

## **BAB II**

### **TINJAUAN PUSTAKA DAN LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Analisis kebutuhan merupakan langkah awal untuk menentukan perangkat lunak yang dihasilkan. Analisis kebutuhan menjadi faktor yang sangat menentukan keberhasilan perangkat lunak itu sendiri. Jika pada tahap analisis kebutuhan terjadi kesalahan, maka perangkat lunak yang dihasilkan pun menjadi tidak berguna.

Analisis kebutuhan yang baik belum tentu menghasilkan perangkat lunak yang baik pula. Namun jika pada tahap ini sudah terjadi kesalahan, maka sudah pasti akan menghasilkan perangkat lunak yang gagal.

Adapun faktor utama yang harus dipenuhi dalam membuat analisis kebutuhan adalah sebagai berikut:

1. Komunikasi yang baik

Membangun komunikasi yang baik dengan pengguna merupakan hal yang sangat diperlukan guna hasil dari perangkat lunak yang dikerjakan sesuai dengan apa yang diharapkan. Hal ini juga menjadi bantuan yang sangat penting guna menjadikan komunikasi yang terbuka dan lancar.

2. Mengetahui kebutuhan

Dalam hal ini penulis harus mengetahui tentang apa yang akan dikerjakan oleh programmer. Kebutuhan dan batasan yang diinginkan pengguna harus ditentukan dan dinyatakan secara tegas.

### 3. Terbuka dengan langkah yang dilakukan

Bersikap terbuka selama proses pembuatan perangkat lunak akan sangat membantu. Hal ini sesuai dengan pilihan yang diminta oleh pengguna. Antara programmer dengan pengguna akan mudah didalam mengambil sebuah keputusan.

### 4. Gunakanlah istilah yang sederhana

Jangan gunakan istilah yang tidak dimengerti oleh pengguna terutama dalam proses analisis kebutuhan. Hal ini akan mengakibatkan komunikasi akan lebih sulit dan tentunya berdampak pada hasil perangkat lunak yang dibuat.

## **2.2 Landasan Teori**

Untuk mendukung pembuatan laporan ini, maka perlu dikemukakan hal-hal atau teori-teori yang berkaitan dengan permasalahan dan ruang lingkup pembahasan sebagai landasan dalam pembuatan laporan ini.

## **2.3 Zakat**

### **2.3.1 Pengertian Zakat**

Penulis mengenal zakat sebagai salah satu dari lima rukun Islam yang di dalam Al-Quran seringkali dikaitkan dengan shalat. Zakat berasal dari bentuk kata yang berarti suci, baik, berkah, tumbuh, dan berkembang. Menurut terminologi syariat, zakat adalah nama bagi sejumlah harta tertentu yang telah mencapai syarat tertentu yang diwajibkan oleh Allah untuk dikeluarkan dan diberikan kepada yang

berhak menerimanya dengan persyaratan tertentu pula. Kaitan antara makna secara bahasa dan istilah ini berkaitan erat sekali, yaitu bahwa setiap harta yang sudah dikeluarkan zakatnya akan menjadi suci, bersih, baik, berkah, tumbuh dan berkembang. Seperti yang tertulis dalam surat At-Taubah: (103) sebagai berikut:

"Ambillah zakat dari sebagian harta mereka, dengan zakat itu kamu membersihkan dan mensuci-kan mereka."(At-Taubah: 103).

Pada dasarnya ada 3 macam zakat, yaitu zakat maal atau zakat atas harta kekayaan dan zakat fitri yaitu zakat untuk membersihkan diri yang dibayarkan pada bulan Ramadhan menjelang Idul Fitri dan zakat profesi yaitu dimana pendapatan seseorang akan dinilai melalui nishabnya untuk dibayarkan.

### **2.3.2 Perhitungan Zakat**

Penghitungan zakat disesuaikan dengan menggunakan mazhab Syafi'i. Adapun sebelum memulai pembahasan mendalam dan terperinci mengenai setiap kategori aset wajib zakat.

Perhitungan zakat akan dijelaskan sebagai berikut ;

#### **a. Zakat Fitrah**

Kadar zakat fitrah yang harus dibayar adalah satu sha' dari makanan pokok Negara se tempat, dan satu sha' untuk ukuran sekarang kira-kira 2,176 kg (ketentuan ini sesuai makanan pokok gandum) dibulatkan menjadi 2,5 kg atau Kadar nilai zakat disesuaikan dengan harga makanan pokok masing-masing negara.

## **b. Zakat Mal**

adalah zakat yang dikenakan atas harta yang dimiliki oleh individu dengan syarat-syarat dan ketentuan-ketentuan yang telah ditetapkan secara syarak.

### **Zakat Emas Dan Perak**

Nishab emas = 20 Mithqal = 85 gr emas murni

Nishab perak = 200 dirham = 643 gr perak Syarat:

berlakunya waktu 1 tahun dalam keadaan nishab.

Jumlah zakatnya =  $2,5\% \times (\text{jumlah emas/perak})$

**c. Zakat Profesi** adalah zakat yang dikeluarkan dari penghasilan profesi bila telah mencapai nishab, sangat dianjurkan untuk menghitung zakat dari pendapatan kasar (brutto), untuk lebih menjaga kehati-hatian. Nisab sebesar 5 wasaq / 652,8 kg gabah setara 520 kg beras. Besar zakat profesi yaitu 2,5 %. Terdapat 2 kaidah dalam menghitung zakat profesi, yaitu:

#### 1. Menghitung dari pendapatan kasar (brutto)

Besar Zakat yang dikeluarkan = Pendapatan total (keseluruhan) x 2,5 %

#### 2. Menghitung dari pendapatan bersih (netto)

a. Pendapatan wajib zakat = Pendapatan total – Pengeluaran perbulan\*

b. Besar zakat yang harus dibayarkan = Pendapatan wajib zakat x 2,5 %

Keterangan :

\* Pengeluaran per bulan adalah pengeluaran kebutuhan primer (sandang, pangan, papan )

\* Pengeluaran perbulan termasuk : Pengeluaran diri, istri, 3 anak, orang tua dan Cicilan Rumah. Bila dia seorang istri, maka kebutuhan diri, 3 anak dan cicilan Rumah tidak termasuk dalam pengeluaran perbulan.

### **2.3.3 Aplikasi Hitung Zakat**

Aplikasi Hitung Zakat merupakan suatu alat yang mampu menghitung zakat berdasarkan jenisnya sesuai dengan mazhab yang dipakai yaitu mazhab Syafi'i. Perangkat lunak ini dibuat berdasarkan perhitungan yang didasarkan pada Al-Quran dan Sunnah.

Dilain pihak, Google merilis kode-kode Android dibawah lisensi Apache, sebuah lisensi perangkat lunak dan open platform perangkat seluler.

Pada masa saat ini sebagian besar vendor-vendor smartphone sudah memproduksi smartphone berbasis android, vendor-vendor itu antara lain HTC, Motorola, Samsung, LG, HKC, Sony Ericsson dan masih banyak lagi Vendor smartphone di dunia yang memproduksi android. Hal ini, karena android itu adalah sistem operasi yang open source sehingga bebas didistribusikan dan dipakai oleh vendor manapun.

Tidak hanya menjadi sistem operasi di smartphone, saat ini Android menjadi pesaing utaa dari Apple pada sistem operasi Tablet PC. Android itu sendiri adalah platform yang sangat lengkap baik itu sistem operasinya, Aplikasi dan Tool Pengembangan, serta dukungan yang sangat tinggi dari komunitas Open Source dunia sehingga android terus berkembang pesat.

## 1.4 Android

### 2.4.1 Pengertian Android

Menurut Nazridin Safaat.H (2012), Android adalah kumpulan perangkat lunak yang ditujukan bagi perangkat bergerak mencakup sistem operasi, *middleware*, dan aplikasi kunci. *Android Standart Development Kit* (SDK) menyediakan perlengkapan dan *Application Programming Interface* (API) yang diperlukan untuk mengembangkan aplikasi pada *platform* Android menggunakan bahasa pemrograman Java (<https://id.wikipedia.org/wiki/Android>).

Android dikembangkan oleh Google bersama *Open Handset Alliance* (OHA) yaitu aliansi perangkat selular terbuka yang terdiri dari 47 perusahaan *Hardware*, *Software* dan perusahaan telekomunikasi ditujukan untuk mengembangkan standar terbuka bagi perangkat seluler.

### 2.4.2 Sejarah Perkembangan Android

Pada mulanya terdapat berbagai macam sistem operasi pada perangkat selular, diantaranya sistem operasi Symbian, Microsoft Windows Mobile, Mobile Linux, iPhone, dan sistem operasi lainnya. Namun diantara sistem operasi yang ada belum mendukung standar dan penerbitan API yang dapat dimanfaatkan secara keseluruhan dan dengan biaya yang murah. Kemudian Google ikut berkecimpung didalamnya dengan platform Android, yang menjanjikan keterbukaan, keterjangkauan, *open source*, dan *framework* berkualitas.

Pada tahun 2005, Google mengakuisisi perusahaan Android Inc. untuk memulai pengembangan platform Android. Dimana terlibat dalam pengembangan

ini Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Pada pertengahan 2007 sekelompok pemimpin industri bersama-sama membentuk aliansi perangkat. Selular terbuka, Open Handset Alliance (OHA). Bagian dari tujuan aliansi ini adalah berinovasi dengan cepat dan menanggapi kebutuhan konsumen dengan lebih baik, dengan produk awalnya adalah platform Android. Dimana Android dirancang untuk melayani kebutuhan operator telekomunikasi, manufaktur handset, dan pengembang aplikasi. OHA berkomitmen untuk membuat Android *open source* dengan lisensi Apache versi 2.0.

### **2.4.3 Kelebihan Android**

Berikut ini adalah beberapa kelebihan dari OS android.

#### 1. *User Friendly*

Kata *User Friendly* sangat melekat pada sistem operasi *Windows* milik *Microsoft*, ibarat pengguna dengan sangat mudah mengoperasikan komputernya hanya dengan belajar beberapa hari bahkan beberapa jam saja, dan ini juga melekat pada Android yang berjalan pada *Smartphone*.

#### 2. *Notifications*

Pengguna dapat dengan sangat mudah mendapatkan notifikasi dari *smartphone* Android dengan mengatur beberapa akun pada sebuah aplikasi.

#### 3. Tampilan

Android tidak kalah bagus dari IOS milik Apple, karena memang dari awal Android hampir mengusung teknologi IOS, hanya saja dapat dibilang ini versi murahnya.

#### 4. *Open Source*

*Operating system* ini memang dibuat *open source* oleh penciptanya, karena memang berbasis kernel Linux dan sangat banyak *Custom Room* yang dibuat untuk masing-masing perangkat Android.

#### 5. Aplikasi

Sangat banyak aplikasi yang disajikan bahkan jutaan pilihan aplikasi yang menarik dari yang gratis hingga berbayar.

#### **2.4.4 Kelemahan Android**

Berikut ini adalah beberapa kelemahan dari OS android.

1. Sistem operasi Android tampaknya menuntut pengguna untuk harus memiliki koneksi internet dalam keadaan aktif. Seperti minimalnya perlu koneksi internet GPRS, hal ini agar perangkat siap untuk online sesuai dengan kebutuhan pengguna.
2. Memang terdapat banyak Aplikasi Android yang dapat digunakan secara gratis, akan tetapi seringkali pada aplikasi yang digunakan akan memunculkan iklan yang cukup mengganggu
3. Baterai pada smartpone dengan sistem Android akan sangat boros dibandingkan OS lainnya, hal tersebut disebabkan dengan banyaknya proses yang berjalan secara background yang membuat energi baterai menjadi cepat habis.

#### **2.5 Teknologi Pengembangan Aplikasi**

##### **2.5.1 *United Markup Language* (UML)**

*Unified Markup Language* menurut Grady Booch (2013), UML di definisikan sebagai bahasa *visual* untuk menjelaskan, memberikan spesifikasi,



merancang, membuat model dan mendokumentasikan aspek-aspek dari sebuah sistem. Dalam aplikasi Sistem Informasi Geografis Fasilitas Umum Kota Purwokerto berikut adalah Model UML yang dipakai dalam pengembangan aplikasi ini antara lain *Use Case Diagram*, *Activity Diagram*, dan *Flowchart Diagram*.

a. *Use Case Diagram*

*Use Case Diagram* mendeskripsikan sebuah interaksi antara satu atau lebih *actor* dengan sistem yang akan dibuat. Dapat dikatakan *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut adalah keterangan tentang tabel *Use Case Diagram*.

b. *Activity Diagram*

*Activity Diagram* merupakan diagram yang digunakan untuk menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis.

c. *Flowchart Diagram*

*Flowchart diagram* merupakan sebuah diagram dengan simbol-simbol grafis yang menyatakan aliran algoritma atau proses yang menampilkan langkah-langkah yang disimbolkan dalam bentuk kotak, beserta urutannya dengan menghubungkan masing masing langkah tersebut menggunakan tanda panah. Diagram ini bisa memberi solusi selangkah demi selangkah untuk penyelesaian masalah yang ada di dalam proses atau algoritma tersebut.

### 2.5.2 Basisdata (*Database*)

*Database* merupakan kumpulan informasi yang disimpan dalam sebuah komputer secara sistematis sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data (*database*) tersebut.

Penerapan *database* dalam suatu informasi disebut dengan *database System*. *Database* digunakan untuk menyimpan informasi atau data yang terintegrasi dengan baik di dalam komputer.

*Database* adalah sekumpulan data yang sudah disusun sedemikian rupa dengan ketentuan atau aturan tertentu yang saling berelasi sehingga memudahkan pengguna dalam mengelolanya juga memudahkan memperoleh informasi. Selain itu ada pula yang mendefinisikan *database* sebagai kumpulan *file*, tabel, atau arsip yang saling terhubung yang disimpan dalam media elektronik.

*Database* terbentuk dari sekelompok data-data yang memiliki jenis/sifat yang sama. Sebagai contoh data nama, data kelas, data alamat dikelompokkan dalam data baru yaitu mahasiswa. Demikian juga, kumpulan dari data-data mahasiswa, data-data dosen, data-data keuangan dan lainnya dapat dikumpulkan lagi menjadi kelompok besar, misalkan data-data jurusan atau fakultas pada sebuah universitas. Bahkan dalam perkembangannya, data-data tersebut dapat berbentuk berbagai macam data, misalkan dapat berupa program, lembaran-lembaran untuk *entry* (memasukkan) data, laporan-laporan. Semuanya itu dapat dikumpulkan menjadi satu yang disebut dengan *database*.

*Database* perlu disimpan di dalam basis data untuk keperluan penyediaan informasi lebih lanjut. Data didalam basis data supaya dirganasasikan sedemikian rupa, sehingga membentuk informasi yang lebih berkualitas.

Perancangan basis data merupakan upaya untuk membangun sebuah *database* dalam suatu lingkungan bisnis. Untuk membangun sebuah basis data terdapat tahapan-tahapan yang perlu dilalui yaitu:

1. Perencanaan basis data
2. Mendefinisikan sistem
3. Analisa dan mengumpulkan kebutuhan
4. Perancangan basis data
5. Perancangan aplikasi
6. Membuat *prototipe*
7. Implementasi
8. Konversi data
9. Pengujian
10. Pemeliharaan operasional

### **2.5.3 Bahasa pemrograman Java**

Pemrograman Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer atau *smartphone*. Bahasa pemrograman ini banyak mengadopsi sintaks yang terdapat pada bahasa C dan C++ namun dengan sintaks model yang lebih. Aplikasi berbasis Java pada umumnya dikompile ke dalam *p-code* (*bytecode*) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM).

Program Java merupakan bahasa yang bersifat umum (*general purpose*), dan secara khusus didesain untuk memanfaatkan dependensi implementasi

seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi Java mampu berjalan di beberapa *platform* sistem operasi yang berbeda, Java juga dikenal dengan slogannya “Tulis sekali, jalankan dimanapun”. Bidang ini merupakan bahasa pemrograman yang paling populer digunakan dan secara luas dimanfaatkan dalam berbagai jenis perangkat lunak aplikasi *mobile* ataupun aplikasi berbasis *web*.

Berikut adalah beberapa penjelasan tentang Kelebihan yang dimiliki oleh Java yaitu:

a. *Multiplatform*:

Java dapat dijalankan dalam beberapa *platform* komputer dan sistem operasi yang berbeda.

b. OOP atau *Object Oriented Programming*:

Java memiliki *library* yang lengkap. *Library* disini adalah sebuah kumpulan dari program yang disertakan dalam Java. Hal ini akan memudahkan pemrograman menjadi lebih mudah karena kelengkapan *library* semakin beragam jika ditambah dengan karya komunitas Java.

c. *Multithread*:

kemampuan suatu program komputer untuk mengerjakan beberapa proses dalam suatu waktu. *Thread* dalam Java memiliki kemampuan untuk memanfaatkan kelebihan multi prosessor apabila sistem operasi yang digunakan mendukung multi prosessor.

d. Dapat didistribusi dengan mudah

Java memiliki *library* rutin yang lengkap untuk dirangkai pada protocol TCP/IP (seperti HTTP dan FTP) dengan mudah. Kemampuan *networking* Java lebih kuat dan lebih mudah digunakan. Java memudahkan tugas pemrograman jaringan yang sulit seperti membuka dan mengakses sebuah socket koneksi. Java juga memudahkan pembuatan CGI (Common Gateway Interface).

e. Bersifat dinamis

Java dirancang untuk beradaptasi dengan lingkungan yang sedang berkembang, Java bersifat dinamis dalam tahap *linking*. *Class* yang ada dapat di *link* sebatas yang diperlukan, apabila diperlukan modul kode yang baru dapat di *link* dari beberapa sumber, bahkan dari sumber dalam jaringan Internet.

Berikut adalah beberapa penjelasan tentang kekurangan yang dimiliki oleh java yaitu:

a. Pada satu slogannya.

“Tulis sekali dan jalankan dimana saja” ternyata tidak sepenuhnya benar. Beberapa hal harus disesuaikan jika dijalankan pada *platform* yang berbeda. Misalnya untuk J2SE dengan *platform* SWT-AWT bridge tidak dapat berfungsi di Mac OS X.

b. Kemudahan aplikasi Java didekompilasi.

Dekompilasi adalah suatu proses membalikkan sebuah aplikasi menjadi kode sumbernya. Hal ini memungkinkan terjadi pada Java karena berupa *bytecode* yang menyimpan bahasa tingkat tinggi. Hal ini terjadi pula pada *platform* .NET dari *Microsoft* sehingga program yang dihasilkan mudah dibajak kodenya karena sulit untuk disembunyikan.

c. Penggunaan memori yang banyak.

Penggunaan memori untuk program berbasis Java jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnya seperti *C/C++* dan *Pascal*. Biasanya ini bukan merupakan masalah bagi pihak yang menggunakan teknologi terbaru (karena trend memori terpasang makin murah), tetapi menjadi masalah bagi mereka yang masih harus berkuat dengan mesin komputer berumur lebih dari 4 tahun.

## 2.6 Perangkat Pendukung

### 2.6.1 Eclipse

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di semua *platform* (<https://id.wikipedia.org/wiki/Eclipse>). Dengan *software* eclipse penggunanya dapat *men-develop* atau membuat sebuah program tidak hanya di Android, kita bisa membuat program-program lain semisal *ber-platform* Java atau yang lainnya. Berikut ini adalah sifat dari Eclipse:

a. *Multi-platform:*

Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.

b. *Multilanguage*

Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti *C/C++*, Cobol, Python, Perl, PHP, Java dan lain sebagainya.

c. *Multi-role*

Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plug-in*.

*Android Development Tools* (ADT) adalah *plugin* yang didesain untuk IDE Eclipse yang memberikan kita kemudahan dalam mengembangkan aplikasi Android dengan menggunakan IDE Eclipse. Dengan menggunakan ADT untuk Eclipse akan memudahkan kita dalam membuat aplikasi *project* Android, membuat GUI aplikasi, dan menambahkan komponen-komponen yang lainnya, begitu juga kita dapat melakukan *running* aplikasi menggunakan Android SDK melalui Eclipse. Dengan ADT juga kita dapat melakukan pembuatan *package* Android (.apk) yang digunakan untuk distribusi aplikasi Android yang kita rancang. Mengembangkan aplikasi Android dengan menggunakan ADT di eclipse sangat dianjurkan dan sangat mudah untuk memulai mengembangkan aplikasi Android.

Semakin tinggi *platform* Android yang digunakan, dianjurkan menggunakan ADT yang lebih terbaru, karena biasanya munculnya *platform* baru diikuti oleh munculnya versi ADT yang terbaru. Untuk melakukan instalasi ADT di-eclipse dapat dilakukan secara *online* maupun *offline*.

### 2.6.2 SQLite

SQLite merupakan sebuah *software* sistem manajemen basisdata relasional yang bersifat *ACID-compliant* dan memiliki ukuran pustaka kode yang relatif kecil, ditulis dalam bahasa C dan tidak perlu tersambung dengan koneksi internet. SQLite adalah proyek yang bersifat *public domain* yang dikerjakan oleh D. Richard Hipp (<https://id.wikipedia.org/wiki/SQLite>).

Sqlite salah satu *database* baru yang perkembangannya cukup cepat, dan pada saat ini sudah cukup populer. *Database* ini bersifat *Open Source* dan meraih *best open source* 2005. Para developer dapat menggunakan add on Mozilla dengan nama *SQLite Manager* untuk membuat mengelola dan menghasilkan file *database* SQLite.

SQLite adalah sebuah *open source* yang telah ada cukup lama, cukup stabil dan sangat terkenal pada perangkat kecil termasuk Android menyediakan *database* relational yang ringan untuk setiap aplikasi menggunakan SQLite. Aplikasi dapat mengambil sebuah keuntungan dari itu untuk mengatur relasional *database engine* dan untuk menyimpan data secara aman dan efisien. SQLite di Android termasuk dalam *Android Runtime*, sehingga setiap versi dari Android dapat membuat *database* dengan SQLite.