BAB II

LANDASAN TEORI

2.1. Tinjauan Jurnal

Annisa (2017:1), menyatakan bahwa:

Mahasiswa baru belajar akuntansi dalam mencari informasi tentang syarat dasar akuntansi ke dalam kamus memakan waktu lama karena butuh waktu untuk membuka halaman kamus istilah akuntansi. Solusi dari masalah ini adalah untuk memberikan informasi digital sebagai media untuk menerjemahkan bentuk kamus. Untuk merancang sistem aplikasi menggunakan metode interpolasi. Alasan untuk menggunakan pencarian ini metode interpolasi yang hasil pencarian dapat ditampilkan dalam istilah akuntansi yang optimal dan cepat. pencarian data menggunakan metode interpolasi dalam teori untuk menghitung dan menentukan posisi indeks, hasil yang diperoleh dengan indeks yang sama antara teori dan praktek pengujian menggunakan program yang telah dibuat. Hasil dari penelitian ini adalah proses menerjemahkan istilah akuntansi untuk interpolasi metode pencarian mudah, baik dan benar.

Weni dkk (2013:1), menyatakan bahwa:

Sistem operasi android merupakan system operasi mobile berbasis kernel linux yang dikembangkan oleh android inc dan kemudian diakui sisi oleh google, penelitian yang dilakukan penulis dalam menghasilkan sebuah apikasi kamus jaringan berbasis android dapat dilakukan dengan baik serta hasil yang dicapai cukup baik karena system bisa melakukan modifikasi sangat cepat.

2.2. Konsep Dasar Program

2.2.1. Java

Menurut Suyanto (2015:2), menyatakan bahwa:

Java diciptakan oleh suatu tim yang dipimpin oleh Patrick Naughton dan James Gosling dalam suatu proyek dari sun microsystem yang memiliki kode green dengan tujuan untuk menghasilkan bahasa komputer sederhana yang dapat dijalankan di peralatan sederhana dengan tidak terikat pada arsitekture tertentu, mulanya disebut oak, tetapi karena oak sendiri merupakan nama dari bahasa pemrograman komputer yang sudah ada, maka sun mengubahnya menjadi java.sun kemudian meluncurkan *browser* dari java yang disebut *hot java* yang mampu menjalankan applet. Setelah itu teknologi java diadopsi oleh Netscape yang memungkinkan program java dijalankan di browser netscape yang ke mudian diikuti *Internet Explore*. Karena keunikan dan kelebihannya, teknologi java mulai menarik banyak vendor seperti IBM, Symantec, Inprise, dll. Sun merilis versi awal java secara resmi pada awal tahun 1996 yang kemudian terus berkembang hingga muncul jdk 1.1 kemudian jdk 1.2 yang mulai disebut sebagai versi java2 karena banyak mengandung peningkatan dan perbaikan. Perubahan utama adalah swing yang merupakan teknologi GUI *(Graphical User Interface)* yang mampu menghasilkan windows yang portable. Dan pada tahun 1998-1999 lahirlah teknologi J2EE (Java 2 Enterprise Edition).

Java juga memiliki banyak fitur-fitur yang menarik menurut Suyanto (2015:3). Antara lain sebagai berikut:

1. *Applet*

   Program *Java* yang dapat berjalan di atas *browser*, yang dapat membuat halaman *HTML* lebih dinamis dan menarik.

2. *Java Networking*

   Sekumpulan API (*Application Programming Interface*) yang menyediakan fungsi-fungsi untuk aplikasi – aplikasi jaringan, seperti penyediaan akses untuk *TCP, UDP, IP Adrress* dan *URL*. Tetapi Java *Networking* tidak menyediakan akses untuk ICMP dikarenakan alasan sekurity dan pada kondisi umum hanya administrator (*root*) yang bisa memanfaatkan *protokol ICMP*.

3. *Java Database Connectivity* (JDBC**)**

   JDBC menyediakan sekumpulan API yang dapat digunakan untuk mengakses *database* seperti *Oracle*, *MySQL*, *PostgreSQL*, *Microsoft SQL Server*.

4. *Java Security*

   Java Security menyediakan sekumpulan API untuk mengatur security dari *aplikasi Java* baik secara *high level* atau *low level*, seperti *public/private key management* dan *certificates*.

5. *Java Swing*

   Java Swing menyediakan sekumpulan API untuk membangun aplikasi-aplikasi GUI (*Graphical User Interface*) dan model GUI yang diinginkan bisa bermacam–macam, bisa *model Java*, model *Motif CDE* atau model yang *independent* terhadap *platform* yang digunakan.

6. *Java RMI*

   Java RMI menyediakan sekumpulan API untuk membangun aplikasi – aplikasi Java yang mirip dengan model RPC (*Remote Procedure Call*) jadi *object-object Java* bisa di call secara *remote* pada jaringan komputer.

7. *Java 2D/3D*

   Java 2D/3D menyediakan sekumpulan API untuk membangun *grafik – grafik 2D/3D* yang menarik dan juga akses ke printer.

8. *Java Server Pages*

   Berkembang dari *Java Servlet* yang digunakan untuk menggantikan aplikasi–aplikasi CGI, JSP (*Java Server Pages*) yang mirip ASP dan *PHP* merupakan alternatif terbaik untuk solusi aplikasi *Internet*.

9. JNI (*Java Native Interface*)

   JNI menyediakan sekumpulan API yang digunakan untuk mengakses fungsi–fungsi pada *library* yang dibuat dengan bahasa *pemrograman* yang lain seperti *C,C++,* dan *Basic.*

10. *Java Sound*

    Java Sound menyediakan sekumpulan API untuk *manipulasi sound*.

11. *Java IDL + CORBA*

    Java IDL (*Interface Definition Language*) menyediakan dukungan Java untuk *implementasi* CORBA (*Common Object Request Broker*) yang merupakan *model distributed-Object* untuk solusi aplikasi besar di dunia *networking.*

12. *Java Card*

    Java Card utamanya digunakan untuk aplikasi – aplikasi pada *smart card*, yang sederhana wujudnya seperti *SIM Card* pada *handphone*.

13. JTAPI (*Java Telephony API*)

    Java Telepony API menyediakan sekumpulan API untuk memanfaatkan *devices –devices telephone*, sehingga akan cocok untuk aplikasi– aplikasi CTI (*Computer Telephony Integration*) yang dibutuhkan seperti ACD (*Automatic Call Distribution*), *PC-PBX* dan lainnya.

2.2.2. *JDK (Java Development Kit)*

Menurut Khannedy (2011:1), mengemukakan bahwa: "*Java Development Kit* merupakan perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode *java* menjadi *bytecode* yang dapat dimengerti dan dapat dijalankan oleh *Java Runtime Envirotment*".

2.2.3. *SDK* (*Software Development Kit*)

Menurut Ichwan dan Hakiky (2011:4), mengemukakkan bahwa: SDK (*Software Development Kit*) adalah satu alat pengembang aplikasi untuk software tertentu. Demikian pula dengan *android, google* telah menyiapkan sebuah SDK yang dapat digunakan sebagai alat pengembang aplikasi mobile berbasis sistem operasi android.

2.2.4. *Android*

> Menurut Juhara (2016:1), menyatakan bahwa:
> Android adalah s*istem Operasi* berbasis *linux* yang dimodifikasi untuk perangkat bergerak (*Mobile Devices)* yang terdiri dari *Sistem Operasi*, *middleware*, dan aplikasi-aplikasi utama. Awalnya, *Android* dikembangkan oleh *Android Inc*. Perusahaan ini kemudian dibeli oleh *google* pada tahun 2005. Sistem operasi *Android* kemudian diluncurkan bersamaan dibentuknya Organisasi *Open Hanset Alliance* pada tahun 2007. Selain *google*, beberapa nama-nama besar juga ikut serta dalam *open handset alliance* antara lain: *Motorola, Samsung, LG, Sony Ericsson, T-Mobile, Vodafone, Toshiba,* dan *Intel.*

Android telah mengalami pembaharuan sejak pertama kali dirilis, rata-rata versi terbaru dari android dirillis setiap 6 bulan. Berikut beberapa jenis versi pembaharuan android, level *Application Programming Interface* (API) dan penjelasannya menurut Juhara (2016:2), Yaitu:

1. *Cupcake*

   Android 1.5 (*API* Level 3) yang dikenal dengan nama kode *cupcake* ini dirillis pada april 2009 API level 3 membawa perbaikan dari versi sebelumnya dan tentu saja fitur baru. Misalnya dukungan *On-screen keyboard, video recording, home screen widget, fitur auto pairing* pada *bluetooth, kernel linux versi 2.6.7* dan kemampuan memperbaiki *filesystem sd card* yang rusak.

2. *Donut*

   Android 1.6 (*API* Level 4) yang dirillis pada Oktober 2009 ini merupakan *rillis minor* dengan fitur utama berupa dukungan teknologi *CDMA, gesture, text-to-speech engine,* dan fitur pencarian cepat yang memungkinkan anda mencari semua hal seperti kontak, *riwayat jelajah internet, boomark*, dan lainnya.

3. *Éclair*

   Android 2.1 (*API* Level 7) yang dirillis pada januari 2010 ini mengusung fitur baru seperti *live wallpaper* dan beberapa fitur baru seperti android 2.0 misalnya menggunakan banyak akun, kontak cepat, *bluetooth* 2.1 dan *profil bluetooth* baru yaitu *Object Push Profile* (OPP) dan *Phone Book Access Profile* (PBAP).

4. *Froyo*

   Android 2.2 (*API* Level 8) dirillis pada Mei 2010, fitur baru yang dibawa adalah dukungan *OpenGL ES 2.0*, instalasi aplikasi penyimpanan *eksternal* (*SD card*), dan *Android Cloud to Device Messaging* yang memungkinkan aplikasi melakukan *push messaging dan portable hostspot* yang

memungkinkan piranti android menjadi *hotspot wi-fi* untuk berbagi *koneksi internet*.

5. *Gingerbread*

Android 2.3 (*API* Level 9 dan 10) mengusung beberapa fitur baru, antara lain dukungan banyak kamera, *Near Field Communication (NFC*), *download manager service,* dukungan terhadap sensor lain seperti *giroskop*, dan *barometer*.

6. *Honeycomb*

Android 3.0 (*API* Level 11) membawa perubahan besar terutama pada tampilan *User Interface* (UI). Yang berubah *drastis*, supaya optimal untuk piranti layar besar seperti *tablet*. Pada versi ini diperkenalkan *fragment, action bar, system clipboard, dan cursor loader.*

7. *Ice Cream Sandwich*

Android 4.0 (*API* Level 14), membawa perubahan besar terutama pada tampilan *User Interface* (UI) android 3.0 supaya cocok pada layar kecil sehingga memungkinkan aplikasi anda tampak konsisten di tablet maupun ponsel. Fiture baru seperti *android baem* dan *wi-fi direct* juga ditambahkan.

8. *Jellybean*

Android 4.1 (*API* Level 16), adalah rilis minor yang membawa perbaikan fungsionalis dan *performa rendering User-Interface (UI). Versi 4.2 (API Level 17*) mengusung fiture baru seperti *gesture typing* dan dukungan *multiuser* di piranti tablet. Versi 4.3 (API Level 18) membawa perbaikan berupa dukungan *Bluetooth low energy* dan *OpenGL ES 3.0* beberapa fiture baru antara lain dukungan terhadap bahasa internasional dan penulisan text

dua arah (kiri ke kanan atau kanan ke kiri untuk bahasa-bahasa tertentu seperti bahasa arab)

9. *Kitkat*

Android 4.4 (*API* Level 19), ini mengusung sejumlah perbaikan dan fitur terutama dukungan teknologi *Near Field Communication (NFC*) melalui *host card emulation*, pencetakan *ke printer nirkabel, webview* dengan *rendering engine chromium,* dan dukungan yang lebih baik bagi piranti yang menggunakan *ram* rendah.

2.2.5. Eclipse

Menurut Faridayonarisa (2012), *Eclipse* adalah sebuah IDE *(Integrated Development Environment)* untuk mengembangkan perangkat lunak dan dapat dijalankan di semua platform *(platform-independent)*, sifat dari *Eclipse* adalah :

a. *Multi-platform*: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
b. *Multi-language*: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, Cobol, Python, Perl, PHP, 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.

2.2.6. XML (*Extensible Markup Language*)

Menurut Wardhani (2016:1), Mengemukakkan bahwa:
XML (*Extensible Markup Language*) di manfaatkan dalam mendefinisikan dokumen dengan format standar dimana yang dapat dibaca dan di dukung oleh aplikasi-aplikasi XML yang kompatibel. Bahasa format xml bisa digunakan dengan halaman HTML, akan tetapi XML itu sendiri bukan bahasa markup. Sebaliknya, XML itu merupakan "*metabahasa*" yang dapat di pakai dalam membuat bahasa markup untuk aplikasi khusus. Sebagai contoh nya itu, dapat menggambarkan item yang bisa diakses di saat membutuhkan Halaman Web Dimana pada dasarnya, XML ini dapat memungkinkan Anda untuk membuat database informasi tanpa memiliki database yang sebenarnya. Yaa meskipun secara default hanya digunakan dalam aplikasi Web, banyak program lainnya juga yang dapat menggunakan dokumen XML, misalnya kode sumber aplikasi Android.

2.3. Metode *Algoritma*

Menurut Sitorus (2015:42) Algoritma merupakan fondasi yang harus dipahami atau dikuasai oleh seseorang yang akan menyelesaikan suatu masalah dengan komputer, dalam hal ini membuat program. Prinsip kerja algoritma menjelaskan deskripsi pelaksanaan suatu proses, sehingga proses akan dikerjakan sesuai dengan algoritma yang telah ditulis.

Pencarian *linier* dapat dilakukan pada barisan bilangan yang terurut secara menaik (*ascending*) atau menurun (*descending*) ataupun tidak terurut. Pencarian Linier dilakukan dengan cara membandingkan data yang dicari (X) dengan data dalam barisan A[1] … A[n] dengan dimulai dari data elemen pertama pada barisan A. Jika perbandingan bernilai sama, maka pencarian dihentikan dan dinyatakan sukses. Sedangkan apabila perbandingan tidak bernilai sama maka :

1. Jika data tidak terurut (data acak), maka pencarian akan dilanjutkan ke data selanjutnya.

2. Jika data terurut secara menaik (*ascending*), maka pencarian hanya akan dilanjutkan ke data selanjutnya yang berada di sebelah kanan data yang sedang dibandingkan apabila data yang dicari (X) lebih besar daripada data yang sedang dibandingkan sekarang.

3. Jika data terurut secara menurun (*descending*), maka pencarian hanya akan dilanjutkan ke data selanjutnya yang berada di sebelah kanan data yang sedang dibandingkan apabila data yang dicari (X) lebih kecil daripada data yang sedang dibandingkan sekarang. Jika syarat – syarat di atas dipenuhi, maka pencarian data akan dilakukan sampai data yang dicari (X) ditemukan sehingga pencarian dinyatakan sukses atau sampai elemen terakhir dari barisan A dan tidak ada elemen A yang sama dengan data yang dicari (X) sehingga pencarian dinyatakan gagal.

2.4. Pengujian Aplikasi

> Menurut Ayuliana (2009:1), mengemukakkan bahwa: Metode ujicoba *blackbox* memfokuskan pada keperluan *fungsional* dari *software*. Karna itu uji coba *blackbox* memungkinkan pengembang *software* untuk membuat himpunan kondisi *input* yang akan melatih seluruh syarat-syarat *fungsional* suatu *program*. Ujicoba *blackbox* bukan merupakan alternatif dari ujicoba *whitebox*, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya.

Ujicoba *blackbox* berusaha untuk menemukan kesalahaan dalam beberapa kategori menurut Ayuliana (2009:1), diantaranya:

1. Fungsi-fungsi yang salah atau hilang.
2. Kesalahan *interface.*
3. Kesalahan dalam struktur data atau akses *database eksternal.*

4. Kesalahan performa.
5. Kesalahan *inisialisasi* dan *terminasi*.

Menurut Ayuliana (2009:1), Ujicoba didesain untuk dapat menjawab pertanyaan-pertanyaan berikut:

1. Bagaimana validitas fungsionalnya diuji?
2. Jenis input seperti apa yang akan menghasilkan kasus uji yang baik ?
3. Apakah sistem secara khusus sensitif terhadap nilai input tertentu ?
4. Bagaimana batasan-batasan kelas data diisolasi?
5. Berapa rasio data dan jumlah data yang dapat ditoleransi oleh sistem?
6. Apa akibat yang akan timbul dari kombinasi spesifik data pada operasi sistem?

2.5. Peralatan Pendukung

Menurut Sulistyorini (2009:1), mengemukakkan bahwa: Saat ini piranti lunak semakin luas dan besar lingkupnya, sehingga tidak bisa lagi dibuat asal-asalan. Piranti lunak saat ini seharusnya dirancang dengan memperhatikan hal-hal seperti scalability, security, dan eksekusi yang robust walaupun dalam kondisi yang sulit. Selain itu arsitekturnya harus didefinisikan dengan jelas, agar bug mudah ditemukan dan diperbaiki, bahkan oleh orang lain selain programmer aslinya.

Menurut Sulistyorini (2009:1), mengemukakkan bahwa: *Aplikasi mobile learning* ini dirancang menggunakan UML dan dibuat ke dalam *diagram-diagram* yaitu:

1. Activity Diagram

Activity Diagram bersifat dinamis. *Activity Diagram* adalah tipe khusus dari *diagram state* yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dari suatu sistem. Diagram ini terutama penting dalam pemodelan

fungsi – fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek.

2. *Use Case Diagram*

Use Case Diagram bersifat statis. *Use Case Diagram* memperlihatkan himpunan *use case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna.

3. *Sequence Diagram*

Sequence Diagram bersifat dinamis. *Diagram sequence* merupakan diagram interaksi yang menekankan pada pengiriman pesan (*message*) dalam suatu waktu tertentu.

4. *Class Diagram*

Diagram kelas bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolabirasi serta relasi.

5. *Deployment Diagram*

Deployment Diagram bersifat statis. *Deployment Diagram* memperlihatkan konfigurasi saat aplikasi dijalankan (saat *run time*). Dengan ini memuat simpul–simpul (*node*) beserta komponen–komponen yang ada di dalamnya. *Deployment diagram* berhubungan erat dengan *diagram kompoen* dimana *deployment diagram* memuat satu atau lebih komponen - komponen diagram ini sangat berguna saat aplikasi berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing*).