

## BAB II

### LANDASAN TEORI

#### 2.1. Konsep Dasar Program

Dalam suatu program atau aplikasi terdapat konsep dasar program untuk memahami apa yang terkandung dalam program tersebut, khususnya program yang berhubungan dengan tugas akhir yaitu sebagai berikut :

##### 2.1.1. *Bates-Jensen Wound Assessment Tools (BJWAT)*

*Diabetes Mellitus* merupakan penyakit kronis yang kompleks dan memerlukan perawatan medis berkelanjutan. Khususnya pada penderita *Diabetes Mellitus* yang mengalami luka luar. Sedikit luka pada penderita diabetes merupakan salah satu hal yang serius. Luka bisa menjadi rumit dan membutuhkan waktu penyembuhan yang lama jika tidak dilakukan perawatan yang baik. Tingkat keparahan luka sangat mempengaruhi lama waktu penyembuhan luka pada penderita *Diabetes Mellitus*. Menurut Harris dalam Rasyid dkk (2018) penilaian terhadap luka kaki diabetes dapat dilakukan dengan menggunakan instrumen pengkajian luka untuk prediksi penyembuhan luka seperti *Bates-Jansen Wound Assessment Tool* dan *The New Diabetic Foot Ulcer Assessment Scale (DFUAS)*.

##### 2.1.2. Program

Menurut Harumy (2016:4), menyatakan bahwa “Program adalah formulasi sebuah algoritma dalam bentuk bahasa pemrograman. Sehingga siap untuk dijalankan pada mesin komputer”

Menurut Sutarman dalam Maarif, dkk. (2017:37), menyatakan bahwa Program adalah:

barisan perintah atau instruksi yang disusun sehingga dapat dipahami oleh komputer dan kemudian dijalankan sebagai barisan perhitungan numerik, dimana barisan perintah tersebut berhingga, berakhir, dan menghasilkan output.

Berdasarkan pemaparan para ahli diatas, program adalah formulasi sebuah algoritma dalam bentuk bahasa pemrograman yang disusun dari barisan perintah atau intruksi yang dapat dipahami oleh komputer. Barisan perintah tersebut berhingga ,berakhir,dan menghasilkan *output*.

### **2.1.3. Bahasa Pemrograman**

Menurut Suprpto dalam Maarif, dkk. (2017:37), “Bahasa pemrograman adalah notasi yang digunakan untuk menulis program (komputer). Bahasa ini dibagi menjadi tiga tingkatan yaitu bahasa mesin, bahasa tingkat rendah dan bahasa tingkat tinggi”.

Bahasa pemrograman yang lebih mudah dipelajari adalah bahasa pemrograman aras tinggi. Disebut aras tinggi karena bahasanya mendekati level bahasa manusia sehingga lebih mudah dipahami. Beberapa jenis bahasa pemrograman aras tinggi yang biasa digunakan antara lain: *Java, Android, HTML (Hypertext Markup Language)*.

### **2.1.4. Sistem Pakar**

#### **1. Sistem**

Menurut Sujarweni (2015:1) mengatakan bahwa “Sistem adalah kumpulan elemen-elemen yang saling berkaitan dan bekerja sama dalam melakukan kegiatan untuk mencapai suatu tujuan”.

Klasifikasi suatu sistem menurut Hutahaean (2015:6) adalah :

1. Sistem abstrak (*abstract system*)

Sistem abstrak adalah sistem yang berupa pemikiran-pemikiran atau ide yang tidak tampak secara fisik.

2. Sistem fisik (*physical system*)

Sistem fisik adalah sistem yang ada secara fisik.

3. Sistem alamiyah (*natural system*)

Sistem alamiyah adalah sistem yang terjadi melalui proses alam, tidak dibuat oleh manusia. Misalnya sistem perputaran bumi.

4. Sistem buatan manusia (*human made system*)

Sistem buatan manusia adalah sistem yang dibuat oleh manusia yang melibatkan interaksi antara manusia dengan mesin (*human machine system*).

5. Sistem tertentu (*deterministic system*)

Sistem tertentu adalah sistem yang beroperasi dengan tingkah laku yang sudah dapat diprediksi, sebagai keluaran sistem yang dapat diramalkan atau disimpulkan hasilnya.

6. Sistem tak tertentu (*probabilistic system*)

Sistem tak tertentu adalah sistem yang kondisi masa depannya tidak dapat diprediksi karena mengandung unsur probalistik.

7. Sistem tertutup (*close system*)

Sistem tertutup adalah sistem yang tidak dapat terpengaruh dan tidak berhubungan dengan lingkungan luar, sistem bekerja secara otomatis tanpa ada turut campur lingkungan luar. Secara teoritis sistem tertutup ini ada, kenyataannya tidak ada sistem yang benar-benar tertutup, yang ada hanya *relatively closed system*.

## 8. Sistem terbuka (*open system*)

Sistem terbuka adalah sistem yang berhubungan dan terpengaruh dengan lingkungan luarnya. Sistem ini menerima *input* dan *output* dari lingkungan luar atau subsistem lainnya. Karena sistem terbuka terpengaruh lingkungan luar maka harus mempunyai pengendalian yang baik.

## 2. Pakar

Menurut Rosnelly dalam Priandi, dkk (2019:8), “Pakar adalah seorang individu yang memiliki pengetahuan khusus, pemahaman, dan metode-metode yang digunakan untuk memecahkan persoalan dalam bidang tertentu”.

## 3. Sistem Pakar

Menurut Rahmi Ras, F., Nelly Astuti, H., & Efori, B. (2017) mengatakan bahwa “Sistem pakar adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer yang dirancang untuk memodelkan kemampuan menyelesaikan masalah seperti layaknya seorang pakar”. Dari pengertian tersebut penulis berpendapat bahwa sistem pakar yaitu suatu sistem yang didalamnya berisi kecerdasan seorang pakar yang dipindahkan kedalam suatu sistem yang nantinya akan diolah oleh suatu program agar dapat dapat digunakan dalam berbagai bidang. Program sistem pakar sangat membantu untuk mengambil keputusan karena penerapannya sangat luas.

Sistem pakar memungkinkan seseorang yang kurang memahami suatu permasalahan terbantu melalui aplikasi sistem pakar yang dibuat. Permasalahan yang telah diidentifikasi dapat diproses dan diolah sedemikian rupa sehingga dihasilkan suatu kesimpulan mengenai suatu permasalahan.



## 1. Manfaat Sistem Pakar

Menurut Sutopo dalam (Hayadi et al. 2017) menyatakan bahwa “sistem pakar menjadi sangat populer karena sangat banyak kemampuan dan manfaat yang diberikannya”. Sebagai contoh manfaatnya antara lain:

- a. Meningkatkan produktivitas, karena sistem pakar dapat bekerja lebih cepat daripada manusia.
- b. Membuat seseorang yang awam bekerja seperti layaknya seorang pakar.
- c. Meningkatkan kualitas, dengan memberi nasehat yang konsisten dan mengurangi kesalahan.
- d. Mampu menangkap pengetahuan dan kepakaran seseorang.
- e. Memudahkan akses pengetahuan seorang pakar.
- f. Bisa digunakan sebagai media pelengkap dalam pelatihan. Pengguna pemula yang bekerja dengan sistem pakar akan menjadi lebih berpengalaman karena adanya fasilitas penjelas yang berfungsi sebagai guru.

## 2. Kekurangan Sistem Pakar

Menurut Hayadi B, Herawan (2016:3) selain manfaat, ada juga kekurangan yang ada pada sistem pakar, diantaranya:

- a. Biaya yang sangat mahal untuk membuat dan memeliharanya.
- b. Sulit dikembangkan karena keterbatasan keahlian dan ketersediaan pakar.
- c. Sistem pakar tidak 100% bernilai benar.

## 3. Ciri-ciri sistem pakar

Menurut Hayadi B, Herawan (2016:3) Sistem pakar mempunyai ciri-ciri, diantaranya:

- a. Terbatas pada domain keahlian tertentu

- b. Dapat memberikan penalaran untuk data yang tidak pasti
- c. Dapat mengemukakan rangkaian alasan yang diberikannya dengan cara yang dapat dipahami.
- d. Berdasarkan pada kaidah atau rule tertentu.
- e. Dirancang untuk dapat dikembangkan secara bertahap.
- f. Pengetahuan dan mekanisme inferensi jelas terpisah.
- g. Keluarannya bersifat anjuran.
- h. Sistem dapat mengaktifkan kaidah secara searah yang sesuai yang dituntun oleh dialog dengan pemakai.

#### 4. Karakteristik Sistem Pakar

Menurut Budhiartha dan Suhartono (2014), Sistem pakar banyak digunakan pada aplikasi terkini dan kompleks karena :

- a. Sistem pakar dapat bertindak sebagai konsultan, instruksi, atau pasangan/rekan.
- b. Meningkatkan *availability* atau kepakaran tersedia pada semua perangkat komputer.
- c. Mengurangi bahaya.
- d. Permanen.
- e. Pengetahuan dapat tidak lengkap, namun keahlian dapat diperluas sesuai kebutuhan. Program konvensional “lengkap” sebelum mereka dapat digunakan.
- f. *Database* yang cerdas, sistem pakar dapat digunakan untuk mengakses *database* secara cerdas, misalnya *data mining*. Sistem pakar biasanya di desain untuk memiliki karakteristik sebagai berikut:
  - a. *High performance*.
  - b. *Adequate response time*.



c. *Good reliability.*

d. *Understandable.*

### 2.1.5. Aplikasi

Menurut Alfeno, dkk (2017:28), “Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut”. aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpacu pada sebuah komputasi yang diinginkan atau diharapkan.

### 2.1.6. Android

Menurut Satyaputra dan Aritonang (2014:4), “*Android* merupakan sistem operasi yang bersifat *open source* (sumber terbuka). Disebut *open source* karena *source code* (kode sumber) dari sistem operasi *Android* dapat dilihat, di-download, dan dimodifikasi secara bebas”.



Menurut Safaat (2015:1), “*Android* adalah Sebuah kumpulan perangkat lunak untuk perangkat *mobile* yang mencakup sistem operasi, *middleware* dan aplikasi utama *mobile*.” Namun, seiring perkembangannya, *Android* berubah menjadi *platform* yang begitu cepat dalam melakukan inovasi. Hal ini tidak lepas dari pengembangan utama dibelakangnya yaitu *google*. *Google* lah yang mengakuisi *android*, yang kemudian membuatkan sebuah *platform*.

#### 1. Sejarah *Android*

Pada awal mulanya, *Android Inc* merupakan sebuah perusahaan *software* kecil yang didirikan pada bulan Oktober 2003 di Palo Alto, California, USA. Didirikan oleh beberapa senior di beberapa perusahaan yang berbasis *IT & Communication*;

Andy Rubun, Nick Sears, dan Chirs White. *Android Inc* didirikan untuk mewujudkan mobile device yang lebih peka terhadap lokasi dan prefensi pemilik. Dengan kata lain, *Android Inc* ingin mewujudkan *mobile device* yang lebih mengerti pemiliknya.

## 2. Perkembangan Versi *OS Android*

Keunikan dari nama sistem operasi (*OS Android*) adalah dengan menggunakan nama makanan hidangan penutup (*Dessert*). Selain itu juga nama-nama *OS Android* memiliki huruf awal berurutan sesuai abjad. Alasan *Android* merilis versinya dengan nama makanan ringan dengan tujuan agar pengguna dapat dengan mudah mengingatnya. Berikut merupakan nama-nama versi *Android* yang telah dirilis hingga saat ini :

### a. *Android Beta*

Pertama kali dirilis pada 5 November 2007, kemudian pada 12 November 2007 *Software Development Kit (SDK)* dirilis oleh *Google*.

### b. *Android 1.0 Astro*

Pertama kali dirilis pada 23 November 2008. Sebenarnya *Android* versi pertama ini akan dinamai dengan nama "*Astro*" tidak jadi disematkan pada versi pertama dari *OS Android* ini. *HTC dream* adalah ponsel pertama menggunakan *OS* ini.

### c. *Android 1.1 Bender*

Pertama kali dirilis pada 9 Februari 2009. Versi *Android* kedua ini juga mengalami masalah penamaan yang sama dengan versi pertamanya. Pada awalnya akan diberi nama "*Bender*", akan tetapi karena alasan melanggar *trade mark*, nama "*Bender*" tidak jadi disematkan pada versi android ini. Awalnya versi *OS Android* ini dirilis untuk perangkat *T-mobile G1* saja. Versi



ini merupakan update untuk memperbaiki beberapa bugs, mengganti API dan menambahkan beberapa fitur.

d. *Android 1.5 Cupcake*

Pertama kali dirilis pada 30 April 2009. Pada penamaan versi *Android* tersebut perusahaan menggunakan nama pencuci mulut (*desert*) mulai digunakan, karena ini merupakan versi ketiga maka penamaan tersebut diawali dengan huruf “C” dan jadilah “*Cupcake*” menjadi nama resmi dari *OS Android* ketiga ini. *OS* ini berbasis pada kernel *Linux 2.6.27* dan menambahkan beberapa update serta UI baru dari versi *Android* sebelumnya. Mulai terdapat “*widget*” yang dapat mengubah ukuran sesuai keinginan. Kemudian terdapat fitur mengupload sebuah video atau gambar.

e. *Android 1.6 Donut*

Dirilis pertama kali pada 15 September 2009. Terdapat peningkatan pada fitur pencarian dan UI yang lebih *user friendly*. Pada versi *OS* sudah mendukung teknologi *CDMA/EVDO*, *802.1x*, *VPNs*. Kemudian mendukung layar dengan resolusi *WVGA*.

f. *Android 2.0/2.1 Eclair*

Dirilis pertama kali pada 9 Desember 2009. Terjadi penambahan fitur untuk pengoptimalan *hardware*, peningkatan *Google maps 3.1.2*, perubahan UI dengan *browser* baru dan dukungan *HTML5*, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, *digital zoom*, dan *bluetooth 2.1*, beberapa versi updatenya antara *Android v.2.0* kemudian *v2.02* dan kemudian yang terakhir *v.2.1*.

g. *Android 2.2 Froyo (Froze Yoghurt)*

Dirilis pertama kali pada 20 Mei 2010 pada *smartphone Google Nexus One*. Pada versi ini sudah mendukung terhadap *Adobe Flash Player 10.1*. Peningkatan pada kecepatan membuka dan menutup aplikasi, serta penggunaan *SD Card* sebagai tempat penyimpanan aplikasi. Ketika *Android Froyo* hadir mulai muncul banyak diskusi yang membahas mengenai persaingan antara *Android* dengan *iOS* yang akan semakin ketat di masa yang akan datang. Beberapa versi update yang dirilis antara lain *Android v2.2.1* hingga *v.2.2.3*.

h. *Android 2.3 Gingerbread*

Pertama kali diperkenalkan pada 6 Desember 2010. Terjadi banyak peningkatan pada versi *Android* yang satu ini dibandingkan dengan versi sebelumnya. Dirancang untuk memaksimalkan kemampuan aplikasi dan *game*. Serta mulai digunakan layar resolusi *WXGA* dan di atasnya. Beberapa versi *update* yang dirilis antara lain *v.2.3.3* hingga *v.2.3.7*. Sampai saat ini *Android Gingerbread* merupakan versi *Android* yang memiliki pengguna terbanyak dibandingkan dengan seri *Android* lainnya, yaitu mencapai 65% dari seluruh versi *Android* yang dirilis.

i. *Android 3.0/3.1 Honeycomb*

Pertama kali diperkenalkan pada 22 Februari 2011 dan *Motorola Xoom* adalah yang pertama kali menggunakannya. *Android* versi ini merupakan *OS* yang didesain khusus untuk pengoptimalan penggunaan pada tablet *PC*.

j. *Android 4.0 ICS (Ice Cream Sandwich)*

Pertama kali dirilis pada 19 Oktober 2011. *Smartphone* yang pertama kali menggunakan *OS Android* ini adalah *Samsung Galaxy Nexus*. Secara teori

semua perangkat yang menggunakan versi *Android* sebelumnya, *Gingerbread*, dapat diupdate ke *Android Ice Cream Sandwich*.

k. *Android versi 4.1 (Jelly Bean)*

*Android Jelly Bean* yang diluncurkan pada acara *Google I/O* lalu membawa sejumlah keunggulan dan fitur baru. Penambahan baru diantaranya meningkatkan input keyboard, desain baru fitur pencarian, UI yang baru dan pencarian melalui *voice search* yang lebih cepat. Tak ketinggalan *Google Now* juga menjadi bagian yang diperbarui. *Google Now* memberikan informasi yang tepat pada waktu yang tepat pula. Salah satu kemampuannya adalah dapat mengetahui informasi cuaca, lalu lintas ataupun hasil pertandingan olahraga. Sistem operasi *Android Jelly Bean 4.1* muncul pertama kali dalam produk tablet *Asus*, yakni *Google Nexus 7*.

l. *Android versi 4.2 (Jelly Bean)*

Fitur *photo sphere* untuk panorama, *daydream* sebagai *screensaver*, *power control*, *lock screen widget*, menjalankan banyak *user* (dalam tablet saja), *widget* terbaru. *Android 4.2* pertama kali dikenalkan melalui *LG Google Nexus 4*.

m. *Android versi 4.4 (KitKat)*

Berbagai fitur yang disediakan oleh *OS Android Kitkat* ini salah satunya adalah perbaikan sistem penyimpanan sementara pada penggunaan memori, yang mana kinerja prosesor telah diminimalisir terhadap penyimpanan *registry* data sementara pada *RAM* dan secara langsung akan ditampung oleh kapasitas memori internal yang tersedia, sehingga loading prosesor akan terasa lebih ringan.

n. *Android versi 5.0-5.1 (Lollipop)*

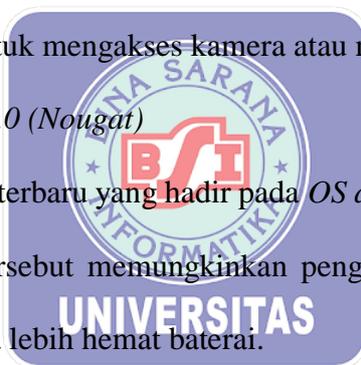
Diresmikan pada tanggal 25 Juni 2014. *Android* versi 5.0-5.1 ini memiliki beberapa perubahan, salah satunya adalah *user interface* yang didesain ulang dan dibangun dengan yang dalam bahasa desain disebut sebagai *material design*. Perubahan lain termasuk perbaikan pemberitahuan, yang dapat diakses dari *lock screen* dan tampilan pada banner dibagian atas *screen*.

o. *Android versi 6.0 (Marshmallow)*

Sebuah sistem *opt-in* sekarang digunakan pada *Android* versi 6.0 ini. Dimana pengguna akan diminta untuk memberikan atau menolak izin individu, seperti kemampuan untuk mengakses kamera atau mikrofon.

p. *Android versi 7.0 (Nougat)*

Salah satu fitur terbaru yang hadir pada *OS android Nougat* ini adalah *Project Svalte*. Fitur tersebut memungkinkan penggunaan kinerja *RAM* yang lebih minim sehingga lebih hemat baterai.



q. *Android versi 8.0 (Oreo)*

Sistem operasi ini dirilis secara stabil mulai Agustus 2017 dan sudah mengalami pembaruan lewat versi *Android 8.1 Oreo* yang terbaru. Sistem operasi ini menawarkan pengalaman *multitasking* yang makin mumpuni dibanding versi sebelumnya. Selain itu ada juga *Project Treble* yang memungkinkan pengguna mendapat pembaruan lebih cepat.

r. *Android 9.0 (Pie)*

Terakhir ada *Android 9.0 Pie* yang secara resmi diperkenalkan pada Agustus 2018. Sistem operasi *Android* ini memberi banyak perubahan, terutama untuk HP dengan desain baru yaitu *navigasi* berupa *gesture* yang menggantikan

tombol fisik *Home*, *Back* dan *Recent Apps*. Fitur lainnya yang cukup berguna adalah sistem notifikasi, pengatur kecerahan hingga sistem *screenshot* terbaru.

### 2.1.7. *Java*

Menurut Utomo dalam Maarif, dkk. (2018:93) mengemukakan bahwa:

*Java* merupakan salah satu bahasa pemrograman yang bersifat *multiplatform* dengan slogan dari para pengembangnya adalah '*Write one run everywhere*' sehingga aplikasi yang dikembangkan menggunakan bahasa java akan dapat dijalankan pada berbagai macam *platform* atau sistem operasi.

*Java* adalah bahasa pemrograman berorientasi objek murni yang dibuat berdasarkan kemampuan-kemampuan terbaik bahasa pemrograman objek sebelumnya seperti C++, Ada, simula. Bahasa pemrograman *Java* yang pada awalnya bernama bahasa pemrograman OAK dipelopori oleh James Gosling, Patrick Naughton, Chris Warth, Ed Frank dan Mike Sheridan dari *Sun Microsystem.Inc* pada tahun 1991, ketika perusahaan tersebut memulai *Green Project* (proyek penelitian membuat bahasa pemrograman yang dapat berjalan diatas berbagai *platform*). Karakteristik *java* antara lain:

1. Bersifat *multiplatform*
2. Berorientasi Objek
3. Terdistribusi
4. Bersifat *multithread*

### 2.1.8. *HTML*

Menurut Madcoms (2016:33) "*HTML (Hypertext markup language)* adalah suatu bahasa yang digunakan untuk membuat sebuah halaman *web*". *HTML* itu bahasa yang fleksibel karena tidak tergantung pada suatu *platform* (sistem operasi) tertentu. *HTML* terdiri dari *tag-tag* yang mendefinisikan elemen tertentu pada sebuah halaman *web*.

*HTML* merupakan bahasa yang tidak *case sensitive*, tidak seperti bahasa pemrograman *server-side* seperti *PHP* dan *ASP*. *HTML* bisa disebut bahasa yang digunakan untuk menampilkan dan mengelola *hypertext*.

### 2.1.9. Database

Menurut Connolly dan Begg (2015:63), “*Database* adalah kumpulan dari data-data yang berhubungan secara logika dan deskripsi dari data yang dirancang untuk memenuhi kebutuhan informasi sebuah organisme”.

Menurut A.S, R., & Shalahuddin, M. (2015) “Basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan”. Berdasarkan teori tersebut, dapat disimpulkan bahwa basis data adalah suatu pemeliharaan data yang sudah diolah dan membuat informasi yang dibutuhkan serta menambah, mengakses, dan memproses data yang disimpan diperlukan sistem manajemen basis data (*DBMS*).

## 2.2. Peralatan Pendukung (*Tools System*)

Peralatan pendukung (*Tools System*) adalah alat yang digunakan untuk menggambarkan bentuk logika model dari suatu sistem dengan menggunakan simbol-simbol, lambang-lambang, diagram-diagram yang menunjukkan secara tepat arti dan fungsinya.

### 2.2.1. Basic For Android (B4A)

*Basic4Android* merupakan sebuah *development tool* sederhana dan *powerful* yang digunakan untuk membangun aplikasi *Android* dengan bahasa yang mirip dengan *Visual Basic*. Pada *Basic4Android*, aplikasi *Android* (APK) yang di compile adalah aplikasi *Android nativ/asli* dan tidak ada ketergantungan dengan *file msvbvm60.dll* seperti *Visual Basic*. Dengan kata lain aplikasi *B4A* yang di *compile* bersifat *NO*

*Dependencies* (tidak tergantung pada file lain). *B4A* mencakup semua fitur yang dibutuhkan untuk cepat mengembangkan jenis aplikasi *Android*. *B4A* digunakan oleh puluhan ribu pengembang dari seluruh dunia, termasuk perusahaan seperti *NASA*, *HP*, *IBM* dan lain-lain. *Basic4Android* termasuk *designer GUI* untuk aplikasi *Android* dengan dukungan *Built-in* untuk *multiple screens* dan *orientations*, dan tidak dibutuhkan lagi penulisan *XML* yang rumit. *Basic4Android* memiliki *libraries* (perpustakaan) yang membuatnya menjadi lebih mudah untuk mengembangkan macam-macam aplikasi *Android* yang *advance*.

A. *Library*-nya adalah:

1. *SQL databases*
2. *GPS*
3. *Serial ports (Bluetooth)*
4. *Camera*
5. *XML parsing*
6. *Web services (HTTP)*
7. *Services (background tasks)*
8. *JSON*
9. *Animations*
10. *Network (TCP and UDP)*
11. *Text To Speech (TTS)*
12. *Voice Recognition*
13. *WebView*
14. *AdMob (ads)*
15. *Charts*
16. *OpenGL*



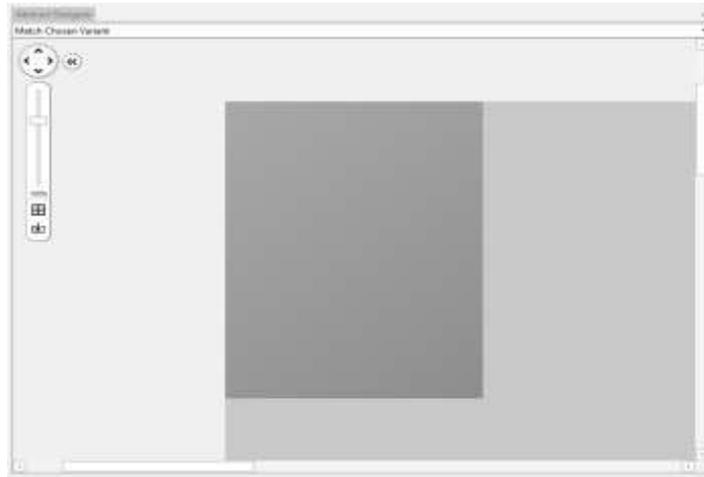


Sumber : Aplikasi B4A

Gambar II.2 Tampilan *Visual Designer*

### 3. *Form Abstract Design*

Proses desain antarmuka yang terhubung dengan perangkat *emulator*/perangkat *smartphone*(B4A Designer).

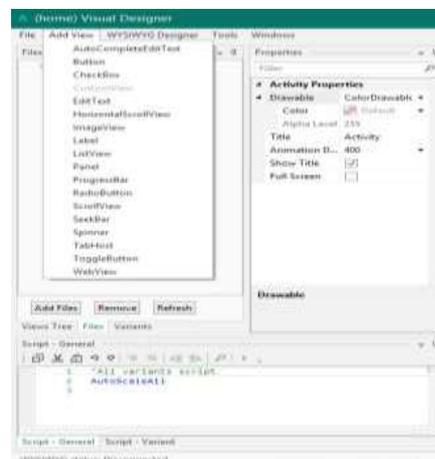


Sumber : Aplikasi B4A

Gambar II.3 Tampilan *Abstract Designer*

### 4. *Form Toolbar Add View*

Fitur-fitur yang dibutuhkan untuk membuat antarmuka terdapat dalam *toolbar Add View*.



Sumber : Aplikasi B4A

Gambar II.4 Tampilan *Add View*

## 5. *Form Properties Designer*

Tentunya dalam pembuatan antarmuka membutuhkan adanya perubahan tampilan sesuai yang diinginkan. *Pada form properties* inilah tempatnya.

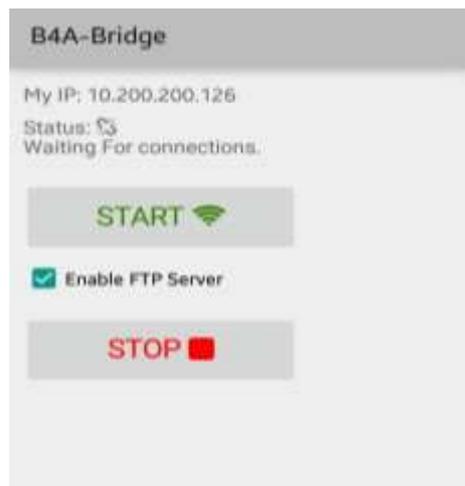


Sumber : Aplikasi B4A

Gambar II.5 Tampilan *Properties* desain *Add View*

### 2.2.2. *B4A Bridge*

*B4A Bridge* merupakan perangkat yang digunakan untuk menghubungkan program yang telah dibuat dengan perangkat *android*. *B4A Bridge* terbuat dari dua komponen, satu komponen berjalan pada perangkat dan komponen kedua (yang merupakan bagian dari IDE) untuk terhubung dan berkomunikasi dengan perangkat. Koneksi dilakukan menggunakan jaringan *nirkabel* lokal. *B4A* mendukung semua fitur IDE yang meliputi pemasangan aplikasinya, melihat *log*, *debugging* dan perancangan *visual* (mengambil tangkapan layar tidak didukung). *Android* tidak memungkinkan aplikasi untuk diam-diam mengunduh aplikasi lain. Oleh karena itu, ketika menjalankan aplikasi dengan menggunakan *B4A Bridge* akan melihat dialog yang meminta persetujuan instalasi. (Seagrave, 2015:48)

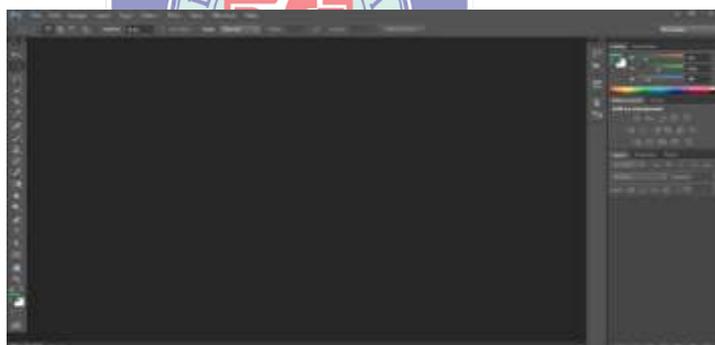


Sumber : B4A Bridge

Gambar II.6 Tampilan B4A Bridge

### 2.2.3. Adobe Photoshop

*Adobe Photoshop* adalah perangkat lunak editor citra buatan *Adobe Systems* yang dikhususkan untuk pengeditan foto/gambar dan pembuatan efek.



Sumber : *Adobe Photoshop CS 6*

Gambar II.7 Tampilan *Adobe Photoshop CS6*

Perangkat lunak ini banyak digunakan oleh fotografer digital dan perusahaan iklan. Penggunaan *Photoshop* untuk pembuatan gambar antarmuka ataupun tombol juga bisa digunakan untuk memperindah tampilan.

### 2.2.4. Brackets

Menurut (Herlangga, 2014), “*Brackets* adalah *code editor* yang secara khusus dikembangkan untuk tujuan *web design* dan *front-end development*”. 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*.



Sumber : *Brackets*



Gambar II.8 Tampilan *Brackets*

Kelebihan *Brackets* antara lain :

1. *Live HTML Development*

Dapat melihat langsung hasil perubahan kode yang ditulis tanpa harus melakukan *save* terlebih dahulu.

2. *JS Debugging dengan Theseus*

*Brackets* menggunakan *theseus* untuk inspeksi dan *debugging javascript*-nya.

3. *Linux : New & Improved*

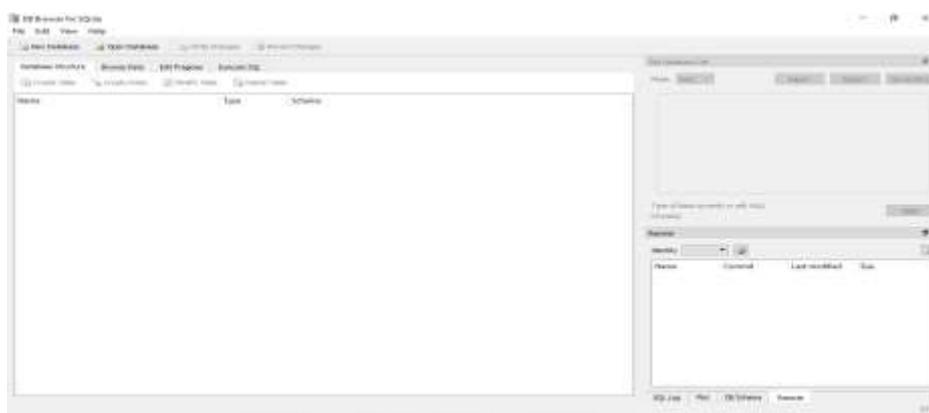
Saat *Brackets* telah mengembangkan untuk *LINUX*.

4. *Effective Development*

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

### 2.2.5. SQLite

*SQLite* adalah *embedded database* yang *open source*. Artinya *SQLite* merupakan *DBMS* dimana pengguna dapat menggunakannya dengan leluasa tanpa harus membayar untuk mendapatkannya. pengguna dapat mengambil *source code* -nya secara gratis. sebagai *embedded database*, maka pengguna tidak perlu melakukan instalasi untuk menggunakan *SQLite* (no-konfigurasi).



Sumber : SQLite

Gambar II.9 Tampilan SQLite

Berikut ketentuan lainnya terkait *SQLite*:

1. *SQLite* dibangun menggunakan bahasa C dengan menyediakan *API* sederhana;
2. *SQLite* tidak mandiri proses seperti *DBMS* pada umumnya. artinya kita dapat membuat link statis maupun dinamis sesuai kebutuhan aplikasi yang kita buat; serverless (tidak memerlukan server untuk mengoperasikannya);
3. *SQLite* disimpan dalam penyimpanan tunggal dan *multiplatform* (dapat digunakan pada semua OS).

### 2.2.6. Android SDK

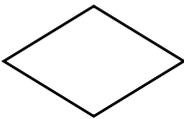
*Android SDK* adalah *tools* bagi para *programmer* yang ingin mengembangkan aplikasi berbasis *Google Android*. *Android SDK* mencakup seperangkat alat

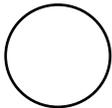
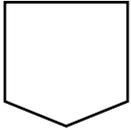
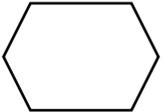
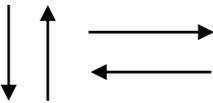
pengembangan yang kompherensif. *Android SDK* terdiri dari *debugger*, *libraries*, *handset emulator*, dokumentasi, contoh kode, dan tutorial.

### 2.2.7. Flowchart

Menurut Sitorus (2015:14), mengemukakan bahwa “*Flowchart* menggambarkan urutan logika dari suatu prosedur pemecahan masalah, sehingga *Flowchart* merupakan langkah-langkah penyelesaian masalah yang ditulis dalam simbol-simbol tertentu”.

Tabel II.1. Simbol-simbol *Flowchart*

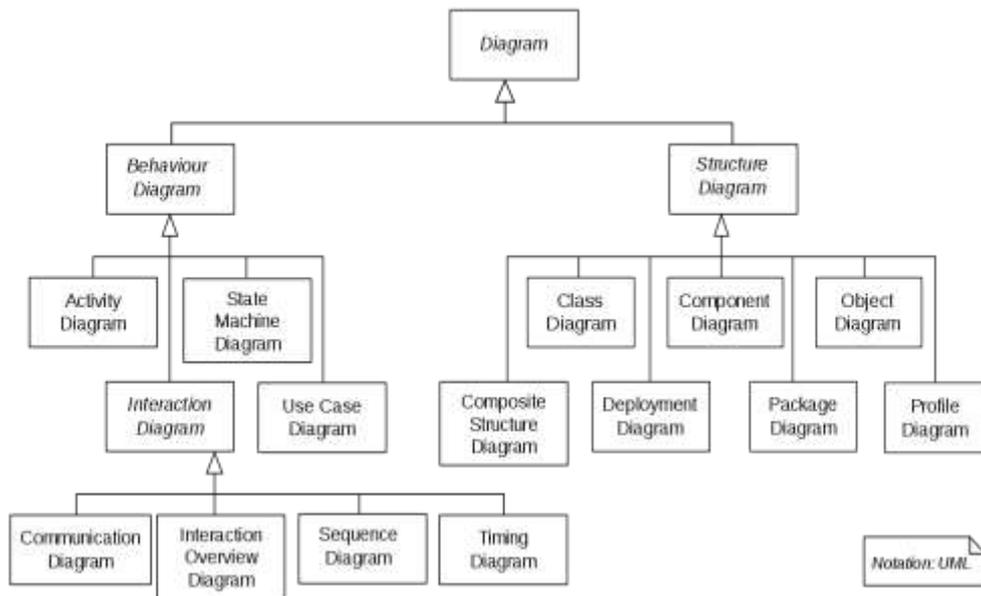
NO.	Simbol	Nama	Fungsi
1.		Terminal	Menyatakan permulaan atau akhir suatu program.
2.		Input/Output	Menyatakan proses <i>input</i> atau <i>output</i> tanpa tergantung jenis peralatannya.
3.		Process	Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer.
4.		Decision	Menunjukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya atau tidak.

5.		<i>Connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama.
6.		<i>Offline Connector</i>	Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang berbeda.
7.		<i>Predefined Process</i>	Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal.
8.		<i>Fancher Card</i>	Menyatakan <i>input</i> berasal dari kartu atau <i>output</i> data ke kartu.
9.		<i>Punch Tape</i>	Mencetak melalui <i>paper tape</i> /bisa juga digunakan untuk <i>symbol</i> uang.
10.		<i>Document</i>	Mencetak keluaran dalam bentuk dokumen (melalui printer).
11.		<i>Flow</i>	Menyatakan jalannya arus suatu proses.

Sumber : Sitorus (2015:15)

### 2.2.8. UML (*Unified Modeling Language*)

Menurut Rosa dalam Afrizal (2017:4) UML merupakan bahasa pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML mendefinisikan beberapa jenis digram resmi, diantaranya: *use case diagram*, *class diagram*, *sequence diagram*.

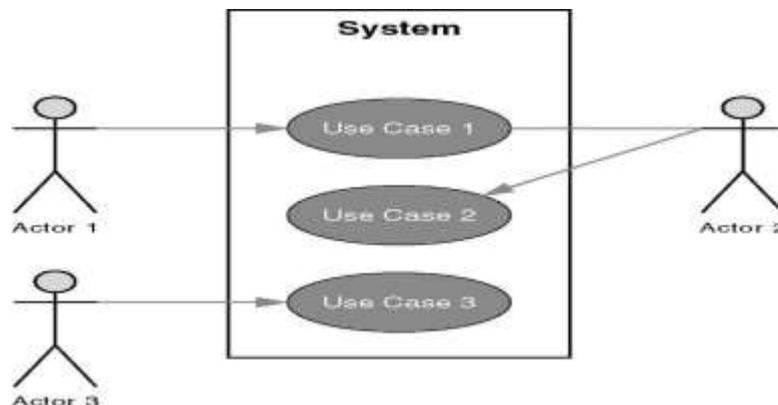


Sumber : Codepolitan.com

Gambar II.10 UML (*Unified Modeling Language*)

### 1. *Use case diagram*

Diagram yang menggambarkan actor, use case dan relasinya sebagai suatu urutan tindakan yang memberikan nilai terukur untuk aktor. Sebuah use case digambarkan sebagai elips horizontal dalam suatu diagram UML *use case*



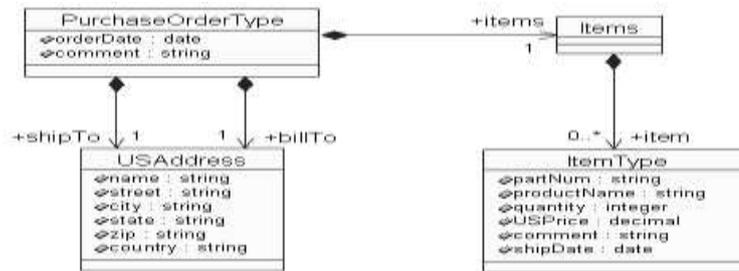
Sumber : Ropianto (2016:47)

Gambar II.11 *Use case diagram*

### 2. *Class diagram*

Class diagram menggambarkan struktur statis dari kelas dalam sistem anda dan

menggambarkan atribut, operasi dan hubungan antara kelas. Class diagram membantu dalam memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. Selama tahap desain, class diagram berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat.

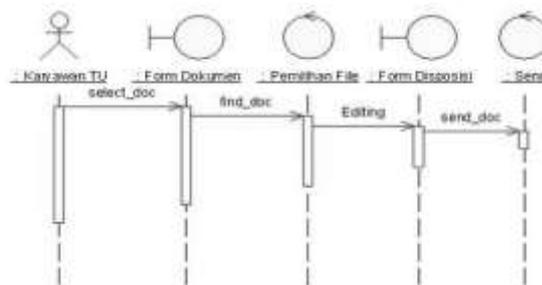


Sumber : Ropianto (2016:45)

Gambar II.12 Class diagram

### 3. sequence diagram

*Sequence* diagram menjelaskan interaksi objek yang disusun berdasarkan urutan waktu. Secara mudahnya *sequence* diagram adalah gambaran tahap demi tahap, termasuk kronologi (urutan) perubahan secara logis yang seharusnya dilakukan untuk menghasilkan sesuatu sesuai dengan use case diagram.

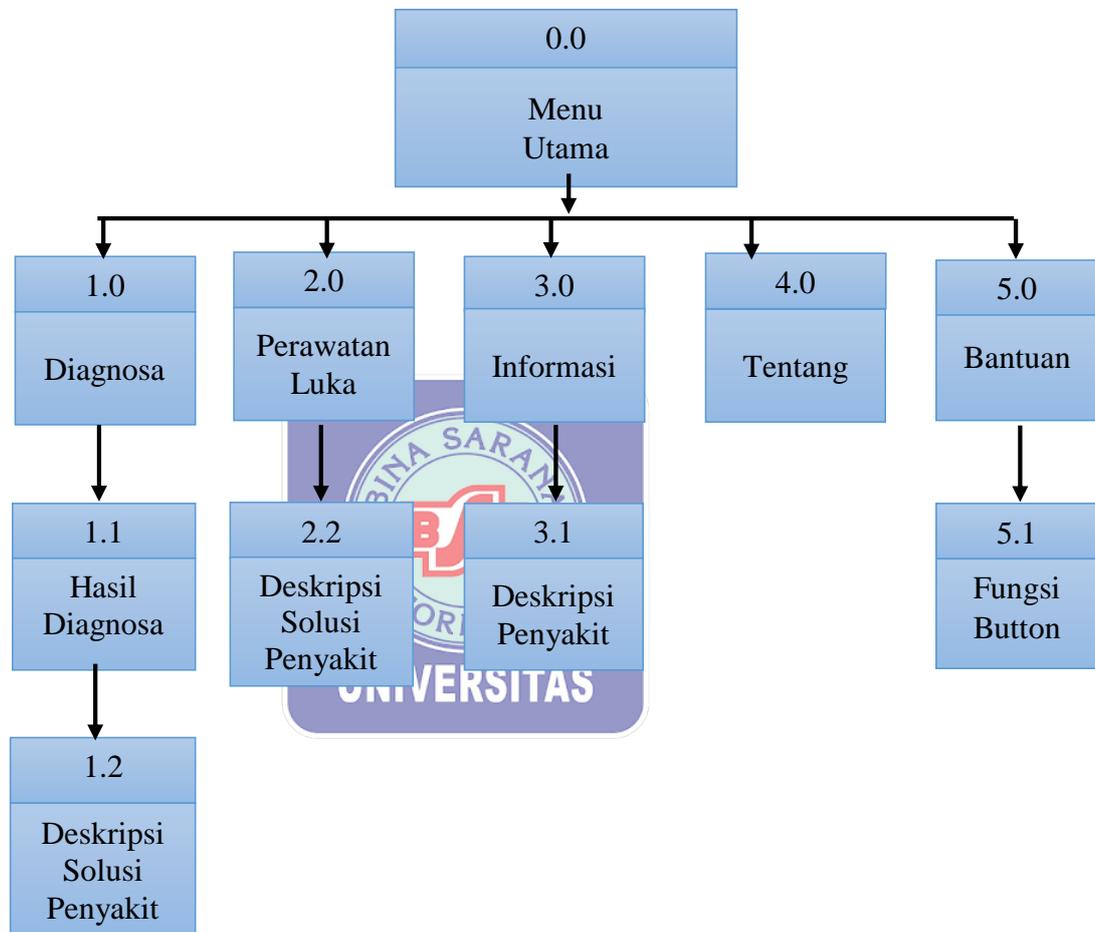


Sumber : Ropianto (2016:48)

Gambar II.13 Sequence diagram

### 2.2.9. Diagram HIPO (*Hierarchical Plus Input Process and Output*)

Menurut Sasmita, dkk (2016:4) mengemukakan bahwa diagram HIPO (*Hierarchical Plus Input Process and Output*) menggambarkan hubungan dari modul-modul dalam suatu sistem secara berjenjang



Gambar II.14 Diagram HIPO (*Hierarchical Plus Input Process and Output*)

Selain itu diagram HIPO juga menunjukkan secara garis besar hubungan dari *input*, proses dan *output*, dimana bagian *input* menunjukkan *item-item* data yang akan digunakan oleh bagian proses yang berisi langkah-langkah kerja dari fungsi atau modul dan bagian *output* berisi hasil pemrosesan data.