

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Penulisan tugas akhir ini berdasarkan referensi dari beberapa skripsi sebelumnya yang berkaitan dengan sistem yang akan dibuat. Adapun beberapa penelitian terdahulu yang dijadikan sebagai referensi adalah sebagai berikut.

Menurut (Indriyawati & Suprayogi, 2011) dalam jurnal yang berjudul “Pengembangan Radio *Online* Sebagai Pemanfaatan Teknologi Informasi dan Komunikasi Untuk Pengembangan Potensi Mahasiswa di Lingkungan Universitas Semarang “ menyatakan bahwa Radio FTIK USM yang sekarang ini jangkauan siarannya masih terbatas karena masalah pemancar. Salah satu teknologi yang digunakan adalah dengan menggunakan metode *streaming*. Dengan teknologi *streaming* yang memungkinkan distribusi data audio, video yang besar secara real time melalui internet, sehingga siaran yang di transmisikan dapat diterima oleh pendengar dimanapun dia berada.

Menurut (Ayubi, 2016) dalam skripsi yang berjudul “Rancang Bangun Aplikasi Radio *Streaming* dengan menggunakan *Shoutcast Distributed Network Audio Server (DNAS)*” menyatakan perkembangan media massa *Radio Streaming* sebagai teknologi komunikasi audio di Indonesia telah menunjukkan peningkatannya. Pengguna *radio streaming* dari waktu ke waktu semakin bertambah seiring dengan semakin meningkatnya jumlah dan beraneka ragamnya jenis stasiun radio di Indonesia sesuai dengan minat dan permintaan para konsumennya. Atas dasar inilah penulis membuat sebuah aplikasi radio *streaming* yang bertujuan membantu pengguna untuk mendengarkan radio *streaming*, menampilkan *content program* radio dan menampilkan jadwal siaran dari radio Ilmu komputer.

Menurut (Triyono, Al Azam, & Roliawati, 2012) dalam jurnal yang berjudul “Aplikasi Radio *Streaming* Menggunakan Server ICECAST2 Berbasis *Android*” menyatakan bahwa radio konvensional yang saat ini berada di frekuensi 500 KHz -

108 MHz yang mempunyai keterbatasan wilayah dan geografis ketika pendengar sudah berpindah kota. Berdasarkan uraian diatas maka dibuatlah sebuah aplikasi yang bertujuan untuk memudahkan pengguna dalam berinteraksi langsung dan lebih dekat dengan penyiar lewat aplikasi *Android Client Radio Streaming (ACRS)*. Dengan fitur aplikasi *direct dial* dan *direct sms*, *launcher* jejaring sosial *facebook* dan *twitter* yang menghubungkan stasiun radio langsung dari aplikasi *ACRS*.

Berdasarkan beberapa penelitian diatas, penelitian yang dilakukan peneliti akan membuat aplikasi radio *streaming* bingen radio berbasis android dengan beberapa fitur solusi pada aplikasi yang dibuat, yaitu:

1. Aplikasi dapat menampilkan informasi penyiar radio, *event* dari cafe dan promo yang diadakan dari cafe.
2. Aplikasi akan dilengkapi dengan informasi jadwal acara yang ada pada siaran radio.

2.2 Landasan Teori

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

2.2.1 Radio

Radio adalah teknologi yang digunakan untuk pengiriman sinyal dengan cara modulasi dan radiasi elektromagnetik (gelombang elektromagnetik). Gelombang ini melintas, dan merambat lewat udara, dan bisa juga merambat lewat ruang angkasa yang hampa udara, karena gelombang ini tidak memerlukan medium pengangkut (seperti molekul udara).

Sejarah radio adalah sejarah teknologi yang menghasilkan peralatan radio yang menggunakan gelombang radio. Awalnya sinyal pada siaran radio ditransmisikan melalui gelombang data yang kontinu baik melalui modulasi amplitudo (AM), maupun modulasi frekuensi (FM). Metode pengiriman sinyal seperti ini disebut analog. Selanjutnya, seiring perkembangan teknologi ditemukanlah internet, dan sinyal digital yang kemudian mengubah cara transmisi sinyal radio (Sawyer & Williams, 2001).

Radio memiliki beberapa jenis diantaranya adalah sebagai berikut :

1. Radio AM (*Amplitude Modulation*)

Radio AM (*Amplitude Modulation*) merupakan jenis radio yang menggunakan prinsip kerja modulasi gelombang radio dan audio. Baik gelombang radio dan audio memiliki besaran yang tetap, namun dengan proses modulasi, amplitudo gelombang radio akan disesuaikan dengan amplitudo gelombang audio. Dalam konsep ini, gelombang radio menjadi gelombang pengantar atau *carrier* sementara gelombang suara akan menjadi gelombang pembawa informasi (Sawyer & Williams, 2001).

2. Radio FM (*Frequency Modulation*)

Radio FM (*frequency modulation*) adalah jenis radio yang memodulasi frekuensi gelombang radio. Gelombang penghantar, dalam hal ini gelombang radio, dimodulasi frekuensinya mengikuti gelombang pembawa informasi atau gelombang suara. Perbedaan antara radio AM dan FM adalah jenis modulasi yang dilakukan, dimana radio AM memodulasi amplitudo gelombang sedangkan radio FM memodulasi frekuensi gelombang. Pengembangan radio FM merupakan dampak dari kekurangan yang dimiliki radio AM, yang dikembangkan sebelumnya.

Radio FM memiliki kekurangan sangat rentan terhadap interferensi khususnya oleh gangguan cuaca. Armstrong memulai risetnya untuk menggunakan modulasi gelombang dengan membuat amplitudo gelombang radio konstan. Di tahun 1933, Armstrong berhasil mengembangkan radio FM dengan hasil suara yang lebih baik, jernih dan kuat terhadap gangguan cuaca. Meski penemuan ini sangat bermanfaat, radio FM tidak langsung dinikmati publik secara masal. Hal ini dikarenakan tingginya biaya penggantian *transmitter* dan *receiver* yang sebelumnya menggunakan prinsip radio AM (Sawyer & Williams, 2001).

3. Radio Internet

Radio internet (dikenal juga sebagai *web radio*, *radio streaming* dan *e-radio*) bekerja dengan cara mentransmisikan gelombang suara lewat internet. Prinsip kerjanya hampir sama dengan radio konvensional yang gelombang pendek (*short wave*), yaitu dengan menggunakan *medium streaming* berupa gelombang yang kontinyu. Sistem kerja ini memungkinkan siaran radio terdengar ke seluruh dunia asalkan pendengar memiliki perangkat internet. Itulah sebabnya banyak kaum ekspatriat yang menggunakan radio internet untuk mengobati rasa kangen pada negara asalnya. Di Indonesia, umumnya radio internet dikolaborasikan dengan sistem radio analog oleh stasiun radio teresterial untuk memperluas jangkauan siarannya (Sawyer & Williams, 2001).

4. Radio Satelit

Radio satelit mentransmisikan gelombang audio menggunakan sinyal digital. Berbeda dengan sinyal analog yang menggunakan gelombang kontinyu, gelombang suara ditransmisikan melalui sinyal digital yang terdiri atas kode-kode biner 0 dan 1. Sinyal ini ditransmisikan ke daerah jangkauan yang jauh lebih luas karena menggunakan satelit. Hanya saja siaran radio hanya dapat diterima oleh perangkat khusus yang bisa menerjemahkan sinyal terenkripsi. Siaran radio satelit juga hanya bisa diterima di tempat terbuka dimana antena pada pesawat radio memiliki garis pandang dengan satelit pemancar. Radio satelit hanya bisa bekerja yang tidak memiliki penghalang besar seperti terowongan atau gedung. Oleh karena itu perangkat radio satelit banyak dipromosikan untuk radio mobil. Untuk mendapat transmisi siaran yang baik (Sawyer & Williams, 2001).

5. Radio HD (*Hight Definition*)

Radio yang dikenal juga sebagai radio digital ini bekerja dengan menggabungkan sistem analog dan digital sekaligus. Dengan begitu memungkinkan dua stasiun digital dan analog berbagi frekuensi yang

sama. Efisiensi ini membuat banyak konten bisa disiarkan pada posisi yang sama. Kualitas suara yang dihasilkan HD radio sama jernihnya dengan radio satelit, tetapi layanan yang ditawarkan gratis. Namun untuk dapat menerima siaran radio digital pendengar harus memiliki perangkat khusus yang dapat menangkap sinyal digital (Sawyer & Williams, 2001).

2.2.2 Aplikasi

Aplikasi adalah penggunaan dalam suatu komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses *input* menjadi *output*.

Program merupakan kumpulan *instruction set* yang akan dijalankan oleh pemroses, yaitu berupa *software*. Bagaimana sebuah sistem komputer berpikir diatur oleh program ini. Program inilah yang mengendalikan semua aktifitas yang ada pada pemroses. Program berisi konstruksi logika yang dibuat oleh manusia, dan sudah diterjemahkan ke dalam bahasa mesin sesuai dengan format yang ada pada *instruction set*. Program aplikasi merupakan program siap pakai. Program yang direka untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Contoh-contoh aplikasi ialah program pemroses kata dan *Web Browser*. Aplikasi akan menggunakan sistem operasi (OS) komputer dan aplikasi yang lainnya yang mendukung.

Istilah ini mulai perlahan masuk ke dalam istilah Teknologi Informasi semenjak tahun 1993, yang biasanya juga disingkat dengan app. Secara historis, aplikasi adalah *software* yang dikembangkan oleh sebuah perusahaan. App adalah *software* yang dibeliperusahaan dari tempat pembuatnya. Industri PC tampaknya menciptakan istilah ini untuk merefleksikan medan pertempuran persaingan yang baru, yang paralel dengan yang terjadi antar sistem operasi yang dimunculkan (Hartono, 1999).

2.2.3 Android

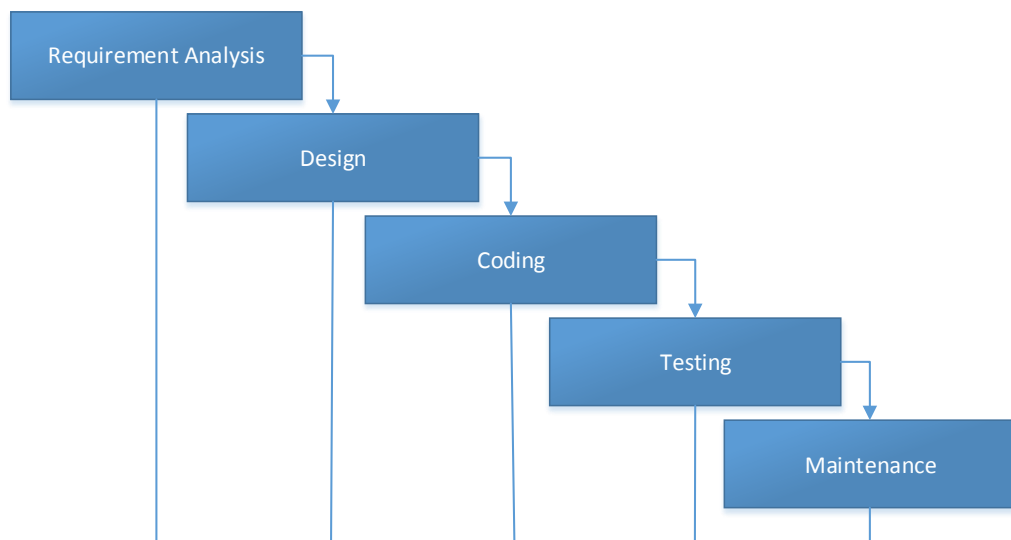
Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis Linux. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan

aplikasi mereka sendiri yang akan digunakan untuk 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. Pada saat perilis perdana *Android*, 5 November 2007, *Android* bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, *Google* merilis kode-kode *Android* di bawah lisensi Apache, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler (Safaat, 2012).

2.2.4 System Development Life Cycle (SDLC)

Metode pengembangan sistem merupakan metode yang digunakan untuk mengembangkan suatu sistem informasi, yaitu suatu proses standar yang diikuti untuk melaksanakan seluruh langkah yang diperlukan untuk menganalisa, merancang, mengimplementasikan, dan memelihara sistem informasi. Daur hidup dari pengembangan sistem ini disebut dengan daur hidup pengembangan sistem *System Development Life Cycle (SDLC)*.

Dalam pengembangan sistem pada aplikasi ini menggunakan *System Development Life Cycle (SDLC)* dengan model *Waterfall*. Model *Waterfall* merupakan salah satu metode dalam SDLC yang mempunyai ciri khas pengerjaan setiap fase dalam *Waterfall* harus diselesaikan terlebih dahulu sebelum melanjutkan ke fase selanjutnya. Artinya, fokus terhadap masing-masing fase dapat dilakukan maksimal karena tidak terdapat pengerjaan yang sifatnya paralel (Mustakini, 1991). Berikut ini merupakan contoh metode *waterfall* seperti pada gambar 2.1.



Gambar 1.1 Metode Waterfall

2.2.5 Teknologi Pengembangan Aplikasi

2.2.5.1 Unified Modeling Language (UML)

UML (*Unified Modeling Language*) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML 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. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek seperti *C++*, *Java*, *C#* atau *VB.NET*. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

Seperti bahasa-bahasa lainnya, UML mendefinisikan notasi dan *syntax* atau semantik. Notasi UML merupakan sekumpulan bentuk khusus untuk menggambarkan berbagai diagram piranti lunak. Setiap bentuk memiliki makna tertentu, dan UML *syntax* mendefinisikan bagaimana bentuk-bentuk tersebut dapat dikombinasikan. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: Grady Booch OOD (*Object-Oriented Design*), Jim Rumbaugh OMT

(*Object Modeling Technique*), dan Ivar Jacobson OOSE (*Object-Oriented Software Engineering*).

1. *Use Case Diagram*

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah interaksi antara aktor dengan sistem.

Seorang atau sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. *Use case* diagram dapat sangat membantu ketika sedang menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan pelanggan, dan merancang *test case* untuk semua *feature* yang ada pada sistem.

2. *Activity Diagram*

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity* diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan *state* diagram khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity* diagram tidak menggambarkan behaviour internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.

3. *Flowchart*

Flowchart merupakan gambar atau bagan yang memperlihatkan urutan dan hubungan antar proses beserta instruksinya. Gambaran ini dinyatakan dengan simbol. Dengan demikian setiap simbol menggambarkan proses tertentu. Sedangkan hubungan antar proses digambarkan dengan garis penghubung.

Flowchart ini merupakan langkah awal pembuatan program. Dengan adanya flowchart urutan poses kegiatan menjadi lebih jelas. Jika ada penambahan proses maka dapat dilakukan lebih mudah. Setelah *flowchart* selesai disusun, selanjutnya pemrogram (*programmer*) menerjemahkannya ke bentuk program dengan bahasa pemrograman.

4. *Class Diagram*

Class diagram adalah sebuah spesifikasi yang jika diinstansi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (attribut atau properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut.

Class diagram menggambarkan struktur dan deskripsi *class*, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain (Dharwiyanti & Wahono, 2003).

2.2.5.2 Bahasa Pemograman Java

Java adalah sebuah bahasa pemrograman yang sangat terkenal. Sebagai bahasa pemrograman, *Java* dapat digunakan untuk menulis program. Sebagai mana diketahui program adalah kumpulan instruksi yang ditujukan untuk komputer. Melalui program, komputer dapat diatur agar melaksanakan tugas tertentu sesuai yang ditentukan oleh pemrogram (orang yang membuat program).

Bahasa pemrograman *java* dikembangkan di Sun Microsystem dan mulai diperkenalkan kepada publik pada tahun 1995. Seperti halnya C++, *java* juga merupakan bahasa yang berorientasi objek. Dengan demikina , *java* juga memudahkan dalam pembuatan aplikasi yang besar.

Sebagai bahasa yang beraras tinggi, yang menggunakan perintah-perintah yang mudah dimengerti oleh orang. *Java* mempunyai keunggulan yakni bersifat universal. Sebagai bahasa yang universal, *java* bisa dijumpai di berbagai *platform* (*Linux*, *UNIX*, *Windows*, *Mac*, dan lain-lain). Artinya, jika sudah menguasai *java* di *platform* PC, sangat mudah untuk berpindah di *Linux* ataupun sistem operasi yang

lain. Hasil kompilasi *java* yang dinamakan bytecode dapat dijalankan di berbagai *platform* sepanjang di sistem target memiliki *Java Runtime Enviroment* (JRE).

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

1. *Multiplatform:*

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

2. OOP atau *Object Oriented Programming:*

Java memiliki *library* yang lengkap. *Library* di sini 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.

3. *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.

4. Dapat di distribusi 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*).

5. 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 kekurangan yang dimiliki oleh *java* yaitu.

1. Pada satu slogan nya.

“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* SWTAWT bridge tidak dapat berfungsi di Mac OS X.

2. Kemudahan aplikasi Java di dekompilasi.

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.

3. 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 tetapi menjadi masalah bagi mereka yang masih harus berlutut dengan mesin komputer berumur lebih dari 4 tahun (Kadir, 2013).

2.2.5.3 Bahasa Pemograman PHP

PHP atau singkatan dari *Personal Home Page* merupakan bahasa script yang tertanam dalam HTML untuk dieksekusi bersifat *server side*. PHP termasuk dalam *open source product*, sehingga *source code* dari PHP dapat diubah dan didistribusikan secara luas.

PHP juga dapat berjalan pada berbagai *web server* seperti IIS (*Internet Information Server*), PWS (*Personal Web Server*), *Apache*, *Xitami*. PHP juga mampu berjalan dibanyak sistem operasi yang beredar saat ini, diantaranya sistem operasi *Microsoft Windows* (semua versi), Linux, Mac OS, Solaris. PHP dapat dibangun sebagai modul *web server Apache* dan sebagai *binary* yang dapat berjalan sebagai CGI (*Common Gateway Interface*). PHP dapat mengirim *HTTP header*, dapat mengatur *cookies*, mengatur *authentication* dan *redirect user*.

Salah satu keunggulan yang dimiliki PHP adalah kemampuan untuk melakukan koneksi ke berbagai macam *software* sistem manajemen basis data atau *Database Management Sistem* (DBMS), sehingga dapat menciptakan suatu halaman *web* dinamis. PHP mempunyai konektivitas yang baik dengan beberapa DBMS seperti *Oracle, Sybase, Msql, MySQL Microsoft SQL Server, Solid, PostgreSQL, Adabas, FilePro, Velocis, dBase, Unix dbm* dan tidak terkecuali semua database berinterface ODBC (Nugroho, 2006).

2.2.5.4 Database

Database adalah kumpulan data-data yang tersimpan, tersusun, dan saling terhubung satu sama lain pada suatu komputer serta digunakan perangkat lunak untuk mengakses maupun mengelolanya sehingga dapat dihasilkan informasi yang berguna. Atau secara singkatnya definisi *Database* yaitu kumpulan data yang tersimpan pada suatu komputer dan saling terhubung antara satu sama lain sehingga dapat digunakan untuk tujuan tertentu.

Database memiliki suatu konsep yang dimana konsep tersebut diantaranya:

1. *Field*

Dapat di katakan *field* merupakan suatu bagian dari *record* atau bagian dari suatu tabel yang merupakan item-item kolom data, atau disebut juga sebagai implementasi dari atribut data.

2. *Record*

Field-field akan diorganisasikan menjadi *record-record*. Jadi *record* dapat dikatakan sebagai struktur yang berisi sejumlah komponen *field*. Pada *record*, *field-field* disusun dalam format yang sudah di tentukan.

3. *File* dan Tabel

Record yang sama diorganisasikan menjadi grup yang di sebut dengan *file*. *File* dapat di katakan sebagai kumpulan fakta dari struktur *record*. Sedangkan tabel merupakan suatu objek pada *Database* yang digunakan untuk menyimpan sekumpulan *file*.

Adapun keuntungan atau manfaat dari penggunaan *Database* diantaranya seperti.

1. **Dapat Meminimalkan Redudansi Data**

Redudansi merupakan penyimpanan data yang sama dalam berkas yang berbeda-beda. Jika data di *update* akan terjadi secara berulang-ulang. Maka dengan *Database* redudansi data dapat di minimalkan bahkan bisa di hindari.
2. **Integritas Data yang Tinggi**

Data yang sama akan saling berelasi, jika ada perubahan pada suatu data maka data yang sama pada file yang sama akan mengalami perubahan. Integritas data sangat penting pada *database* karena dapat memastikan keakuratan, konsistensi, aksesibilitas dan juga kualitas yang tinggi pada suatu data.
3. **Independensi Data**

Pada *database* data-data yang ada tidak bergantung pada *software* yang mengaksesnya, sebab struktur data pada *database* dirancang berdasarkan kebutuhan akan informasi jadi bukan berdasarkan *software*. Independensi juga dimana seseorang tidak bisa melakukan perubahan data jika data tersebut sedang diakses oleh orang lain.
4. **Dapat Berbagi Data**

Dengan sistem *database* maka kita dapat berbagi data atau informasi dengan pengguna lain, sehingga data tersebut dapat digunakan secara bersama-sama.
5. **Tingkat Keamanan Yang Tinggi**

Dengan sistem *database* kita dapat meningkatkan keamanan terhadap suatu data atau informasi sehingga tidak mudah untuk diakses orang lain, misalnya dengan memberikan *password* atau hak akses pada data tersebut.
6. **Mudahnya Mengakses Atau Mendapatkan Data**

Database dapat mempermudah *user* dalam mendapatkan data yang diperlukannya, dengan berbagi data yang ada dan tentunya dengan hak akses yang telah diberikan untuk mengakses *database* tersebut (Nikko, 2017).

2.2.6 Perangkat Pendukung

2.2.6.1 *Android studio*

Android studio adalah sebuah IDE untuk *Android Development* yang diperkenalkan google pada acara Google I/O 2013. *Android studio* merupakan pengembangan dari *Eclipse* IDE, dan dibuat berdasarkan IDE *Java* populer, yaitu *IntelliJ IDEA*. *Android studio* merupakan IDE resmi untuk pengembangan aplikasi *Android*.

Sebagai pengembangan dari *Eclipse*, *Android studio* mempunyai banyak fitur-fitur baru dibandingkan dengan *Eclipse* IDE. Berbeda dengan *Eclipse* yang menggunakan *Ant*, *Android studio* menggunakan *Gradle* sebagai *build environment* (Jadibaru, 2015).

Fitur-fitur lainnya adalah sebagai berikut:

1. Menggunakan *Gradle-based build system* yang fleksibel.
2. Dapat mem-*build multiple* APK.
3. *Template support* untuk *Google Services* dan berbagai macam tipe perangkat.
4. *Layout editor* yang lebih bagus.
5. *Built-in support* untuk *Google Cloud Platform*, sehingga mudah untuk integrasi dengan *Google Cloud Messaging* dan *App Engine*.
6. *Import library* langsung dari *Maven repository*.

2.2.6.2 *MySQL (My Structured Query Language)*

MySQL adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *web* yang menggunakan *database* sebagai sumber dan pengolahan datanya.

MySQL dikembangkan oleh perusahaan Swedia bernama *MySQL AB* yang pada saat ini bernama *Tcx DataKonsult AB* sekitar tahun 1994-1995, namun cikal bakal kodenya sudah ada sejak tahun 1979. Awalnya *Tcx* merupakan perusahaan pengembangan *software* dan konsultan *database*, dan saat ini *MySQL* sudah diambil alih oleh *Oracle Corp*.

Kepopuleran MySQL antara lain karena MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses *database* sehingga mudah untuk digunakan, kinerja *query* cepat, dan mencukupi untuk kebutuhan *database* perusahaan-perusahaan yang berskala kecil sampai menengah, MySQL juga bersifat *Open Source* (tidak berbayar).

MySQL merupakan *database* pertama kali yang didukung oleh bahasa pemrograman *script* untuk internet (PHP dan Perl). MySQL dan PHP dianggap sebagai pasangan *software* pembangun aplikasi *web* yang ideal. MySQL lebih sering digunakan untuk membangun aplikasi berbasis web. Umumnya pengembangan aplikasinya menggunakan Bahasa pemrograman *script* PHP. MySQL didistribusikan dengan lisensi Open Source GPL (*General Publik License*) mulai versi 3.23 pada bulan Juni 2000 (Rudianto, 2011).

2.2.7 Black Box Testing

Metode ujicoba *black box* memfokuskan pada keperluan fungsional dari sistem, karena itu ujicoba *black box* memungkinkan pengembang sistem atau aplikasi untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program. Ujicoba *black box* bukan merupakan alternatif dari ujicoba *white box*, tetapi merupakan pendekatan yang melengkapi untuk menemukan kesalahan lainnya, selain menggunakan metode *white box*.

Uji coba *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, serta kesalahan inisialisasi dan terminasi (Pressman, 2002).