BAB II

LANDASAN TEORI

2.1. Konsep Dasar Program

Dalam pembuatan Tugas Akhir, dibutuhkan teori yang memudahkan pemahaman tentang proses perancangan hingga pembuatan sistem pakar android seperti yang di harapkan. Dengan adanya sistem pakar android dapat digunakan secara maksimal maka diharapkan *user* atau pengguna aplikasi dapat dimudahkan mendeteksi dini dalam khasus rabun mata dan mendapatkan solusi untuk menangani masalah tersebut. Untuk memudahkan dalam memahami Penulis menggunakan beberapa referensi berbeda yang dapat ditemukan di Daftar Pustaka. Berikut adalah teori pendukung yang dapat memperkuat penulisan Tugas Akhir ini.

2.1.1. Sistem

Menurut Azhar Susanto (2013:22) "Sistem adalah kumpulan atau group dari *sub* sistem atau bagian atau komponen apapun baik *phisik* ataupun *non phisik* yang saling berhubungan satu sama lain dan bekerja sama secara harmonis untuk mencapai satu tujuan tertentu".

> Tujuan sistem menurut Ahzar Susanto (2013:23) :
> Tujuan sistem merupakan target atau sasaran akhir yang ingin dicapai oleh suatu sistem. Agar supaya target tersebut bisa tercapai, maka target atau sasaran tersebut harus diketahui terlebih dahulu ciri-ciri atau kriterianya. Upaya mencapai suatu sasaran tanpa mengetahui ciri-ciri kriteria dari sasaran tersebut kemungkinkan besar sasaran tersebut tidak akan pernah tercapai. Ciri-ciri atau kriteria dapat juga digunakan sebagai tolak ukur dalam menilai suatu keberhasilan suatu sistem dan menjadi dasar dilakukannya suatu pengadilan.

Dari pengertian diatas dapat ditarik kesimpulan bahwa sistem merupakan kumpulan suatu komponen sistem yang saling berhubungan satu dengan yang lain untuk mencapai tujuan suatu kegiatan pokok perusahaan.

2.1.2. Pakar

Pakar merupakan orang yang mempunyai kemampuan dalam menghadapi masalah dengan baik dan benar. Lewat pengalaman, seorang pakar mengembangkan kemampuan yang membuatnya dapat memecahkan suatu masalah (Purwanto & Destiani, 2015). Sebagai contoh, dokter adalah seorang pakar yang mampu mendiagnosis penyakit yang diderita pasien serta dapat memberikan penatalaksanaan terhadap penyakit tersebut. Tidak semua orang dapat mengambil keputusan mengenai diagnosis dan memberikan penatalaksanaan suatu penyakit. Contoh yang lain, montir adalah seorang yang punya keahlian dan pengalaman dalam menyelesaikan kerusakan motor atau mobil.

2.1.3. Sistem Pakar

Menurut Hayadi (2016:1) menyimpulkan bahwa "Sistem pakar atau *expert system* disebut juga dengan *knowledge based system* yaitu suatu aplikasi komputer yang ditujukan untuk membantu pengambilan keputusan atau pemecahan persoalan dalam bidang yang spesifik". Sistem ini bekerja dengan menggunakan pengetahuan dan metode analisis yang telah didefinisikan terlebih dahulu oleh pakar yang sesuai dengan bidang keahliannya. Sistem disebut sistem pakar karena fungsi dan perannya sama seperti seorang ahli yang harus memiliki pengetahuan, pengalaman dalam memecahkan suatu persoalan.

Pada dasarnya sistem pakar diterapkan untuk mendukung aktivitas pemecahan masalah, beberapa aktivitas pemecahan masalah yang dimaksud seperti (Lestari, 2012):

a. *Interpretasi*

Membuat kesimpulan atau deskripsi dari sekumpulan data mentah. Pengambilan keputusan dari hasil observasi termasuk pengenalan ucapan, analisis citra, interpretasi sinyal.

b. Prediksi

Memproyeksikan akibat-akibat yang dimungkinkan dari situasi-situasi tertentu, contoh prediksi demografi, prediksi eknomi.

c. Diagnosis

Menentukan sebab malfungsi dalam situasi kompleks yang didasarkan pada gejala-gejala yang teramati diagnosis medis, elektronis, mekanis.

d. Perancangan Desain

Menentukan konfigurasi komponen-komponen sistem yang cocok dengan tujuan-tujuan kinerja tertentu yang memenuhi kendala-kendala tertentu. Contoh perancangan *layout* sirkuit, bangunan.

e. Perencanaan

Merencanakan serangkaian tindakan yang akan dapat mencapai sejumlah tujuan dengan kondisi awal tertentu. Contoh: perencanaan keuangan, militer.

f. *Monitoring*

Membandingkan hasil pengamatan dengan kondisi yang diharapkan, Contoh: *computer aided monitoring system.*

g. *Debugging*

Menentukan dan menginterpretasikan cara-cara untuk mengatasi malfungsi. Contoh: Memberikan resep obat terhadap kegagalan.

h. Instruksi

Mendeteksi dan mengoreksi defisiensi dalam pemahaman domain subjek, Contoh: melakukan instruksi untuk diagnosis dan *debugging*.

i. Kontrol

Mengatur tingkah laku suatu environment yang kompleks, Contoh: Melakukan kontrol terhadap interpretasi, prediksi, perbaikan dan monitoring kelakukan sistem.

Menurut Nita Merlina dan Rahmat Hidayat dalam bukunya Perancangan Sistem Pakar (2012:4), Berikut ini adalah manfaat dan kemampuan sistem pakar:

a. Meningkatkan *output* dan produktivitas.
b. Menurunkan waktu pengambilan keputusan.
c. Meningkatkan kualitas proses dan produk.
d. Menyerap keahlian langka.
e. Fleksibilitas.
f. Operasi peralatan yang lebih mudah.
g. Eliminasi kebutuhan peralatan yang mahal.
h. Transfer pengetahuan ke lokasi terpencil.

Dalam sistem pasti memiliki kelebihan dan kekurangan, dibawah ini akan dijelaskan beberapa kelebihan dan kekurangan sistem pakar.

a. Kelebihan

Ada banyak keuntungan bila menggunakan sistem pakar, diantaranya adalah (Widayanto et al, 2018):

1. Meningkatkan ketersediaan (*increased availability*). Kepakaran atau keahlian menjadi tersedia dalam sistem komputer. Dapat dikatakan bahwa sistem pakar merupakan produksi kepakaran secara masal (*massproduction*).
2. Mengurangi biaya (*reduced cost*). Biaya yang diperlukan untuk menyediakan keahlian per satu orang user menjadi berkurang.
3. Mengurangi bahaya (*reduced danger*). Sistem pakar dapat digunakan dilingkungan yang mungkin berbahaya bagi manusia.
4. Permanen (*permanence*). Sistem pakar dan pengetahuan yang terdapat di dalamnya bersifat lebih permanen dibandingkan manusia yang dapat merasa lelah, bosan, dan pengetahuannya hilang saat sang pakar meninggal dunia.
5. Keahlian multipel (*multiple expertise*). Pengetahuan dari beberapa pakar dapat dimuat ke dalam sistem dan bekerja secara simultan dan kontinyu menyelesaikan suatu masalah setiap saat. Tingkat keahlian atau pengetahuan yang digabungkan dari beberapa pakar dapat melebihi pengetahuan satu orang pakar.
6. Meningkatkan kehandalan (*increased reliability*). Sistem pakar meningkatkan kepercayaan dengan memberikan hasil yang benar sebagai alternatif pendapat dari seorang pakar atau sebagai penengah jika terjadi konflik antara beberapa pakar. Namun hal tersebut tidak berlaku jika sistem dibuat oleh salah seorang pakar, sehingga akan selalu sama dengan pendapat pakar tersebut kecuali jika sang pakar melakukan kesalahan yang mungkin terjadi pada saat tertekan atau stres.

b. Kekurangan

Selain kelebihan di atas, sistem pakar juga memiliki beberapa kelemahan, diantaranya :

1. Biaya yang sangat mahal untuk membuat dan memeliharanya.
2. Sulit dikembangkan karena keterbatasan keahlian dan ketersediaan pakar.
3. Sistem pakar tidak 100% bernilai benar.

Pada dasarnya banyak sistem yang bisa kita gunakan contohnya sistem pakar dan sistem konvensional. Berikut perbandingan antara sistem pakar dan sistem konvensional (Ashari, 2016):

Tabel II.1

Tabel perbandingan Sistem Pakar dan Sistem Konvensional

Sistem Pakar	Sistem Konvensional
1. *Knowledge base* terpisah dari mekanisme	1. Informasi dan pemrosesan umumnya digabung dalam satu
2. Pemprosesan (*interface*)	2. Program *sequential*
3. Program bisa melakukan kesalahan	3. Program tidak pernah salah (kecuali pemrogramannya yang salah)
4. Penjelasan (*explanation*) merupakan bagian dari ES	4. Tidak menjelaskan mengapa input dibutuhkan atau bagaimana hasil diperoleh
5. Data tidak harus lengkap	5. Data harus lengkap
6. Perubahan pada rules dapat dilakukan dengan mudah	6. Perubahan pada program merepotkan
7. Sistem bekerja secara *heuristic* dan *logic*	7. Sistem bekerja jika sudah lengkap

2.1.4. 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 (Pranama 2012). Aplikasi merupakan program yang dikembangkan untuk memenuhi kebutuhan pengguna dalam menjalankan pekerjaan tertentu (Yuhefizar, 2012).

Berdasarkan beberapa pendapat diatas, dapat disimpulkan bahwa aplikasi merupakan sebuah program yang dibuat dalam sebuah perangkat lunak dengan komputer untuk memudahkan pekerjaan atau tugas-tugas seperti penerapan, penggunaan dan penambahan data yang dibutuhkan.

2.1.5. Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi (Safaat, 2014). Android merupakan sistem operasi yang berbasis *Linux* dan dirancang untuk perangkat seluler layar sentuh seperti *smartphone* serta komputer tablet. *Android* pada awalnya dikembangkan oleh perusahaan bernama Android, Inc., dengan dukungan finansial yang berasal dari Google, yang kemudian Google pun membelinya pada tahun 2005. Sistem operasi android tersebut secara resmi dirilis pada tahun 2007.

Android merilis versinya dengan nama makanan ringan sesuai abjad dengan tujuan agar pengguna dapat dengan mudah mengingatnya. Berikut tabel yang menunjukan versi-versi *Android* yang telah dirilis hingga saat ini (Irsyad, 2016) :

Tabel II.2.

Versi Android

Code name	Versi Number	Release Date
(No codename)	1.0	September 23,2008
(*Internally known as *petit Four*)	1.1	Februari 9, 2009
Cupcake	1.5	April 27, 2009
Donut	1.6	September 15,2009
Éclair	2.0 – 2.1	October 26, 2009
Froyo	2.2 – 2.2.3	May 20, 2010
Gingerbread	2.3 – 2.3.7	December 6, 2010
Honeycomb	3.0 – 3.2.6	February 22, 2011
Ice Cream Sandwich	4.0 – 4.0.4	October 18, 2011
Jelly Bean	4.1 – 4.3.1	July 9, 2012
Kitkat	4.4 -4.4.4	October 31, 2013
Lollipop	5.0 – 5.1.1	November 12, 2014
Marsmallow	6.0 – 6.0.1	October 5, 2015
Nougat	7.0 – 7.1.2	August 22, 2016
Oreo	8.0	August 21, 2017

Sumber (Irsyad, 2016)

Android menjadi sistem operasi yang sangat popular dan di gemari oleh pengguna *Smartphone*. Ada beberapa kelebihan dan kekurangan android yaitu :

a. Kelebihan :

1. Antarmuka yang mudah digunakan

   Tidak butuh waktu yang lama bagi seorang pemula untuk dapat menguasai penggunaan ponsel berbasis Android. Hal ini tentu menjadi kelebihan tersendiri bagi Android.

2. Tampilan yang dapat diubah

   Tidak seperti iOS, tampilan Android dapat dengan mudah di kostamisasi. Mulai dari ikon, tema, *wallpaper*, dan masih banyak lagi.

Hal ini menjadikan Android sebagai sistem operasi yang tidak membosankan.

3. *Open Source*

   Seperti yang sudah katakan diatas bahwa Android bersifat *open source*. Sehingga dapat dikembangkan lebih jauh sesuai dengan kebutuhan para penggunanya.

4. Terus diperbarui

   Karena dikembangkan langsung oleh Google, Android secara rutin mendapatkan pembaruan atau *update*. Baik itu berupa penambahan fitur baru, pembaruan keamanan, dan masih banyak lagi.

5. Banyak Aplikasi dan game gratis

   Kelebihan Android yang satu ini saya yakin sering kamu rasakan manfaatnya. Yaitu banyaknya aplikasi dan game yang bisa di *download* gratis dari *Google Play Store*.

b. Kekurangan :

1. Tidak semua *smartphone* Android mendapatkan *update*

   Kekurangan pertama yang sering dirasakan pengguna Android adalah tidak semua *smartphone* mendapatkan *update*. Karena walaupun Google rajin memperbarui Android, semua *update smartphone* kembali lagi pada pabrikan.

2. Terlalu banyak *merk* dan *tipe*

   Yang satu ini sebenarnya bisa jadi kekurangan dan juga kelebihan. Tapi menurut saya, lebih condong ke kekurangan. Karena terlalu banyak tipe dan merk, membuat penggunaannya jadi tidak konsisten. Tidak seperti

iPhone yang hanya memiliki 1 tipe saja dan dikembangkan oleh 1 pabrikan, yaitu Apple.

3. Lag dan Lemot

   Karena banyak merk dan tipe *smartphone* Android, maka spesifikasinya juga berbeda-beda. *Smartphone* Android yang memiliki spesifikasi rendah biasanya akan lebih mudah lag dan lemot.

2.1.6. Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai jenis komputer dan berbagai jenis sistem operasi termasuk telepon genggam (Wahana komputer, 2010:1). Meskipun pada awal saat dirilis sekitar tahun 90-an, Java dirancang untuk digunakan pada sistem-sistem kecil seperti TV kabel atau *home theatre*, sekarang sudah merambah keseluruhan aplikasi pada komputer bahkan beberapa institusi Pendidikan beralih dari pemprograman dengan *pascal* dan C++ ke pemprograman Java.

Java sedikit berbeda dengan kebanyakan program atau aplikasi lain. Java berdiri diatas sebuah mesin interpreter yang diberi nama *Java Virtual Machine* (JVM). JVM inilah yang akan membaca *bytecode* dalam suatu program sebagai representasi langsung program yang berisi mesin. Asalkan sistem operasi mempunyai JVM maka java dapat digunakan.

Secara sederhana pengembangan program berbasis Java dimulai dengan menulis kode sumber Java dan harus diubah menjadi kode siap eksekusi dengan menggunakan *Java Development Kit* (JDK). Di sini letak keunikan Java yaitu menggunakan kode *byte* yang portabel dan modular. Portabel karena dia bukan kode mesin prosesor (perangkat keras) tertentu, justru sebaliknya dia bisa dimuat

ke berbagai landasan komputer maupun sistem operasi. Dia juga modular karena tiap objek dikompilasi menjadi satu *file* kelas (*class*) yang mandiri. Aplikasi lengkap Java merupakan kumpulan beberapa *file* kelas. *File-file* kelas ini dapat disatukan dan dipadatkan menjadi *file jar (Java Archive*). Pada akhirnya, kode *byte* tersebut akan dijalankan sebagai program oleh *Java Runtime Environment* (JRE). Masing-masing landasan komputer dan sistem operasi, tersedia JRE yang berbeda. JRE inilah yang menyembunyikan landasan dan menyediakan lingkungan yang serupa bagi program Java agar dapat bekerja sebagai mestinya.

Kelebihan utama dari Java adalah dapat dijalankan di beberapa sistem operasi komputer, sesuai dengan prinsip tulis sekali, jalankan di mana saja. Dengan kelebihan ini pemrogram cukup menulis sebuah program Java dan dikompilasi (diubah, dari Bahasa yang dimengerti manusia menjadi Bahasa mesin / *bytecode*) sekali lalu hasilnya dapat dijalankan diatas beberapa sistem operasi tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis Java dikerjakan diatas operating system Linux tetapi dijalankan dengan baik diatas *Microsoft Windows, Platform* yang didukung sampai saat ini adalah *Microsoft Windows, Linux, Mac OS dan Sun Solaris.*

2.1.7. HTML

Proses tampilnya sebuah halaman website di *browser* melibatkan HTML. *HyperText Markup Language* (HTML) tergolong dalam salah satu format yang digunakan dalam pembuatan dokumen yang terbaca oleh *web*.

HTML merupakan bahasa pemrograman yang digunakan untuk mendesain sebuah halaman *web* (Prasetio, 2010:4). HTML merupakan bahasa pemrograman *web* yang memberitahukan peramban *web (web browser)* bagaimana menyusun dan menyajikan konten di halaman *web* (Solichin,

2016:10). HTML adalah bahasa *markup* untuk menyebarkan informasi pada *web* (Simarmata, 2010:52).

Berdasarkan teori dari para ahli di atas, maka *hyptertext markup language* (HTML) merupakan bahasa pemrograman yang dikenal oleh *browser* untuk menampilkan informasi lebih menarik di halaman *web* melalui *web browser.*

2.1.8. Database

Menurut Sutarman (2012:15), "*Database* sekumpulan *file* yang saling berhubungan dan terorganisasi atau kumpulan *record-record* yang menyimpan data dan hubungan diantaranya".

Menurut Ladjamudin (2013:129), "*Database* adalah sekumpulan data store (bisa dalam jumlah yang sangat besar) yang tersimpan dalam *magnetic disk, oftical disk, magnetic drum*, atau media penyimpanan sekunder lainya".

Dari pengertian diatas penulis menyimpulkan *Database* adalah sekumpulan *file* yang saling berhubungan yang menyimpan data dan tersimpan dalam sebuah media penyimpanan.

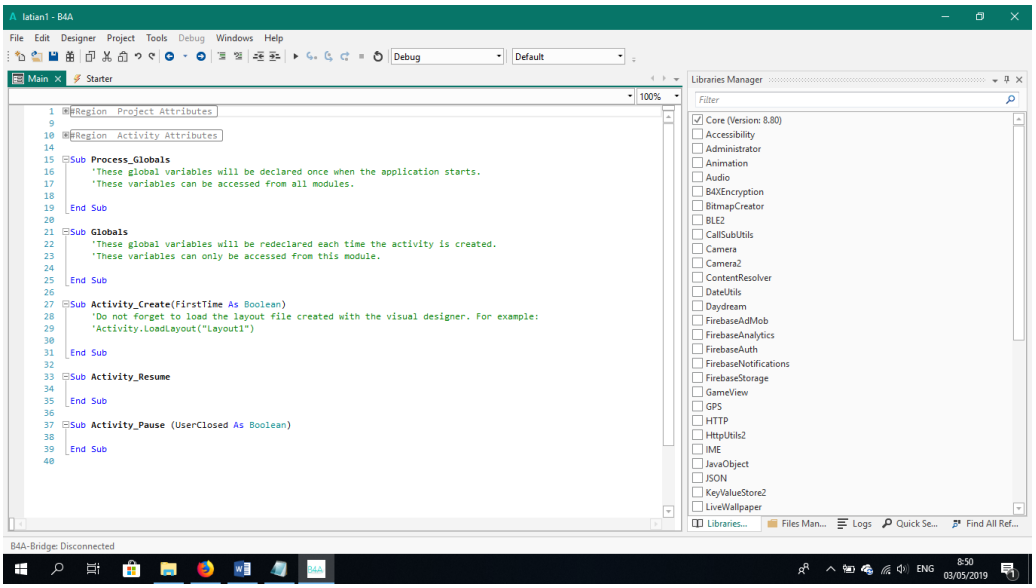
2.2 Teori Pendukung

2.2.1. Basic4Android(B4A)

Basic4android adalah *development tool* sederhana yang *powerful* untuk membangun aplikasi Android. Bahasa *Basic4android* mirip dengan bahasa *Visual Basic* dengan tambahan dukungan untuk objek. Aplikasi ini bisa terhubung melalui *B4Abridge* yang terdapat pada *smartphone* untuk mengetahui hasil dari aplikasi yang dibuat. *Basic4Android* adalah aplikasi yang berfokus pada pengembangan android yang saat ini begitu popular dan berkembang pesat.

Basic4Android memiliki banyak *libraries* yang memudahkan pengembangan aplikasi lanjutan.

Yang termasuk : SQL *databases*, GPS, *Serial ports (Bluetooth)*, Kamera, XML *parsing*, *Web services (* HTTP *)*, Services (*background tasks*), JSON, Animasi, *Network* (TCP dan UDP), *Text To Speech* (TTS), *Voice Recognition*, *WebView*, *AdMob* (ads), *Charts*, *OpenGL*, *Graphics* dan lain-lain. *Basic4Android* bergantung pada dua komponen tambahan (gratis) : Java JDK dan *Android* SDK (Maarif dkk, 2017:38)



Sumber : Aplikasi *Basic4Android*

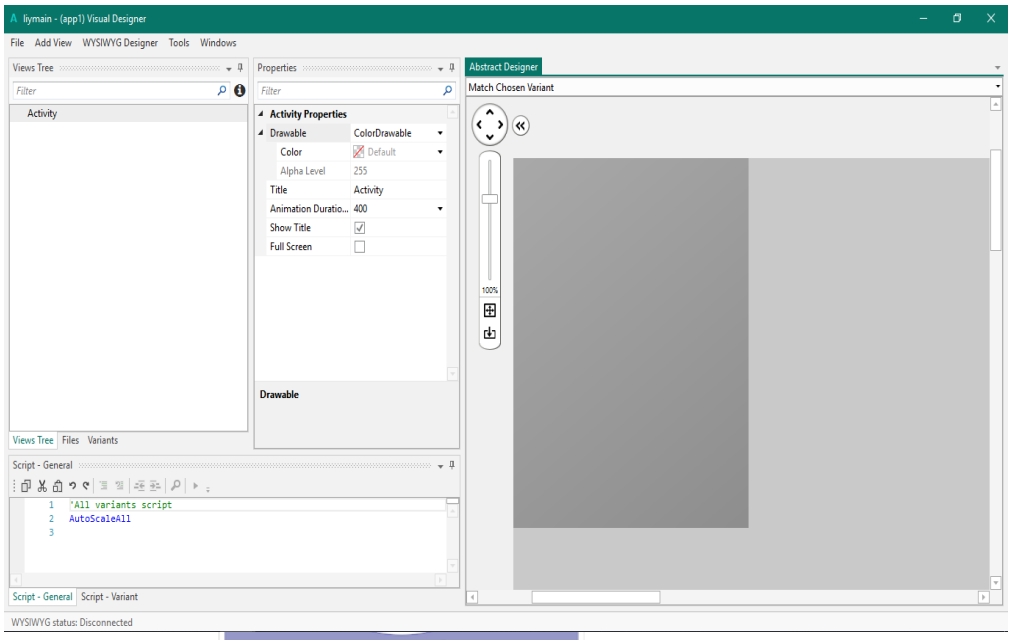
Gambar II.1

Tampilan *Basic4Android*

Fitur yang terdapat di *Basic4Android* antara lain :

a. *Form Visual Designer*

Dalam proses pembuatan desain antarmuka dari suatu aplikasi *form* ini menjadi salah satu hal yang utama karena inilah yang akan banyak digunakan atau dihadapi oleh pengguna aplikasi.

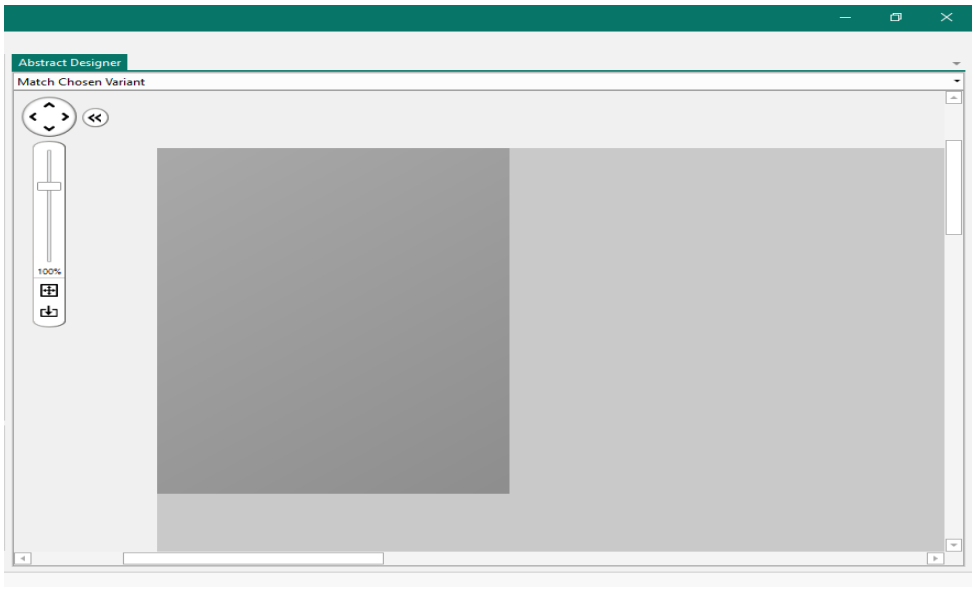


Gambar II.2

Tampilan *Form Visual Designer*

b. *Form Abstract Designer*

Abstract designer berguna untuk mempermudah *programmer* saat mendesain serta merancang tampilan display aplikasi yang sedang dibuat. Form ini membuat *programmer* tahu letak-letak dari tiap objek seperti *button*, *label*, atau *edit text* pada sebuah *layout*.

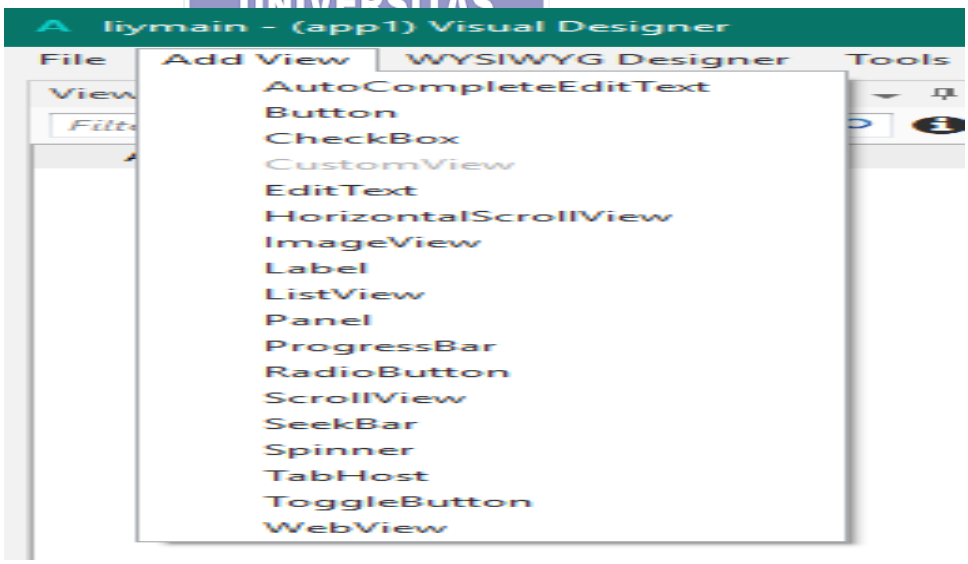


Gambar II.3

Tampilan *Form Abstract Designer*

c. *Form Add View*

Form Add View ini adalah fitur-fitur yang digunakan dalam proses



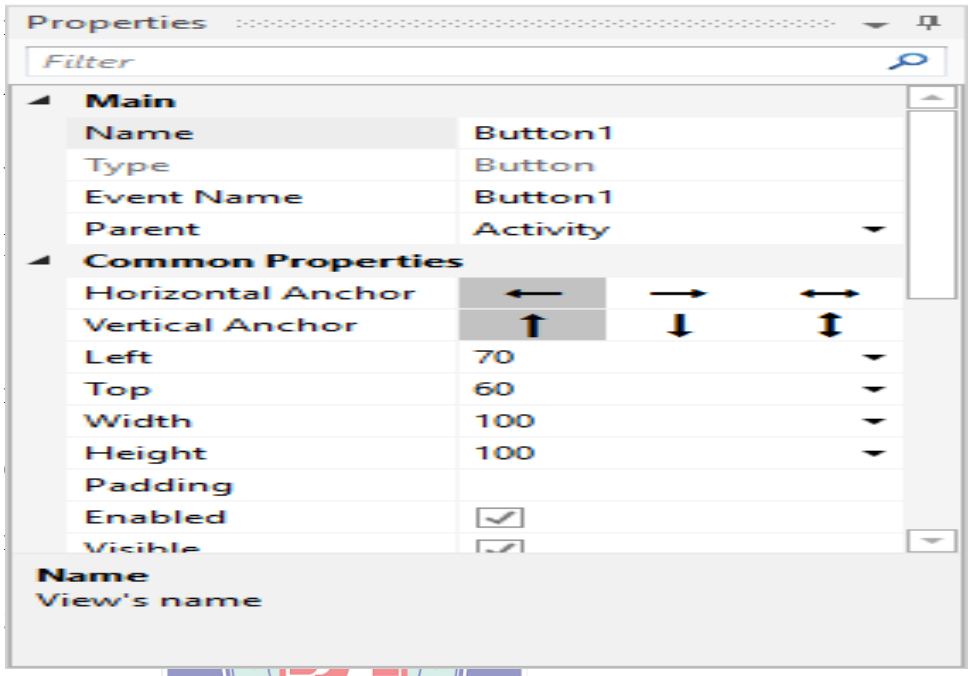
pembuatan *Abstract Designer.*

Gambar II.4

Tampilan *Form Add View*

d. *Form Properties Designer*

Form Properties Designer adalah fitur tambahan yang digunakan u



a

tur tampilan *Add View*

Gambar II.5

Tampilan *Form Properties Designer*

2.2.2. Brackets

Bracket adalah *codeeditor* yang secara khusus dikembangkan untuk tujuan *web design* dan *front-end development* (Herlangga, 2014). Project *Brackets* ini diusung oleh *Adobe* secara *open source* dan dikembangkan secara aktif oleh komunitas *web developer* dan benar-benar dibuat untuk kebutuhan *web development*, khususnya *web design* dan *front-end development*. Kelebihan *Brackets* antara lain:

a. *Live* HTML *Development*

Dapat melihat langsung hasil perubahan kode yang di tulis tanpa harus melakukan save terlebih dahulu.

b. JS *Debugging* dengan *Theseus*

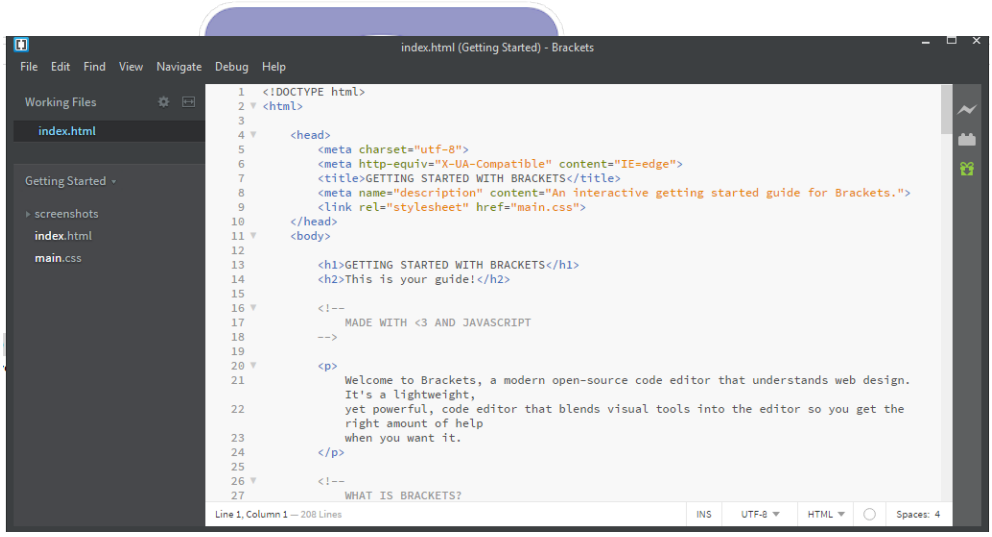
Brackets menggunakan *theuseus* untuk inspeksi dan *debugging* java script-nya.

c. Linux: *New* & *Improved*

Saat *Brackets* telah mengembangkan untuk LINUX.

d. *Effective Development*

Kemampuan yang dapat melakukan *Quick Edit*, sehingga bisa melakukan perubahan *style* dan java *script* tanpa harus berpindah dokumen.



Gambar II.6

Tampilan *Brackets*

2.2.3. SQLite

SQLite adalah perpustakaan perangkat lunak yang bersifat *open source* yang di gunakan untuk menyimpan data pada perangkat yang memiliki memori terbatas (Maarif etal., 2017). SQLite merupakan suatu sistem manajemen *database*, yang mempunyai sifat *ACID-compliant*, yang diprogram dengan

bahasa C, dan mempunyai *size* atau ukuran memori yang relatif kecil. Karena SQLite termasuk *database engine* yang *embedded* (tersemat), jadi perintah SQLite yang bisa digunakan hanya perintah-perintah standar saja. Serta SQLite hanya mendukung tipe data seperti *numeric, datetime,text* dan lain-lain.

Android memiliki fasilitas untuk membuat *database* yang dikenal dengan SQLite, SQLite merupakan salah satu *software* yang *embedded,* kombinasi SQL *interface* dan penggunaan *memory* yang sangat sedikit dengan kecepatan sangat tinggi. SQLite di android termasuk dalam Android *runtime,* sehingga setiap versi dari android dapat membuat *database* dengan SQLite. Dalam sistem android memiliki beberapa teknik untuk melakukan penyimpanan data. Teknik yang umum digunakan adalah sebagai berikut :

a. *Shared Prefences* yaitu menyimpan data beberapa nilai (*value*) dalam bentuk *groups key* yang dikenal dengan *prefences.*
b. *Files* yaitu menyimpan data dalam *file,* dapat berupa menulis ke *file* atau membaca dari *file.*
c. SQLite *Databases*, yaitu menyimpan data dalam bentuk *Database.*
d. *Content Providers*, yaitu menyimpan data dalam bentuk *content providers service.*

Android tidak menyediakan *database* secara otomatis, jika akan menggunakan SQLite maka harus *mengcreate database* sendiri, mendefinisikan tabelnya, index serta datanya. Untuk membuat dan membuka *database* yang paling baik adalah menggunakan *libraries* :

importandroid.database.sqlite.SQLiteOpenHelper. *Libraries* tersebut yang akan menyediakan tiga metode yaitu :

a. *Constructor*, menyediakan representasi versi dari *database* dan skema *database* yang digunakan
b. *OnCreate()*, menyediakan *SQLite database object* yang digunakan dalam definisi tabel dan inisialisasi data
c. *OnUpgrate()*, menyediakan fasilitas konversi *database* dari *database* versi yang lama ke *database* versi yang baru atau sebaliknya.

Ada beberapa keunggulan SQLite, berikut keunggulan SQLite (Yudana, 2017) :

a. SQLite tidak memerlukan proses atau sistem *server* yang terpisah untuk beroperasi (*serverless*)
b. SQLite hadir dengan *zero-configuration*, yang berarti tidak ada *setup* atau administrasi yang dibutuhkan
c. *Database* SQLite yang lengkap disimpan dalam *file* tunggal yang tersimpan dalam *disk* serta bersifat *cross-platform*
d. SQLite bersifat mandiri, yang berarti tidak ada dependensi eksternal.
e. Transaksi SQLite sepenuhnya sesuai dengan ACID, memungkinkan akses yang aman dari banyak proses.
f. SQLite mendukung sebagian besar fitur bahasa query yang ditemukan dalam standar SQL92 (SQL2).
g. SQLite tersedia di semua sistem operasi baik ini *UNIX (Linux, Mac Os-X*, Android, iOS) dan *Windows (Windows 32, Windows CE, Windows RT*).

2.2.4. UML (Unified Modeling Language)

Menurut Windu Gata, Grace (2013:4), *Unified Modeling Language (UML)* adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan

metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem.

Alat bantu yang digunakan dalam perancangan berorientasi objek berbasiskan UML adalah sebagai berikut:

a. *Use Case Diagram*

Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

b. Diagram Aktivitas *(Activity Diagram)*

Activity Diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis.

c. Diagram Urutan *(Sequence Diagram)*

Sequence Diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek.

d. Diagram Kelas (*Class Diagram*)

Merupakan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

e. *Deployment Diagram*

Deployment Diagram digunakan untuk menggambarkan detail bagaimana komponen disusun di infrastruktur sistem.

2.2.5. Hireracy Plus Input-Process_Output (HIPO)

Menurut Mustakini (2014:787) "HIPO (*Hierarchy Input-Proses-Output*) merupakan metodologi yang dikembangkan dan didukung oleh IBM". HIPO sebenarnya adalah alat dokumentasi program. Akan tetapi sekarang, HIPO juga banyak digunakan sebagai alat desain dan teknik dokumentasi dalam siklus pengembangan sistem. HIPO berbasis pada fungsi, yaitu tiap-tiap modul didalam sistem digambarkan oleh fungsi utamanya.

2.2.6. Flowchart

Ladjamudin (2013:211) mengemukakan bahwa, "*flowchart* adalah bagan–bagan yang mempunyai arus yang menggambarkan langkah–langkah penyelesaian suatu masalah". *Flowchart* merupakan cara penyajian dari suatu algoritma.

Berikut ini merupakan simbol-simbol yang dipakai dalam menggambarkan algoritma ke dalam bentuk diagram alir serta kegunaan dari simbol-simbol yang dimaksud yaitu :

a. *Flow Direction Symbols*

   Simbol yang dipakai untuk menghubungkan antara *symbol* satu dengan yang lain.

b. *Processing Symbols*

   Simbol yang menggambarkan jenis operasi pengolahan prosedur.

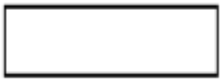
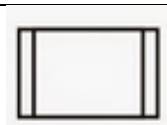
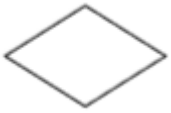
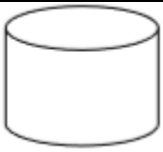
c. *Input Output Symbols*

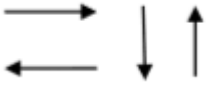
   Simbol yang digunakan untuk menyatakan jenis peralatan yang dipakai sebagai media *input* atau *output.*

Siallagan (2009:6), menjelaskan simbol-simbol dalam *Flowchart* adalah sebagai berikut:

Tabel II.3

Simbol *Flowchart*

No	Simbol	Keterangan
1		Simbol *Start* atau *End* yang mendefinisikan awal atau akhir dari sebuah *flowchart*
2		Simbol pemerosesan yang terjadi pada sebuah alur kerja.
3		Simbol yang menyatakan bagian dari program (sub program).
4		Persiapan yang digunakan untuk memberi nilai awal suatu besaran.
5		Simbol *Input/Output* yang mendefinisikan masukan dan keluaran proses.
6		Menyatakan penyambung ke simbol lain dalam satu halaman.
7		Menyatakan penyambung ke halaman lainnya.
8		Menyatakan pencetakan (dokumen) pada kertas.
9		Menyatakan *desicion* (keputusan) yang digunakan untuk penyeleksian kondisi di dalam program.
10		Menyatakan media penyimpanan drum magnetik.

11		Menyatakan *input*/*output* menggunakan disket.
12		Menyatakan operasi yang dilakukan secara manual.
13		Menyatakan *input*/*output* dari kartu plong.
14		Menyatakan arah aliran pekerjaan (proses).
15		*Delay* (penundaan atau kelambatan).

Sumber : Siallagan, 2009 : 6