan pada banyak sistem operasi yang berbeda karena menggunakan membuat *platform Java* yang independen. Java

melakukan hal ini dengan membuat Java *compiler* merubah kode menjadi Java *byteCode* sebagai pengganti kode mesin. Ini berarti bahwa ketika program ini dijalankan, Java *Virtual machine* menafsirkan *byteCode* dan diterjemahkan ke dalam kode mesin.

#### 2.3.2.6 MySQL

MySQL merupakan *Database server* open source yang cukup populer keberadaannya. Dengan berbagai keunggulan yang dimiliki, membuat *software Database* ini banyak digunakan oleh para praktisi untuk membangun suatu *project*. Adanya fasilitas API (*Application Programming Interface*) yang dimiliki oleh MySQL, memungkinkan bermacam-macam aplikasi komputer yang ditulis dengan berbagai bahasa pemrograman dapat mengakses basis data MySQL.

#### 2.3.3 Rumah Sakit

Rumah sakit adalah tempat dimana orang sakit mencari dan menerima pelayanan kedokteran serta tempat dimana pendidikan klinik untuk mahasiswa kedokteran, perawat dan berbagai tenaga profesi kesehatan lainnya yang diselenggarakan. Menurut UU No. 44 tahun 2009 Rumah sakit adalah institusi pelayanan kesehatan yang menyelenggarakan pelayanan kesehatan perorangan secara paripurna, yang menyediakan pelayanan rawat inap, rawat jalan dan gawat darurat.

#### 2.3.4 USDP (*Unified software Development Process*)

Metode pengembangan perangkat lunak yang digunakan dalam pembangunan aplikasi *mobile* informasi rumah sakit di Kota Surakarta yaitu *Unified software Development Process* (USDP). Metode USDP ini merupakan proses pengembangan perangkat lunak yang berkelanjutan, dimana masing-masing bagian dilakukan secara *iterative* (Nugroho, 2010). USDP menggunakan *Unified Modeling Language* (UML) sebagai kakas utamanya yang disesuaikan dengan fungsinya masing-masing. USDP memiliki lima model yang digunakan yaitu model analisis, model perancangan, model implementasi, model *Deployment* dan model pengujian. Adapun kegunaan dari model-model tersebut adalah sebagai berikut.

1. Model analisis (*Analysis*)

Model analisis memiliki dua kegunaan, yaitu memperhalus dan merinci definisi-definisi masing-masing *use case*. Model analisis perangkat lunak sesungguhnya merupakan spesifikasi rinci dari kebutuhan-kebutuhan pengguna dan bekerja sebagai langkah pertama saat mengembangkan model perangkat lunak. Model analisis dapat digunakan untuk dapat lebih memahami *use case* yang pada prinsipnya menggambarkan kolaborasi yang terjadi antara pengklasifikasian bersifat konseptual (sebagai lawan dari pengklasifikasian perancangan yang akan diimplementasikan).

2. Model Perancangan (*Design*)

Model perancangan mendefinisikan struktur statis sistem seperti subsistem, kelas-kelas, antarmuka-antarmuka dan hubungan antar pengklasifikasi, serta kolaborasi-kolaborasi yang merealisasikan suatu *use case* tertentu dalam kerangka sistem perangkat lunak yang sedang dikembangkan. Realisasi *use case* perancangan sesungguhnya memiliki deskripsi aliran atau even-even (*flow-of-flow* seiring orang menyebutnya sebagai lintasan atau skenario), juga memiliki *diagram* kelas yang memperlihatkan kelas-kelas perancangan yang berpartisipasi dan juga memiliki *diagram-diagram* interaksi yang memperlihatkan suatu aliran atau skenario dalam *terminology* interaksi yang terjadi diantara objek-objek perancangan. Jika diperlukan, *diagram-diagram* realisasi perancangan *use case* dapat dibuat untuk memperlihatkan keberadaan subsistem-subsistem atau antarmuka yang terlibat dalam realisasi *use case*.

3. Model *Deployment*

*Deployment diagram* yang menggambarkan detail bagaimana komponen di-*deploy* dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, *server* atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat fisik. Sebuah *node* adalah *server*, *workstation*, atau piranti keras lain yang digunakan untuk men-*deploy* komponen dalam lingkungan sebenarnya.

#### 4. Model Implementasi

Tahapan ini merupakan tahapan lanjutan dari desain aplikasi sistem, yaitu menafsirkan atau menterjemahkan desain aplikasi sistem ke dalam bahasa pemrograman yang dapat di mengerti oleh pemakai dan sistem komputer. Dalam tahapan ini dijelaskan secara detail penggunaan sistem dari proses memperbaharui informasi yang ada hingga proses *preview*.

#### 5. Model Pengujian (*Testing*)

Model pengujian merupakan model akhir dari pengembangan USDP. Model pengujian mendeskripsikan kasus-kasus dan prosedur-prosedur pengujian yang tujuannya adalah melakukan verifikasi perangkat lunak yang dihasilkan dengan cara melihat dan memastikan apakah masing-masing *use case* telah diimplementasikan dengan cara sesuai dengan fungsionalitas utamanya yang tercakup didalamnya.

#### 2.3.5. *Unified Modeling Language* (UML)

UML (*Unified Modelling Language*) sebagai bahasa yang digunakan untuk menentukan visualisasi, membangun, mendokumentasikan, dan sistem informasi. UML (*Unified Modelling Language*) juga berarti bahasa pemodelan standar yang memiliki sintaks dan semantik. Ketika membuat model menggunakan konsep UML terdapat aturan-aturan yang harus diikuti. Elemen pada model-model yang dibuat berhubungan satu dengan lainnya dan harus mengikuti standar yang ada. UML bukan hanya sekedar *diagram*, tetapi juga menceritakan konteksnya.

UML menunjukkan bagaimana mengekspresikan sebuah desain berorientasi objek. 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. Konsep dasar UML terdiri dari *structural classification*, *dynamic behavior*, dan model *management*, dapat dipahami dengan mudah apabila melihat dari *diagram*. UML mendefinisikan *diagram–diagram* yang diantaranya *Use case diagram*, *activity diagram*, dan *Sequence diagram*. Pemodelan (*Modelling*) adalah proses merancang piranti lunak sebelum melakukan pengkodean (*coding*). Model piranti lunak dapat dianalogikan seperti pembuatan *blueprint* pada pembangunan

gedung. Membuat model dari sebuah sistem yang kompleks sangatlah penting karena kita tidak dapat memahami sistem semacam itu secara menyeluruh. Semakin kompleks sebuah sistem, semakin penting pula penggunaan teknik pemodelan yang baik. Kesuksesan suatu pemodelan piranti lunak ditentukan oleh tiga unsur. Ketiga unsur yang dimaksud adalah metode pemodelan (*notation*), proses (*process*) dan *tools* yang digunakan (Dharwiyanti. 2003).

Namun demikian, model-model tersebut dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis *diagram* itu antara lain sebagai berikut.

1. *Diagram* Kelas. *Diagram* ini bersifat statis dengan memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. *Diagram* ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula *diagram* kelas memuat kelas-kelas aktif.
2. *Diagram* Paket (*Package Diagram*). *Diagram* ini bersifat statis dengan memperlihatkan kumpulan kelas-kelas, merupakan bagian dari *diagram* komponen.
3. *Diagram Use-Case*. *Diagram* ini bersifat statis dengan memperlihatkan himpunan *Use-Case* dan aktor-aktor (suatu jenis khusus dari kelas). *Diagram* ini sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.
4. *Diagram* interaksi dan *Sequence* (urutan). *Diagram* urutan bersifat dinamis sebagai jenis *diagram* interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.
5. *Diagram* Komunikasi (*Communication Diagram*). *Diagram* ini bersifat dinamis sebagai pengganti *diagram* kolaborasi UML 1.4 yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan.
6. *Diagram Statechart* (*Statechart Diagram*). *Diagram* ini bersifat dinamis dengan memperlihatkan keadaan-keadaan pada sistem, memuat status (*state*), transisi, kejadian serta aktifitas. *Diagram* ini terutama penting untuk

memperlihatkan sifat dinamis dari antarmuka (*interface*), kelas, kolaborasi dan terutama penting pada permodelan sistem-sistem yang reaktif

7. *Diagram* Aktivitas (*Activity diagram*). *Diagram* ini bersifat dinamis sebagai tipe khusus dari *diagram* status yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dalam suatu sistem. *Diagram* ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek.
8. *Diagram* Komponen (*Component Diagram*). *Diagram* komponen ini bersifat statis dengan memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya. *Diagram* ini berhubungan dengan *diagram* kelas dimana komponen secara tipikal dipetakan ke dalam satu atau lebih kelas-kelas, Antarmuka-Antarmuka serta kolaborasi-kolaborasi.
9. *Diagram* Deployment (*Deployment diagram*). *Diagram* ini bersifat statis dengan memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*). Memuat simpul-simpul beserta komponen-komponen yang ada didalamnya. *Diagram* Deployment berhubungan erat dengan *diagram* komponen dimana *diagram* ini memuat satu atau lebih komponen-komponen. *Diagram* ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing*).

## 2.4. Pengujian Sistem

Dalam pengujian sistem Go Hospital berbasis Android dilakukan proses pengujian menggunakan *Black Box Testing* (pengujian kotak hitam). Pengujian perangkat lunak metode *Black Box* merupakan pendekatan pengujian dimana data pengujian berasal dari persyaratan fungsional yang ditentukan tanpa memperhatikan struktur program akhir. *Black Box Testing* adalah metode pengujian perangkat lunak (*software*) yang tes fungsionalitas dari aplikasi yang bertentangan dengan struktur internal atau kerja. Pengujian *Black Box* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya fungsi-fungsi yang salah atau hilang, kesalahan *interface*, kesalahan dalam struktur data atau akses *Database* eksternal, kesalahan performa, kesalahan inisialisasi dan terminasi.

Pengetahuan khusus dari kode aplikasi / struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Uji kasus dibangun di sekitar spesifikasi dan persyaratan, yaitu aplikasi apa yang seharusnya dilakukan. Menggunakan deskripsi eksternal perangkat lunak termasuk spesifikasi, persyaratan, dan desain untuk menurunkan uji kasus. Tes ini dapat menjadi fungsional atau non-fungsional, meskipun biasanya fungsional. Perancang uji memilih *input* yang *valid* atau tidak *valid* dengan menentukan *output* yang benar. Tidak ada pengetahuan tentang struktur internal benda uji itu.