

BAB II

TINJAUAN PUSTAKA

2.1 Pengertian Aplikasi

Aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti sistem perniagaan, game, pelayanan masyarakat, periklanan, atau semua proses yang hampir dilakukan manusia (Syahdaffa, 2016:23).

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. *Language software* berfungsi agar dapat menulis program dengan bahasa yang lebih mudah, dan akan menterjemahkannya ke dalam bahasa mesin supaya bisa dimengerti oleh komputer. Bila hendak mengembangkan suatu program aplikasi untuk memecahkan permasalahan yang besar dan rumit, maka supaya program aplikasi tersebut dapat berhasil dengan baik, maka dibutuhkan prosedur dan perencanaan yang baik dalam mengembangkannya.

Sekarang, banyak sekali program-program aplikasi yang tersedia dalam bentuk paket-paket program. Ini adalah program-program aplikasi yang sudah ditulis oleh orang lain atau perusahaan-perusahaan perangkat lunak. Beberapa perusahaan perangkat lunak telah memproduksi paket-paket perangkat lunak yang mempunyai reputasi *internasional*. Program-program paket tersebut dapat diandalkan, dapat memenuhi kebutuhan pemakai, dirancang dengan baik, relatif bebas dari kesalahan-kesalahan, *user friendly* (mudah digunakan), mempunyai dokumentasi manual yang

memadai, mampu dikembangkan untuk kebutuhan mendatang, dan didukung perkembangannya.

Akan tetapi, bila permasalahannya bersifat khusus dan unik, sehingga tidak ada paket-paket program yang sesuai untuk digunakan, maka dengan terpaksa harus mengembangkan program aplikasi itu sendiri.

2.2 Sistem

Sistem adalah sekumpulan komponen yang saling bekerja sama untuk mencapai suatu tujuan. Menurut Sistem adalah suatu jaringan kerja dari prosedur- prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu. Dari penelitian diatas dapat disimpulkan bahwa sistem informasi saling berinteraksi atau bekerja sama untuk mencapai suatu tujuan (Jogiyanto, 2004).

2.2.1 Karakteristik Sistem

Menurut (Jogiyanto, 2004) Sistem yang baik mempunyai faktor penentu atau karakteristik tertentu, yaitu komponen-komponen (*components*), batas sistem (*boundary*), lingkungan luar sistem (*environment*), perhubungan (*interface*), masukan (*input*), keluaran (*output*), pengelolaan (*process*), dan sasaran (*objective*), atau tujuan (*goal*).

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen- komponen sistem atau elemen-elemen sistem yang dapat berupa subsistem atau bagian-bagian dari sistem.

2. Batas Sistem

Batas sistem (*boundary*) merupakan daerah yang membatasi daerah antara satu sistem dengan sistem yang lainnya atau dengan lingkungan lainnya. Batas sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan.

3. Lingkungan Luar Sistem

Lingkungan luar (*environment*) dari suatu sistem adalah apapun diluar dari sistem yang mempengaruhi operasi sistem. Lingkungan luar sistem dapat bersifat menguntungkan dan juga merugikan sistem tersebut. Lingkungan yang menguntungkan merupakan energi dari sistem dengan demikian harus tetap dijaga dan dipelihara. Sedangkan lingkungan luar sistem yang merugikan harus ditahan dan dikendalikan, jika tidak maka akan mengganggu kelangsungan hidup sistem.

4. Penghubung Sistem

Penghubung (*interface*) merupakan media penghubung antara satu sub dengan subsistem lainnya. Melalui penghubung ini memungkinkan sumberdaya mengalir dari satu sub ke subsistem lainnya. Keluaran (*output*) dari sub sistem menjadi masukan (*input*) untuk subsistem yang lainnya dengan melalui penghubung.

5. Masukan Sistem

Masukan (*input*) adalah energi yang dimasukkan kedalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*). *Maintenance* input adalah energi yang dimasukkan agar sistem dapat beroperasi. *Signal input* adalah energi yang diproses untuk mendapatkan keluaran.

6. Keluaran Sistem

Keluaran (*output*) adalah hasil energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolahan yang akan merubah masukan menjadi keluaran. Suatu sistem produksi akan mengolah masukan berupa bahan baku dan bahan-bahan lain menjadi keluaran berupa barang-barang jadi.

8. Sasaran Sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objective*). Jika suatu sistem tidak mempunyai sasaran, maka operasi sistem tidak akan ada gunanya. Sasaran dari sistem sangat menentukan masukan yang dibutuhkan sistem dan keluaran yang akan dihasilkan sistem. Suatu sistem dikatakan berhasil bila mengenai sasaran dan tujuan.

2.2.2 Klasifikasi Sistem

Menurut (Jogiyanto,2004) Sistem dapat diklasifikasikan dari beberapa pandangan, antara lain adalah sebagai berikut :

1. Sistem diklasifikasikan sebagai sistem abstrak (*abstract system*) dan sistem fisik (*physical system*).
2. Sistem diklasifikasikan sebagai sistem alamiah (*natural system*) dan sistem buatan manusia (*human made system*).
3. Sistem diklasifikasikan sebagai sistem tertentu (*deterministic system*) dan sistem terbuka (*probabilistic system*).
4. Sistem diklasifikasikan sebagai sistem tertutup (*close system*) dan sistem terbuka (*open system*).

2.3 Tanda Tangan Digital

Digital Signature adalah suatu jenis teknologi yang digunakan untuk meningkatkan keamanan jaringan. *Digital Signature* memiliki fungsi sebagai penanda pada data yang memastikan bahwa data tersebut adalah

data yang sebenarnya (tidak ada yang berubah). Dengan begitu, *Digital Signature* dapat memenuhi setidaknya dua syarat keamanan jaringan, yaitu *Authenticity* dan *Nonrepudiation*.

Sedangkan jika dilihat dari Cara kerja *Digital Signature* adalah dengan memanfaatkan dua buah kunci, yaitu kunci publik dan kunci privat. Kunci publik digunakan untuk mengenkripsi data, sedangkan kunci privat digunakan untuk mendekripsi data. Pertama, dokumen di-*hash* dan menghasilkan *Message Digest*. Kemudian, *Message Digest* dienkripsi oleh kunci publik menjadi *Digital Signature*.

Untuk membuka *Digital Signature* tersebut diperlukan kunci privat. Bila data telah diubah oleh pihak luar, maka *Digital Signature* juga ikut berubah sehingga kunci privat yang ada tidak akan bisa membukanya. Ini merupakan salah satu syarat keamanan jaringan, yaitu *Authenticity*. Artinya adalah, keaslian data dapat terjamin dari perubahan-perubahan yang dilakukan pihak luar.

2.4 Android

Menurut (Arifianto, 2011), Android merupakan perangkat bergerak pada sistem operasi untuk telepon seluler yang Menggunakan *Linux*. Menurut (Hermawan, 2011), *Android* merupakan *OS (Operating System) Mobile* yang tumbuh ditengah *OS* lainnya yang berkembang dewasa ini. Berdasarkan pendapat diatas, maka dapat ditarik kesimpulan bahwa *Android* adalah sistem operasi Menggunakan *Linux* yang sedang berkembang ditengah *OS* lainnya.

2.5 Android Studio

Android studio adalah lingkungan pengembangan terpadu, *integrated device electronic* (IDE) resmi untuk pengembangan *platform android*. *Android studio* pada awalnya tahap *preview* / percobaan versi 0.1

yang dipakai pada tanggal 1 Mei 2013 dan memasuki tahap *beta* pada bulan Juni 2014. Ketika sudah mulai stabil, *android studio* dirilis pada Desember 2014 dengan *versi 1.0*. *Android studio* yang digunakan dalam pembuatan aplikasi ini adalah *Android studio versi 2.1.2*. Berdasarkan *JetBrains IDEA IntelliJ Software*, *android studio* dirancang khusus untuk pengembangan *android* yang tersedia untuk *windows*, *Mac OS X*, dan *Linux* (Dewi, 2017).

2.5.1 Android Software Development Kit (SDK)

SDK adalah singkatan dari *Software Development Kit* yaitu merupakan *software* yang dibuat untuk membangun aplikasi *android*. (Dewi, 2017). Saat ini disediakan *Android SDK (Software Development Kit)* sebagai alat bantu dan *API* untuk mulai mengembangkan aplikasi pada *platform Android* menggunakan bahasa pemrograman *Java*. Ketika proses *install* berlangsung akan disediakan pilihan untuk *package* dari *Android* yang akan digunakan untuk pengembangan, mulai dari *Android 1.5* hingga *Android 4.0*.

2.5.2 Kelebihan Android Studio

Tampilan *android studio* dilengkapi dengan *multiscreen*, dimana pengembang dapat menulis kode di satu sisi dan melihat tampilannya di sisi lain mengedit kode (Dewi, 2017).

2.5.3 Kekurangan Android Studio

Ukuran file *android studio* untuk *windows* yang direkomendasikan sangat besar hampir 1 GB. Sedangkan jika *download* tanpa *SDK tools* hanya sekitar 240 MB. Selain itu *android studio* memerlukan ukuran *RAM* yang cukup besar, *Android studio* sendiri merekomendasikan minimal 2 GB *RAM* untuk bisa menjalankan *android studio* dengan lancar (Dewi, 2017).

2.6 Flutter

Flutter adalah *Mobile App SDK (Software Development Kit)* untuk membuat aplikasi *Android* dan *iOS* dari satu *codebase* dengan performa tinggi. Artinya kita hanya perlu mempelajari *Flutter* untuk membangun aplikasi *mobile* untuk 2 *platform*. Dalam *mobile apps development* android dan *iOS* ada banyak *tools* yang dapat digunakan untuk mempermudah dalam proses *developmentnya*, lebih spesifik lagi dalam *mobile apps development* di Indonesia ada beberapa *tools* pengembangan *mobile apps*, diantaranya *Native, Hybrid, dan Cross-platform*.

Flutter merupakan *SDK* besutan *Google* yang ditujukan untuk *mobile apps development* yang memiliki kinerja tinggi. *Mobile apps* yang di kembangkan dengan *Flutter* dapat dipublikasi ke *platform* *Android* maupun *iOS* dari *codebase* tunggal. Pertama dirilis pada Mei 2017 dan Pada event *Google I/O* awal Juni yang lalu *Flutter* mendapatkan *update versi preview* 1. *Flutter* dapat mudah dipelajari karena menggunakan bahasa pemrograman *Dart* terlebih jika sudah familiar dengan *java* dan *java script*.

Flutter ditulis dengan bahasa pemrograman *C, C++, Dart*, menggunakan *Skia* milik *Google* sebagai mesin *render 2D*, dan *Blink* digunakan untuk sistem *render* serta didukung dengan *Mojo IPC*. *Flutter* dapat dijalankan sistem operasi *android*.

4.1 atau lebih dan *iOS* 8 atau lebih tinggi. *Apps* dari *Flutter* juga dapat dijalankan pada *android emulator* serta *IOS emulator*.

Pada *Google I/O* yang lalu *Flutter* juga mendapatkan banyak fitur baru. diantaranya *Material Design Widget, FlutterFire (Firebase untuk Flutter), Admobs*, serta *support* untuk *resource animasi GIF dan WebP*.

Flutter menggunakan mesin *render* bawaannya sendiri pada saat menampilkan *widgetnya*. Berbeda dengan *framework cross-platform* seperti *React Native, Fuse, Native Script* dsb yang menggunakan *Webview* untuk menampilkan *widgetnya*. karena tidak menggunakan *UI bawaan* dari *OEM* yang tertanam pada *device*, maka *Flutter* akan memiliki tampilan *UI*

yang konsisten dan unik pada semua *platform* saat *mobile apps* dipublikasikan.

Seperti disebutkan diatas, ada banyak *framework* cross-platform *mobile apps development*, jika dibandingkan dengan *framework cross-platform* yang lain (Krisnada, 2019).

2.6.1 Kelebihan *Flutter*

1. Animasi *Super Smooth*, jika tidak ada elemen yang berlebihan maka *mobile apps* yang dikembangkan dengan menggunakan *Flutter* dapat berjalan secara konstan 60-120 *fps*. Karena menggunakan sistem *render* miliknya sendiri.
2. *IDE* terkenal, Dapat dikembangkan dengan *IDE* seperti *Android Studio*, *intellij IDEA* serta *VSCode*.
3. Mudah dipelajari, untuk yang lebih familiar dengan bahasa *Java*, *Objective-C* ataupun *Swift* *Flutter* akan mudah dipahami
4. Penggunaan *Object Oriented Programming* (OOP).
5. *Hot Reload*, *build mobile apps* untuk *debug* lebih cepat.
6. Dalam tulisan kali ini, penulis akan sedikit mengupas mengenai *Flutter*, yaitu *software development kit (SDK) cross-platform* yang digunakan untuk *mobile apps development android* dan *iOS* yang mungkin belum banyak diketahui dan digunakan oleh *mobile apps developer* di Indonesia.

2.6.2 Kekurangan *Flutter*

1. Tidak adanya *JSX*, karena tidak adanya *JSX* maka *file Dart* tidak terdapat pengelompokan kode *components*, *styling* karena penulisan kode akan dilakukan *inline* didalam objek kode tersebut
2. *File APK* dan *IPA* lebih besar, konsekuensi dari konstannya *performa mobile apps* yang dikembangkan dari *Flutter* berimbas

pada ukuran *file* yang dihasilkan, karena *Flutter* menggunakan *Widgetnya* sendiri.

3. Logika animasi lebih sulit, selain *File* yang dihasilkan lebih besar, performa tinggi dari *mobile apps* yang dihasilkan juga berimbas pada saat penulisan logika pembuatan *mobile apps*.
4. *Environment* pengembang masih dalam perkembangan, karena *Flutter* baru berumur 7-8 tahun maka jika terjadi *bug* atau kesalahan pada proses pengembangan, akan sulit untuk mencari solusinya.

Cara kerja *Flutter* di *Android* dan *iOS* kode yang ditulis akan dikompilasi dengan mesin C dan C++ dengan menggunakan *Android NDK*, kerangka kode akan dibentuk dalam kode *native* yang akan dikompilasi lagi dengan *Dart Compiler*. Sedangkan pada *iOS* kode akan dikompilasi dengan *LLVM* dan *mobile apps* dijalankan dengan kumpulan instruksi *native* tanpa *interpreter/penerjemah*.

Flutter merupakan *SDK* yang dibuat oleh *Google* yang ditujukan untuk mendukung pengembangan *platform* baru dari *Google* di masa yang akan datang yaitu *Google Fuchsia*.

2.7 Web service

Menurut Sibagariang (2016) *Web servise* merupakan suatu komponen *software* yang merupakan *self containing*, aplikasi *modular self-describing* yang dapat dipublikasikan, dialokasikan, dan dilaksanakan pada *web*.

Web service adalah teknologi yang mengubah kemampuan internet dengan menambahkan kemampuan *transactional web*, yaitu kemampuan *web* untuk saling berkomunikasi dengan pola *program-to-program*. Fokus *web* selama ini didominasi oleh komunikasi *program-to-user* dengan interaksi *business-to-consumer*, sedangkan *transactional web* akan didominasi oleh *program-to-program* dengan interaksi *business-to-business*.

Komponen *web service* yang mana menyediakan fasilitas komunikasi jarak jauh antara dua aplikasi yang merupakan *layer arsitektur web service*.

1. *Layer 1 : Protokol* internet standar yang digunakan sebagai sarana *transportasi* adalah *HTTP* dan *TCP/IP*.
2. *Layer 2 : Simple Object Access Protocol (SOAP)* berbasiskan XML dan digunakan untuk pertukaran informasi antar sekelompok layanan.
3. *Layer 3 : Web Service Definition Language (WSDL)* digunakan untuk mendeskripsikan atribut layanan.
4. *Layer 4 : Universal Description, Discovery and Integration*, yang mana merupakan direktori pusat untuk deskripsi layanan.

2.8 Flowchart

Menurut Seprida (2017), badan alir (*flowchart*) adalah bagan (*chart*) yang menunjukkan alir (*flow*) di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi.

