

IMPLEMENTASI ALGORITMA *LEVENSHTEIN DISTANCE* DAN ALGORITMA *KNUTH MORRIS PRATT* PADA APLIKASI ASMAUL HUSNA BERBASIS ANDROID

Uli Fitrianti, Mutammimul Ula

Teknik Informatika, Sistem Informasi Universitas Malikussaleh
Lhokseumawe Jl. Cot Tgk Nie-Reulet, Aceh Utara, 141 Indonesia
email : ullyfitrianti@gmail.com, moelula@gmail.com

ABSTRAK

Perkembangan teknologi *mobile phone* memicu perkembangan aplikasi-aplikasi yang berbasis android. Salah satunya adalah perkembangan perangkat yang dapat memberikan informasi secara terangkum layaknya sebuah asmaul husna digital. Adanya aplikasi asmaul husna yang terintegrasi dengan perangkat seluler akan sangat mempermudah para pengguna dalam belajar dan mengingat tentang 99 nama-nama asmaul husna yang telah terangkum dalam beberapa kategori bahasan dan dapat menutupi kekurangan dari asmaul husna berbentuk buku yang kurang praktis baik dari segi bentuk maupun dari segi kemampuan seperti pembelajaran terkait informasi tertentu masih harus dilakukan secara manual. Algoritma *Levenshtein Distance* dan algoritma *Knuth Morris Pratt* merupakan algoritma *string matching*. Pada penelitian tugas akhir ini penulis membuat suatu aplikasi asmaul husna menggunakan algoritma *Levenshtein Distance* dan algoritma *Knuth Morris Pratt* berbasis android yang dapat memberikan kemudahan bagi setiap orang dalam belajar dan mengingat nama Allah SWT. Hasil dari penelitian ini adalah penggunaan algoritma *Levenshtein Distance* untuk proses pengecekan ejaan jawaban yang diinputkan oleh pengguna dan algoritma *Knuth Morris Pratt* untuk proses pencocokan jawaban berupa *string* yang sudah benar dengan *string* yang tersimpan dalam *database* yang dapat memudahkan para pengguna mengetahui hasil *score* akhir jawaban yang benar dan jawaban yang salah.

Kata kunci : *String Matching, Levenshtein Distance, Knuth Morris Pratt, Aplikasi Asmaul Husna, Android*

1. Pendahuluan

Perkembangan Teknologi Informasi dan Komunikasi pada saat ini berkembang dengan sangat pesatnya. Perkembangan itu menyebabkan terjadinya perubahan perilaku dan aktivitas masyarakat dalam kehidupan sehari-hari.

Salah satu teknologi yang kini berkembang dengan sangat cepatnya adalah teknologi informasi dan komunikasi *mobile* (handphone). Teknologi *mobile* yang saat ini tidak hanya digunakan sebagai alat komunikasi saja, tetapi juga sebagai alat untuk memudahkan pengguna dalam kehidupan sehari-hari. Hal tersebut dapat terjadi karena pada teknologi *mobile* terdapat adanya banyak fasilitas, antara lain: pengaksesan internet, e-mail, organizer, musik, permainan dan sebagainya yang dapat digunakan dimana saja, kapan saja secara lebih cepat dan mudah. (Yunis Aprilianti : 2013)

Asmaul husna adalah salah satu ilmu pengetahuan, yang berkaitan dengan ilmu-ilmu keislaman. Dengan mempelajari Asmaul Husna kita dapat mengetahui nama-nama Allah SWT yang baik, dan agung serta indah sesuai dengan sifat-sifat-Nya. Yang kita ketahui selama ini ada 99 nama, namun sesungguhnya para ulama berbeda pendapat mengenai jumlahnya. Ada yang berpendapat 100, 132, 200, 1000 bahkan 4000 bahkan mungkin lebih. Namun yang lebih penting dari semua itu bukanlah jumlahnya, melainkan Dzat-Nya, Dzat Allah yang harus kita kenali sebagai sang maha pencipta, maha penguasa, dan Maha Pemilik dari Alam Semesta dan seluruh isinya. (Sri Ariyati : 2016)

Mengingat bahwa asmaul husna adalah ilmu keislaman yang sangat penting untuk dipelajari. Agar pembelajaran asma'ul husna ini menjadi menarik dan tidak membosankan, penulis tertarik untuk merancang dan membuat aplikasi Asmaul husna berbasis android yang menyenangkan sekaligus memberikan manfaat yang berguna. Diharapkan aplikasi ini memberikan kemudahan bagi setiap orang dalam belajar dan mengingat nama Allah SWT dimanapun dan kapanpun, karena seseorang yang dinyatakan cinta kepada Allah

SWT dapat dibuktikan dengan seberapa sering mereka menyebut dan mengingat Allah SWT dalam keadaan apapun.

Aplikasi Asmaul husna berbasis android ini membutuhkan algoritma pencocokan *string* yang akan membantu proses pencocokan pada kata asma'ul husna dengan artinya. Algoritma yang akan digunakan pada aplikasi ini adalah algoritma *Levenshtein Distance* dan algoritma *Knuth Morris Pratt*. Algoritma *Levenshtein Distance* adalah salah satu algoritma yang mengukur kemiripan (kesamaan) dua buah kata (*string*) sehingga algoritma ini digunakan untuk mengecek kebenaran ejaan yang ada pada arti asma'ul husna. Sedangkan Algoritma *Knuth Morris Pratt* adalah algoritma pencocokan *string* dengan susunan karakter dalam *string* yang dicocokkan dengan urutan karakter dalam *string* yang sama sehingga algoritma ini digunakan untuk mencocokkan kata yang telah diinputkan oleh *user*.

Berdasarkan latar belakang tersebut di atas, maka peneliti bermaksud membuat **"Implementasi Algoritma *Levenshtein Distance* dan Algoritma *Knuth Morris Pratt* Pada Aplikasi Asmaul Husna Berbasis Android"** untuk mempermudah mempelajari dan mengingat Asma Allah SWT

2. TINJAUAN PUSTAKA

2.1 ASMAUL HUSNA

Kata Asmaul Husna berasal dari bahasa arab yang merupakan gabungan dari dua kata yaitu al-Asma' dan al-Husna. Al- Asma' adalah bentuk jama' dari ismun yang berarti nama bentuk mashdar dari al-Ahsan yang berarti baik, bagus. Asma'ul husna adalah salah satu ilmu pengetahuan, yang berkaitan dengan ilmu-ilmu keislaman. Dengan mempelajari Asmaul Husna kita dapat mengetahui nama-nama Allah SWT yang baik, dan agung serta indah sesuai dengan sifat-sifat-Nya. Yang kita ketahui selama ini ada 99 nama, namun sesungguhnya para ulama berbeda pendapat mengenai jumlahnya. Ada yang berpendapat 100, 132, 200, 1000 bahkan 4000 bahkan mungkin lebih. Namun yang lebih penting dari semua itu bukanlah jumlahnya, melainkan Dzat-Nya, Dzat Allah yang harus kita kenali sebagai sang maha pencipta, maha penguasa, dan Maha Pemilik dari Alam Semesta dan seluruh isinya. Oleh sebab itu, pengenalan Asmaul Husna sangat penting untuk dipelajari dan ditanamkan sejak usia dini agar dapat mengerti dan mengenali siapa yang telah menciptakan alam semesta beserta isinya.

Sebagaimana kita ketahui bahwa, nama mempunyai arti dan peranan penting. Sebuah nama dalam kehidupan sehari-hari adalah untuk membangun relasi sosial. Sebuah nama juga bisa mewakili the holder of the name untuk bisa dihadirkan meski secara fisik ia tidak bisa hadir untuk disapa dan dikenal. Bagaimana kita bisa berhubungan dengan yang lainnya, berkomunikasi, bekerjasama dengan baik jika kita tidak mengenalnya. Dengan mngetahui dan menyebut namanya, berarti kita juga menghormati dan menghargainya. Dengan demikian, sebuah kerjasama atau hubungan akan dapat terjalin dengan baik. Bagaimana dengan Tuhan?. Kita sebagai makhluknya akan dapat mengenal dan berkomunikasi dengan-Nya jika mengenal bahasa yang dipahami bersama. Lalu bagaimana kita akan menyapanya jika kita tidak bernama?. Jika Allah tidak mengenalkan diri, maka manusia tidak akan mengenalnya, hanya menduga-duga, mengingat Tuhan Maha ghaib. Oleh karena itulah, Tuhan memperkenalkan diri-Nya melalui nabi, rasul dan alam semesta.

Asmaul Husna merupakan nama-nama yang dimiliki Allah. Tidak hanya nama-nama yang indah tetapi juga merupakan sifat-sifat mulia yang dimilikinya. Dengan mengetahui dan merasakannya lebih dalam, maka

manusia akan dapat merasakan bahwa Allah begitu dekat. (Sri Ariyati : 2016)

Seperti firman Allah swt dalam Al-qur'an Surat Al-A'raf ayat 180 :

يٰۤاَۤرَافُ ۙ هُوَ خَلِيْقُكَ
مِنْ اِسْمِ اِلٰهِكَ
عَبْدُ

Artinya :

"hanya milik Allah asma-ul husna, maka bermohonlah kepada-Nya dengan menyebut asma-ul husna itu dan tinggalkanlah orang-orang yang menyimpang dari kebenaran dalam (menyebut) nama-nama-Nya. Nanti mereka akan mendapat Balasan terhadap apa yang telah mereka kerjakan".

Dari Abu Hurairah r.a. ia berkata Nabi Muhammad s.a.w. pernah bersabda:

"Sesungguhnya Allah s.w.t mempunyai 99 nama, iaitu seratus kurang satu, barangsiapa menghitungnya (menghafal seluruhnya) masuklah ia kedalam syurga". (Riwayat Bukhari)

2.2 ALGORITMA LEVENSHTTEIN DISTANCE

Levenshtein Distance dibuat oleh Vladimir Levenshtein pada tahun 1965. Perhitungan edit distance didapatkan dari matriks yang digunakan untuk menghitung jumlah perbedaan string antara dua string. Perhitungan jarak antara dua string ini ditentukan dari jumlah minimum operasi perubahan untuk membuat string A menjadi string B.

Ada 3 macam operasi utama yang dapat dilakukan oleh algoritma ini :

1. Pengubahan karakter
2. Penambahan karakter
3. Penghapusan karakter

Algoritma ini berjalan mulai dari pojok kiri atas sebuah array dua dimensi yang telah diisi sejumlah karakter string awal dan string target dan diberikan nilai cost. Nilai cost pada ujung kanan bawah menjadi nilai edit distance yang menggambarkan jumlah perbedaan dua string. (Richard Junedy : 2014)

Langkah-langkah dari algoritma levenshtein distance (Benisius : 2010) yaitu :

1. Misalkan kata pertama adalah string asal (s) dan kata kedua adalah string target (t)
2. Tentukan m sebagai panjang dari s

3. Tentukan n sebagai panjang dari t
4. $d(0,0) = 0$
5. Jika $m = 0$, return n dan stop
6. Jika $n = 0$, return m dan stop
7. Buat matriks yang berisi 0 sampai m baris dan 0 sampai n kolom
8. Periksa setiap karakter dari s (i dari 1 sampai m) dan dari t (j dari 1 sampai n)
9. Jika $s[i]$ sama dengan $t[j]$, maka $cost = 0$, jika berbeda maka $cost = 1$
10. Tentukan cell $d[i, j]$ dari matriks yang sama dengan angka terendah,

$$\begin{array}{l}
 d[i, j] \\
 = \text{Min} \left\{ \begin{array}{l}
 \dots\dots\dots d[i-1, j] + 1 \quad (1) \\
 \dots\dots\dots d[i, j-1] + 1 \quad (2) \\
 \dots\dots\dots d[i-1, j-1] + cost \quad (3) \\
 \dots\dots\dots
 \end{array} \right.
 \end{array}$$

Keterangan rumus :

d = Cell-cell yang terdapat dalam matrik

i = Karakter yang terdapat pada string asal

j = Karakter yang terdapat pada string target

Min = Nilai minimum

Cost = Nilai 0 dan 1

11. Setelah selesai, maka jarak akan didapat dari cell $d(m, n)$

Berikut ini merupakan pseudocode dari algoritma Levenshtein

Distance :

```

n ← lenght(s)
m ← lenght(t)
if n = 0 then return m
else if m = 0 then return n
else
for i = 0 to n do
cost[0][i] ← i
for i = 0 to n do
cost[0][i] ← i
    
```

```
for i = 1 to n do
  for j = 1 to m do
    if (s[i-1]=t[j-1]) then
      cost[j][i] ← 0
    else cost[j][i] ← 1
    a1 ← cost [j][i-1]+1
    a2 ← cost [j-1][i]+1
    a3 ← cost [j-1][i-1] + cost [j][i]
    cost[j][i] ← min(a1,a2,a3)
  return cost[m][n]
```

2.3 ALGORITMA KNUTH MORRIS PRATT

Algoritma Knuth-Morris-Pratt dikembangkan oleh D. E. Knuth, bersama dengan J. H. Morris dan V. R. Pratt. Pada algoritma ini, kita memelihara informasi yang digunakan untuk melakukan jumlah pergeseran. Algoritma menggunakan informasi tersebut untuk membuat pergeseran yang lebih jauh. Secara sistematis langkah-langkah yang dilakukan algoritma Knuth-Morris-Pratt pada saat mencocokkan string:

1. Masukkan Query kata yang akan dicari.

Dengan permisalan P=Pattern atau pola susunan kata yang dijadikan sebagai contoh atau pola teks yang akan dicari T=Teks

2. Algoritma Knuth-Morris-Pratt mulai mencocokkan pattern atau pola susunan kata yang dijadikan sebagai contoh pada awal teks.

3. Dari kiri ke kanan, algoritma ini akan mencocokkan karakter per karakter pattern atau pola susunan kata yang dijadikan sebagai contoh dengan karakter di teks yang bersesuaian, sampai salah satu kondisi berikut dipenuhi :

a) Karakter di pattern atau pola susunan kata yang dijadikan sebagai contoh dan di teks yang dibandingkan tidak cocok (mismatch).

b) Semua karakter di pattern atau pola susunan kata yang dijadikan sebagai contoh cocok. Kemudian algoritma akan memberitahukan penemuan di posisi ini.

4. Algoritma kemudian menggeser pattern atau pola susunan kata yang dijadikan sebagai contoh berdasarkan tabel next, lalu mengulangi langkah no. 2 sampai pattern atau pola susunan kata yang dijadikan sebagai contoh berada di ujung teks. (Maya Rossaria : 2015)

Berikut ini merupakan pseudocode dari algoritma Knuth Morris Pratt

```

:procedure preKMP(
input P : array[0..n-1] of
char,
input n : integer,
input/output kmpNext :
array[0..n]
of integer)
Deklarasi:
i,j: integer
Algoritma
i := 0;
j := kmpNext[0] := -1;
while (i < n) {
while (j > -1 and not(P[i] =
P[j]))
j := kmpNext[j];
i:= i+1;
j:= j+1;if (P[i] = P[j])
kmpNext[i] := kmpNext[j];
else
kmpNext[i] := j;
endif
endwhile

```

```

procedure KMPSearch(
input m, n : integer
input P : array[0..n-1] of
char,
input T : array[0..m-1] of
char,
output ketemu : array[0..m-1]
of boolean)
Deklarasi:
i, j,next: integer
kmpNext : array[0..n] of
interger
Algoritma:
preKMP(n, P, kmpNext)
i:=0
while (i<= m-n) do
j:=0
while (j < n and T[i+j] = P[j])
do
j:=j+1
endwhile
if(j >= n) then
ketemu[i]:=true;
endif
next:= j - kmpNext[j]
i:= i+next

```


Perjalanan Android dimulai sejak Oktober 2003 ketika 4 orang pakar IT, Andi Rubin, Rich Miner, Nick Sears dan Chris White mendirikan Android.Inc, di California US. Visi Android untuk mewujudkan mobile device yang lebih peka dan mengerti pemiliknya, kemudian menarik raksasa dunia maya Google. Google kemudian mengakuisisi Android pada Agustus 2005. OS Android dibangun berbasis platform Linux yang bersifat open source, senada dengan Linux, Android juga bersifat Open Source. Dengan nama besar Google dan konsep open source pada OS Android, tidak membutuhkan waktu lama bagi android untuk bersaing dan menyisihkan Mobile OS lainnya seperti Symbian, Windos Mobile, Blackberry dan iOS. Kini siapa yang tak kenal Android yang telah menjelma menjadi penguasa Operating System bagi Smartphone. (Hendra Nugraha Lengkong : 2015)

Android adalah sebuah Operating System (OS) atau Sistem Operasi berbasis Linux yang diciptakan untuk telepon seluler. Jika diibaratkan pada komputer, itu seperti OS Windows atau Macintosh. Awalnya Android dikembangkan oleh sebuah perusahaan bernama Android Inc. Pendatang baru pembuat piranti lunak telepon seluler. Namun pada tahun 2005 diakuisisi oleh perusahaan raksasa search engine Google Inc. Android berbasis open source karena berbasis Linux, maka aplikasi Android itu terbuka bisa digunakan dan dikembangkan oleh siapa saja. Android memiliki beberapa versi yaitu : android versi 1.1, android versi 1.5 (Cupcake), android versi 1.6 (Donut), android versi 2.1 (Eclair), android versi 2.2 (Froyo), android versi 2.3 (Ginger Bread), android versi 3.0 (Honeycomb), android versi 4.0 (ICS=Ice Cream Sandwich), dan yang terbaru adalah Android versi 4.1 (Jelly Bean). (Imam FR Kusumaningati : 2012)

2.4.2 Fitur Android

Fitur-fitur android adalah sebagai berikut :

1. Framework Aplikasi

Fitur ini mendukung penggantian komponen dan penggunaan kembali komponen yang sudah dibuat (reusable). Seperti pada umumnya, framework memiliki keuntungan dalam proses pengkodean karena kita tidak perlu membuat kode untuk hal-hal yang pasti dilakukan seperti kode untuk menampilkan gambar, kode untuk koneksi database, dll.

2. Mesin Virtual Dalvik

Lingkungan dimana aplikasi android akan bekerja.

3. Integrated Browser

Berdasarkan Open Source engine WebKit.

4. Grafis

Dengan adanya fitur ini, kita bisa membuat aplikasi grafis 2D dan 3D karena Android memiliki library OpenGL ES 1,0.

5. Sqlite

Tugas dari fitur ini adalah berperan dalam penyimpanan data. Bahasanya mudah dimengerti dan merupakan sistem databasenya android.

6. Media Support

Fitur yang mendukung audio, video dan gambar.

7. GSM Telephony

Tidak semua android punya fitur ini karena fitur ini tergantung dari smartphone yang dimiliki.

8. Bluetooth, EDGE, 3G, WiFi

Fitur ini tidak selalu tersedia pada android karena tergantung Hardware atau smartphone.

9. Dukungan Perangkat Tambahan

Android dapat memanfaatkan kamera, layar sentuh, accelerometer, magnetometers, GPS, akselerasi 2D, dan akselerasi 3D.

10. Multi-Touch

Kemampuan layaknya handset modern yang dapat menggunakan dua jari atau lebih untuk berinteraksi dengan perangkat.

11. Lingkungan Development

Memiliki fitur emulator, tools, untuk debugging, profil dan kinerja memori dan plugin untuk IDE Eclipse.

12. Market

Seperti kebanyakan handphone yang memiliki tempat penjualan aplikasi, Market pada android merupakan katalog aplikasi yang dapat di download dan di install pada handphone melalui internet. (Hendra Nugraha Lengkong : 2015)

2.4.3 Versi Android

1. Android 1.5 Cupcake

Cupcake dirilis 30 April 2009. Cupcake menjadi versi android pertama yang menggunakan nama makanan. Konon katanya versi ini seharusnya versi 1.2, namun Google memutuskan untuk membuat revisi besar dan membuatnya menjadi versi 1.5 Cupcake adalah kue kecil yang dipanggang dalam cetakan berbentuk cup.

2. Android 1.6 Donut

Android V1.6, codename Donut, dirilis pada 15 September 2009. Pada versi ini diperbaiki beberapa kesalahan reboot, perubahan fitur foto dan video dan integrasi pencarian yang lebih baik. Donat merupakan panganan berbentuk cincin. Bulat bolong tengah. Adonan donat dimasak dengan cara digoreng dan biasanya disajikan dengan topping di atasnya.

3. Android 2.0/2.1 Eclair

Android 2.0/2.1 Eclair Dirilis 26 Oktober 2009. Eclair adalah makanan penutup yakni kue yang biasanya berbentuk persegi panjang yang dibuat dengan krim di tengah dan lapisan cokelat di atasnya.

4. Android 2.2 Froyo

Dirilis 20 Mei 2010. Menggunakan codename Froyo, yang merupakan makan penutup yang nama merek sebuah produk yang terbuat dari Yoghurt. Froyo singkatan dari Frozen Yoghurt, Froyo adalah yoghurt yang telah mengalami proses pendinginan, sehingga secara terlihat sama seperti es krim.

5. Android 2.3 Gingerbread

Android versi 2.3 Gingerbread dirilis resmi tanggal 6 Desember 2010. Gingerbread merupakan jenis kue kering yang dengan rasa jahe. Kue jahe biasanya dibuat pada perayaan hari libur akhir tahun di Amerika. Biasanya cemilan kering ini dicetak berbentuk tubuh manusia.

6. Android 3.0 Honeycomb

Dirilis tanggal 22 February 2011. H adalah sereal sarapan manis yang sudah dibuat oleh Posting Sereal. Seperti namanya, Honeycomb/sarang lebah, sereal ini terbuat dari potongan jagung berbentuk sarang lebah dengan rasa madu.

7. Android 4.0 Ice Cream Sandwich

Android 4.0-4.0.2 API Level 14 dan 4.0.3 API Level 15 pertama dirilis 19 Oktober 2011. Dinamai Ice Cream Sandwich. Ice Cream Sandwich es krim, biasanya rasa vanilla yang terjepit di antara dua kue coklat, dan biasanya berbentuk persegi panjang.

8. Android 4.1 Jelly bean

Android Jelly Bean diluncurkan pertama kali pada Juli 2012, dengan berbasis Linux Kernel dari Android 4.1 API Level 16, Android 4.2 API Level 17, Android 4.3 API Level 18. Penamaan mengadaptasi nama sejenis permen dalam beraneka macam rasa buah. Ukurannya sebesar kacang merah. Permen ini keras di luar tapi lunak di dalam serta lengket bila di gigit.

9. Android 4.4 KitKat

Android 4.4 Kitkat API level 19. Google mengumumkan Android KitKat (dinamai dengan izin Nestle dan Hershey) pada 3 september 2013. Dengan tanggal rilis 31 Oktober 2013. KitKat merupakan merk sebuah coklat yang dikeluarkan oleh Nestle. Rilis berikutnya setelah nama KitKat diperkirakan banyak pengamat akan diberi nomor 5.0 dan dinamai 'Pie'. (Hendra Nugraha Lengkong : 2015)

10. Android 5.0/5.1 Lollipop

Android versi ini dirilis pada tanggal 15 Oktober 2014. Versi OS ini mengusung perubahan besar dari segi UI yang nampak lebih flat dengan konsep material design. Versi Android ini sudah mendukung arsitektur 64-bit sehingga sudah memungkinkan untuk pengguna RAM diatas 3 GB pada hardware perangkat. Penggunaan processor 64-bit pun masih banyak diadopsi oleh para vendor, mulai dari penerapan pada perangkat flagship hingga perangkat kelas menengah ke bawah.

11. Android 6.0 Marshmallow

Versi Android ini resmi dirilis pada bulan September 2015.

12. Android 7.0 Nougat

Android versi ini resmi diperkenalkan pada akhir Juni 2016. Banyak netizen yang berspekulasi bahwa kemungkinan besar pemberian nama untuk Android versi "N" ini adalah Nutella. Namun Google menepis kabar tersebut setelah resmi memperkenalkannya bersamaan dengan dipamerkannya patung icon Android yang berdiri diatas potongan Nougat. (Pandu Dryad : 2016)

2.5 ECLIPSE

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak yang dapat dijalankan di semua *platform (platform-independent)*. Berikut ini sifat Eclipse :

- a. **Multi-platform** : eclipse dapat dijalankan di Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.
- b. **Multi-language** : eclipse dikembangkan dengan bahasa pemrograman java, akan tetapi eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/ C++, Cobol, Python, Perl, PHP, dan lain sebagainya
- c. **Multi-role** : selain sebagai IDE untuk pengembangan aplikasi, eclipse bisa juga dipakai untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, tes perangkat lunak, pengembangan web,

dan lain sebagainya. Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis (*free*)

dan *open source*, artinya setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan lebih lanjut oleh pengguna dengan komponen yang dinamakan *plug-in*.

Secara standar eclipse selalu dilengkapi dengan JDT (*Java Development Tools*), *plug-in* yang membuat eclipse kompatibel untuk mengembangkan program *Java*, dan *Plug-in Development Environment* (PDE) untuk mengembangkan *plug-in* baru. Konsep eclipse adalah IDE yang terbuka (*open*), mudah diperluas (*extensible*) untuk apa saja dan tidak untuk sesuatu yang spesifik. (Dedi Satria, 2014)

2.6 ADT (Android Development Tools)

Adapun pengertian ADT menurut beberapa pendapat, yaitu :

1. ADT adalah sebuah plugin untuk Eclipse yang menyediakan alat yang terintegrasi dengan IDE Eclipse. Sedangkan Android SDK (Software Development Kit) menyediakan Tools dan API yang diperlukan untuk mengembangkan aplikasi pada platform Android dengan menggunakan bahasa pemrograman Java. ADT menyediakan fitur yang membantu untuk mengembangkan aplikasi Android dengan cepat. ADT menyediakan GUI untuk mengakses banyak command line tools SD dengan desain UI untuk perancangan, desain, dan pembuatan user interface aplikasi. Karena ADT adalah sebuah plugin untuk Eclipse, maka didapatkan fungsi IDE yang stabil dengan fitur Android yang spesifik yang digabung dengan ADT. (Wandy Damarullah : 2013)

2. Android Development Tools (ADT) adalah plugin untuk Eclipse IDE yang dirancang khusus untuk memberikan integrated environment yang kuat untuk membuat aplikasi android. ADT memberikan kemampuan kepada Eclipse untuk membuat proyek baru Android secara tepat, membuat aplikasi User Interface, menambahkan komponen berdasarkan Android Framework API, melakukan debugging aplikasi yang dibuat dengan menggunakan Android SDK Tools dan bahkan melakukan distribusi aplikasi yang dibuat. Pembuatan aplikasi android dengan Eclipse beserta ADT sangat dianjurkan karena merupakan cara tercepat untuk memulai

membuat proyek Android. Dengan disediakannya project setup, serta tools yang sudah terintegrasi. (Hendra Nugraha Lengkong : 2015)

2.7 JDK (Java Development Kit)

Java Development Kit (JDK) adalah sebuah produk yang dikembangkan oleh oracle yang ditujukan untuk para developer Java. Sejak Java diperkenalkan, JDK merupakan Java Software Development Kit (SDK) yang paling sering digunakan.

JDK dilengkapi dengan banyak komponen untuk melakukan pemograman diantaranya :

1. Java : adalah loader untuk aplikasi Java.
2. Javac : Compiler yang mengonversi source code Java ke Java bytecode.
3. Appletviewer : tool ini bisa digunakan untuk menjalankan dan debug applet Java tanpa web browser.
4. Apt : tool pemrosesan annotation.
5. Extcheck : Utility untuk bisa mendeteksi konflik di file JAR.
6. Idlj : Compiler IDL-ke-Java. Utility ini akan men-generate Java binding dari file IDL.
7. Javadoc : Generator dokumentasi yang akan menghasilkan dokumentasi secara otomatis dari komentar di source code.
8. JAR : Achiver, yang akan memaketkan class library ke file JAR tunggal. Tool ini bisa digunakan untuk mengatur file JAR.
9. Java-C header dan stub generator : dituliskan untuk menuliskan metode native.
10. Javap : Dissassembler file class
11. Javaws : Java Web Start yang berfungsi sebagai launcher untuk aplikasi JNLP.
12. Jconsole : Java Monitoring and Management Console.
13. Jdb : Merupakan debiger Java.
14. Jchat : Merupakan Java Heap Analysis Tool (eksperimental).
15. Jinfo : Utility ini akan memperoleh informasi konfigurasi dari proses Java yang running atau crash dump (eksperimental).
16. Jmap : Utility ini akan memberikan output berupa peta memori dari java dan bisa mencetak dari proses tertentu atau core dump (eksperimental).
17. Jps : Java Virtual Machine Process Status adalah tool yang bisa menggunakan Hotspot Java Virtual Machine (JVM) di sistem target.
18. Jrunscript : Shell Java Command-line.
19. Jstack : Java Virtual Machines Stack trace dari thread Java.
20. Jstatd : jstatd daemon (eksperimental).

21. Policytool : Tool Policy creation dan management yang bisa menentukan policy yang dipakai oleh java runtime.
22. Visual VM : Tool visual untuk mengintegrasikan beberapa tool JDK command line.
23. Wsimport : Men-generate artefak JAX-WS portable.
24. Xjc : bagian dari Java API untuk XML Binding (JAXB). Ini bisa digunakan untuk menerima skema XML dan men-generate class Java.

SQLite DATABASE

Menurut Wahana dalam Rahmi Nanipaima Hutabarat (2016), SQLite adalah database open source yang ditanam pada android. Dukungan yang diberikan sama seperti RDBMS standar dan mendukung pada bahasa sintaks SQL. SQLite merupakan sebuah sistem manajemen basis data relational yang bersifat ACID-compliant dan memiliki ukuran library yang bersifat relatif kecil. Software ini ditulis menggunakan bahasa C.

Untuk menangani database, di Android tersedia paket bernama android.database. didalamnya terdapat semua class untuk bekerja dengan database. Class yang khusus untuk database ini bernama android.Database.sqlite. Ada beberapa komponen yang terdapat didalamnya, diantaranya :

1. SQLiteOpenHelper

Pada SQLiteOpenHelper terdapat method super() yang berguna untuk menentukan nama dan versi database yang akan dipakai. Di class ini adalah method onCreate() dan onUpgrade().

2. SQLiteDatabase

SQLiteDatabase adalah class dasar untuk database SQLite di Android. Fungsinya untuk membuka query, update dan menutup database. SQLiteDatabase punya banyak method, seperti method insert(), update(), dan delete(). Ada juga execSQL() yang bisa megeksekusi SQL secara langsung.

3. Cursor

Query berfungsi mengembalikan objek cursor. Cursor adalah hasil dari query,

yaitu satu barus (record) dari hasil query. Dengan cursor, android bisa melakukan buffer hasil query dengan efisien karena tidak perlu me-load data ke memori.

4. ListView dan ListActivity

ListView adalah view yang memungkinkan anda menampilkan berbagai list

elemen. ListActivity adalah activity khusus yang membuat penggunaan ListViews lebih mudah. Untuk bekerja dengan database dan ListViews, bisa menggunakan SimpleCursorAdapter. SimpleCursorAdapter memungkinkan pembuatan layout untuk tiap row dari ListView. (Rahmi Nanipaima Hutabarat : 2016)

2.9 UML (Unified Modelling Language)

Penelitian ini menggunakan diagram UML (Unified Modelling Language) sebagai pemodelan dari aplikasi yang akan dibangun. Adapun pengertian UML menurut beberapa pendapat, yaitu :

1. Menurut Braun dalam Haviluddin (2011), Unified Modelling Language (UML) adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem virtual.
2. Menurut Whitten dalam Toni (2011), UML adalah satu kumpulan konvensi pemodelan yang digunakan untuk menggambarkan sebuah system software yang terkait dengan obyek.

2.9.1 Use Case Diagram

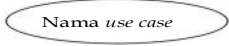


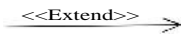

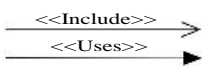
Use case diagram merupakan pemodelan untuk kelakuan (behaviour) sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Use case digunakan untuk mengetahui fungsi apa saja yang ada didalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. (S, Rosa A dan M. Shalahuddin, 2013)

Ada dua hal utama pada use case yaitu pendefinisian apa yang disebut aktor dan use case.

- a. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
- b. Use case merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukaran pesan antara unit atau aktor.

Berikut adalah simbol-simbol yang ada pada diagram use case :

Tabel 2.1. Notasi Use Case Diagram

Simbol	Keterangan
<p><i>Use Case</i></p> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antara unit atau aktor.
<p><i>Aktor/ actor</i></p> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri.
<p><i>Asosiasi/ association</i></p> 	Komunikasi antara aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor.
<p><i>Ekstensi/ extend</i></p> 	Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri.
<p><i>Generalisasi/ generalization</i></p> 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
<p><i>Menggunakan/ include/ uses</i></p> 	Relasi use case tambahan ke use case dimana use case yang ditambahkan memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini.


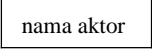

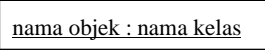


Sumber : S, Rosa A dan M. Shalahuddin (2013)

2.9.2 Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada use case.

Berikut adalah simbol-simbol yang ada pada diagram sekuen :

Tabel 2.2. Notasi *Sequence Diagram*

Simbol	Keterangan
<p>Aktor</p>  <p>nama aktor</p> <p>atau</p>  <p>nama aktor</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor.</p>
<p>Garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek</p>
<p>Objek</p>  <p>nama objek : nama kelas</p>	<p>Menyatakan objek yang berinteraksi pesan</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya. Aktor tidak memiliki waktu aktif.</p>
<p>Pesan tipe send</p> 	<p>I : masukan</p>

3.METODE PENELITIAN

3.1 LANGKAH -LANGKAH PEMBUATAN APLIKASI

3.1.1 Studi Kepustakaan

Sebelum memulai penelitian yang dilakukan terlebih dahulu adalah studi kepustakaan mengenai referensi tentang algoritma Levenshtein Distance dan algoritma Knuth Morris Pratt serta teori pendukung lainnya. Setelah memperoleh referensi tersebut, kemudian merancang aplikasi Asmaul husna dan sistem untuk pembelajaran tentang Asmaul husna dengan menerapkan metode berdasarkan dari studi kepustakaan yang dilakukan tersebut.

3.1.2 Pengumpulan Data

Adapun sebelum membangun aplikasi Asmaul husna maka diperlukan suatu database asmaul husna yang akan digunakan untuk mencocokkan arti yang diinputkan pada fitur bermain dengan arti yang terdapat didalam database untuk memberikan hasil dari proses pencocokan string apakah data ditemukan atau tidak ditemukan.

3.2 ANALISA KEBUTUHAN SISTEM

Tahap metode analisis ini merupakan tahap menganalisa sistem yang akan dibangun. Setelah analisa didapatkan maka langkah selanjutnya adalah membuat sebuah hasil analisa. Hasil analisa tersebut akan menjadi acuan dari perancangan sistem yang akan dibangun.

3.2.1 Analisa Kebutuhan Perangkat Keras (Hardware)

Perangkat keras merupakan perangkat yang sangat diperlukan didalam sistem komputer yang secara fisik dapat diraba dan dilihat, perangkat keras yang digunakan pada pembuatan perancangan sistem ini adalah sebagai berikut :

1. Laptop Toshiba Processor intel core i3
2. Smartphone Oppo R-827

3.2.2 Analisa Kebutuhan Perangkat Lunak (Software)

Perangkat lunak juga merupakan salah satu faktor penunjang untuk membuat dan merancang sebuah sistem. Perangkat lunak berfungsi sebagai pengolah data. Perangkat lunak yang digunakan pada perancangan sistem ini adalah :

1. Sistem Operasi Microsoft Windows 7 Ultimate
2. Microsoft Office Word 2007
3. Microsoft Office Visio 2007
4. Java Development Kit (JDK)
5. Eclipse

6. SQLite Database

3.2.3 Analisa Kebutuhan Input

Di dalam sistem yang dibangun memiliki kebutuhan data yang diinputkan, yaitu arti yang akan dijadikan pattern pencocokan string dengan arti yang terdapat didalam database.

3.2.4 Analisa Kebutuhan Proses

Pemrosesan data akan dilakukan oleh sistem setelah user menginputkan jawaban berupa arti asmaul husna. Proses yang terdapat pada sistem ada dua tahap yaitu proses untuk pengecekan ejaan dan proses untuk pencocokan jawaban setelah dilakukan pengecekan. Pengecekan ejaan tersebut akan dicek dengan menggunakan algoritma Levenshtein Distance dan proses pencocokan jawaban dicocokkan dengan menggunakan algoritma Knuth Morris Pratt. Algoritma pencocokan string Knuth Morris Pratt akan melakukan pencocokan data yang telah tersimpan secara didalam database.

3.2.5 Analisa Kebutuhan Output

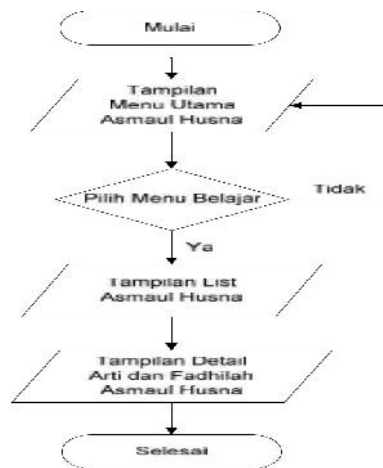
Output yang akan dihasilkan oleh sistem ini yaitu berupa score jawaban yang benar dan jawaban yang salah yang telah diinputkan oleh user. Adapun rancangan interface output dapat dilihat pada BAB IV.

3.3 DIAGRAM SISTEM

Diagram sistem digunakan untuk mengetahui proses apa saja yang berlangsung pada sistem. Diagram sistem untuk aplikasi ini menggunakan flowchart.

3.3.1 Flowchart Menu Belajar Aplikasi Asmaul Husna

Berikut merupakan gambaran skema sistem untuk menu belajar pada aplikasi Asmaul husna :



Gambar 2.3 *Flowchart* Menu Belajar Aplikasi Asmaul Husna

4. ANALISA DAN PEMBAHASAN

4.1 ANALISA SISTEM

Sistem yang akan dirancang oleh penulis dalam penelitian ini adalah aplikasi yang mengambil objek tentang Asmaul Husna. Bentuk aplikasi penulis rancang dalam dua bentuk, yaitu memaparkan pembahasan yang berdasarkan pada referensi-referensi dari bacaan baik dari buku maupun dari internet tentang Asmaul Husna yang telah penulis himpun dan memberikan fitur bermain bagi pengguna yang ingin mengingat kembali nama-nama Asmaul Husna yang telah dipelajari. Pembahasan yang ada pada referensi-referensi bacaan tersebut telah penulis rangkum menjadi lebih singkat dan terbagi dalam nama-nama judul Asmaul Husna guna memudahkan pengguna dalam membaca nantinya. Dan untuk pengguna yang ingin mengingat kembali Asmaul Husna atau detail tentang suatu nama Asmaul Husna yang ditemukannya dalam bacaan, sistem ini menyediakan fitur belajar yang hasilnya berupa detail tentang suatu nama Asmaul Husna yang dicari dan dilengkapi dengan gambar.

Adapun bahasan tentang objek yang dipaparkan dalam sistem adalah nama Asmaul Husna, arti dan fadhilah yang terkandung didalamnya. Pada menu bermain tersedia 10 soal bersifat random yang ada di dalam database yang merupakan kumpulan nama-nama Asmaul Husna dan arti yang telah dipaparkan. Arti yang terdapat dalam database akan dicocokkan dengan arti yang diinputkan dan kemudian mengeluarkan output berupa score hasil jawaban yang benar dijawab oleh user.

4.1.1 Analisa Kebutuhan Sistem

Berdasarkan kebutuhan yang akan diterapkan pada sistem ini, terdapat satu jenis kebutuhan yaitu kebutuhan sebagai user.

1. Kebutuhan user

- Lihat nama-nama Asmaul Husna
- Lihat arti dan fadhilah Asmaul Husna
- Bermain untuk mengingat kembali Asmaul Husna dengan menjawab soal
- Lihat hasil

4.2 PERANCANGAN SISTEM

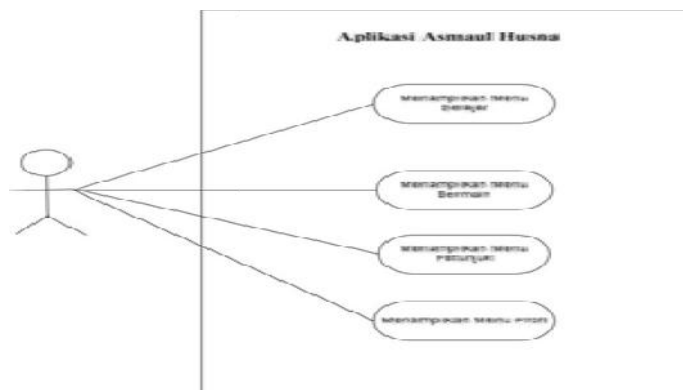
Pada perancangan sistem ini terdapat beberapa langkah yang akan dilakukan. Adapun langkah-langkahnya adalah sebagai berikut :

1. Menggambarkan use case diagram dari Aplikasi Asmaul Husna.

2. Menggambarkan sequence diagram untuk menjelaskan kelakuan objek pada use case.
3. Menggambarkan activity diagram untuk menjelaskan bagaimana aktivitas dari Aplikasi Asmaul Husna.

4.2.1 Use Case Diagram

Use case diagram menggambarkan interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Use case diagram digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.



Gambar 4.1 Use Case Diagram Aplikasi Asmaul Husna

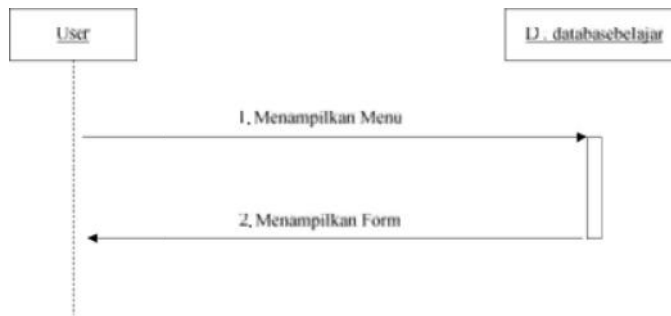
Keterangan :

Pada sistem ini terdapat satu aktor yang berinteraksi dengan sistem yaitu *user* dan empat fungsi yaitu, menampilkan menu belajar, menampilkan menu bermain, menampilkan menu petunjuk dan menampilkan menu profil. *User* dapat berinteraksi langsung dengan keseluruhan fungsi tersebut.

4.2.2 Sequence Diagram

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Adapun *sequence diagram* dari Aplikasi Asmaul Husna ini adalah :

a. *Sequence diagram* untuk menu belajar



Gambar 4.2 *Sequence Diagram* untuk Menu Belajar

Keterangan :

Pada menu utama, user memilih menu belajar Asmaul Husna kemudian halaman menu belajar terdapat 99 nama-nama Asmaul Husna. Kemudian akan muncul list nama-nama Asmaul Husna, jika user memilih salah satu nama Asmaul Husna yang terdapat pada list maka akan muncul detail dari Asmaul Husna tersebut yang berupa nama, arti dan fadhilah yang terkandung didalamnya. Pembahasan tersebut dipaparkan dalam bentuk teks yang disimpan dalam database.

b. *Sequence diagram* untuk bermain

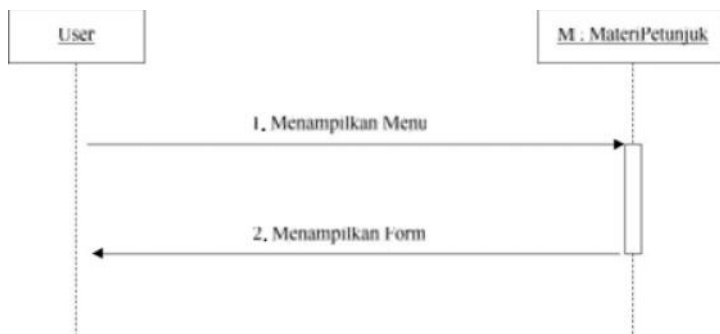


Gambar 4.3 *Sequence Diagram* untuk Menu Bermain

Keterangan :

Pada menu utama, *user* memilih menu bermain kemudian halaman menu bermain terdapat soal yang berisi tentang 99 nama-nama Asmaul Husna secara acak, kemudian *user* dapat menjawab soal-soal yang terdapat pada menu bermain sesuai dengan petunjuk, yaitu *user* dapat menjawab soal dengan cara menginput jawaban ke dalam aplikasi Asmaul Husna, kemudian sistem akan melakukan pengecekan ejaan yang diinputkan oleh *user*, jika ejaan jawaban yang diinputkan oleh *user* benar, maka akan dilakukan proses selanjutnya yaitu melakukan proses pencocokan jawaban yang terdapat didalam database, kemudian sistem akan menampilkan hasil berupa *score* jawaban yang benar yang dijawab oleh *user*. Dengan menjawab soal-soal tersebut maka *user* dapat mengingat kembali apa yang telah dipelajari pada menu belajar.

c. *Sequence diagram* untuk petunjuk

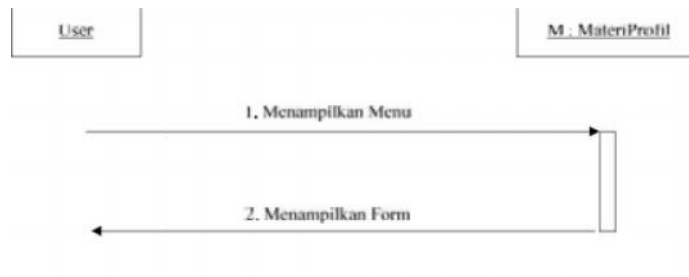


Gambar 4.4 *Sequence Diagram* untuk Menu Petunju

Keterangan :

Pada menu utama, *user* memilih menu petunjuk kemudian halaman menu petunjuk terdapat pembahasan bagaimana tata cara menjalankan aplikasi tersebut. Jika *user* merasa kebingungan dalam menjalankan aplikasi, maka dengan adanya menu petunjuk, maka *user* dapat memilih menu ini, karena menu ini akan sangat membantu memudahkan *user* dalam menggunakan aplikasi Asmaul Husna

d. *Sequence diagram* untuk profil



Gambar 4.5 *Sequence Diagram* untuk Menu Petunjuk

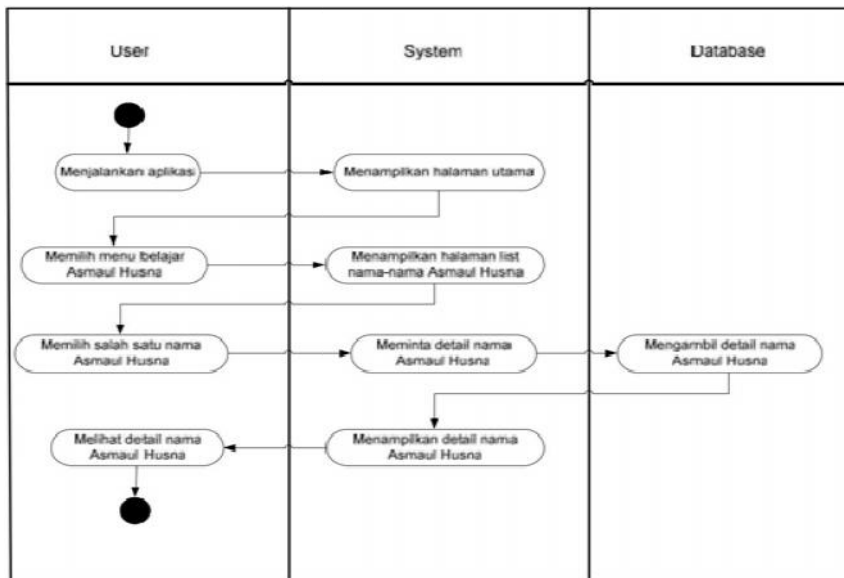
Keterangan :

Pada menu utama, user memilih menu profil kemudian terdapat pembahasan tentang pembuat aplikasi Asmaul Husna.

4.2.3 Activity Diagram

Activity diagram menggambarkan aktivitas dari aplikasi Asmaul Husna yang akan dibangun. Diagram ini bermanfaat untuk membantu memahami proses dari aplikasi Asmaul Husna ini secara keseluruhan.

- a. Activity diagram menu belajar

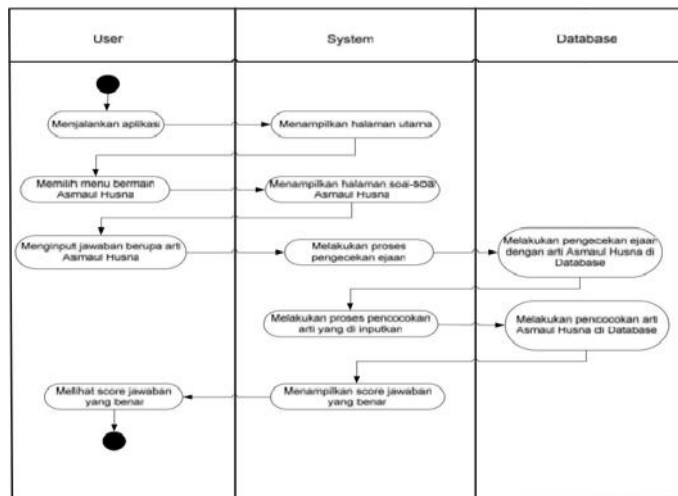


Gambar 4.6 Activity Diagram untuk Menu Belajar

Keterangan :

Pada saat user menjalankan aplikasi, sistem akan menampilkan form menu utama dan user memilih menu belajar Asmaul Husna. Kemudian sistem akan menampilkan form list nama-nama Asmaul Husna. Pada list nama-nama Asmaul Husna, user akan memilih salah satu nama Asmaul Husna, kemudian sistem akan menampilkan detail nama Asmaul Husna yang terdapat didalam database dan user dapat melihat detail Asmaul Husna tersebut.

b. Activity diagram menu bermain

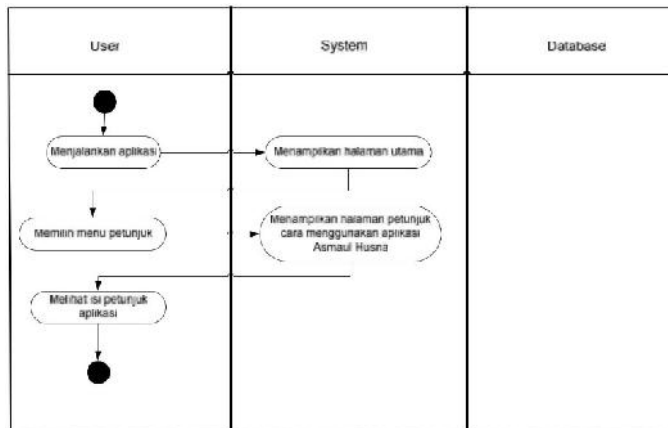


Gambar 4.7 Activity Diagram untuk Menu Bermain

Keterangan :

Pada saat user menjalankan aplikasi, sistem akan menampilkan form menu utama dan user memilih menu bermain Asmaul Husna. Kemudian sistem akan menampilkan form yang berisikan soal-soal tentang Asmaul Husna. Selanjutnya pada saat user menginputkan jawaban berupa arti Asmaul Husna, kemudian sistem akan melakukan proses pengecekan ejaan jawaban pada database Asmaul Husna, dan kemudian jika proses pengecekan telah selesai maka dilanjutkan dengan proses pencocokan jawaban yang sudah benar dengan database yang ada. Kemudian sistem akan menampilkan score jawaban yang benar yang telah dijawab oleh user, dan user dapat melihat score jawaban yang benar tersebut.

c. Activity diagram menu petunjuk

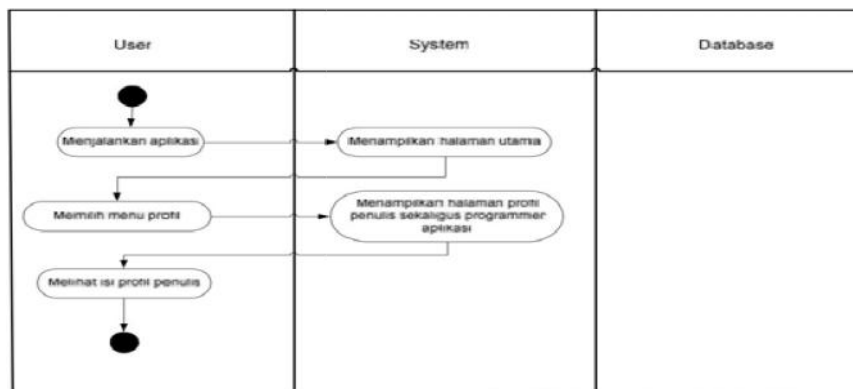


Gambar 4.8 Activity Diagram untuk Menu Petunjuk

Keterangan :

Pada saat user menjalankan aplikasi, sistem akan menampilkan form utama dan user memilih menu petunjuk. Kemudian sistem akan menampilkan form petunjuk yang menjelaskan bagaimana tata cara menggunakan aplikasi Asmaul Husna. Dan kemudian user dapat melihat isi petunjuk aplikasi tersebut.

d. Activity diagram menu profil



Gambar 4.9 Activity Diagram untuk Menu Profil

Keterangan :

Pada saat user menjalankan aplikasi, sistem akan menampilkan form utama dan user memilih menu profil. Kemudian sistem akan menampilkan form profil yang menjelaskan tentang profil pembuat aplikasi Asmaul Husna tersebut. Dan kemudian user dapat melihat isi profil pembuat aplikasi Asmaul Husna tersebut.

4.3 DESAIN TABEL

Berikut ini merupakan desain tabel dari aplikasi Asmaul Husna. Perancangan ini menggunakan database SQLite dengan nama database asmaul husna. Berikut rancangan tabel selengkapnya :

4.3.1 Tabel asmaul husna

Tabel ini berisi kumpulan pembahasan tentang 99 Asmaul Husna. Struktur tabel ini dapat dilihat pada tabel dibawah ini.

Tabel 4.1 Tabel asmaul husna

<i>Field Name</i>	<i>Type</i>	Keterangan
Id (<i>primary key</i>)	Integer	Id judul asmaul husna
Nama	Text	Data judul asmaul husna
Arti	Text	Arti dari judul asmaul husna
Fadhilah	Text	Fadhilah dari judul asmaul husna
Arab	Text	Tulisan arab dari asmaul husna

4.4 IMPLEMENTASI ALGORITMA LEVENSHEIN DISTANCE

Implementasi dari algoritma ini terletak pada proses pengecekan ejaan yang diinputkan oleh *user* kedalam sistem dengan jawaban yang ada didalam *database*. Untuk ejaan yang di cek, proses pengecekan dilakukan berdasarkan perbandingan karakter kedua buah *string*. Algoritma Levenshtein Distance adalah algoritma yang menghitung jumlah pada operasi string paling sedikit yang diperlukan untuk mentransformasikan suatu string menjadi string yang lain. Algoritma

Levenshtein Distance mempunyai tiga langkah dalam pengubahan *string* pertama (*string* awal) menjadi *string* kedua (*string* target), yaitu dengan cara pengubahan karakter, penambahan karakter, dan penghapusan karakter. Pada Algoritma Levenshtein Distance, jawaban yang diinputkan oleh *user* untuk dicek disebut *string* awal (*s*) dan data asmaul husna dalam *database* sebagai pembanding pengecekan disebut *string* target (*t*).

Pada proses pengecekan *string* dilakukan pencarian nilai *cost* pada ujung kanan bawah tabel, proses pengecekan dilakukan mulai dari pojok kiri atas sebuah array dua dimensi yang telah diisi sejumlah karakter pada dua buah *string*.

Berikut ini penulis akan menjelaskan proses pengecekan yang terjadi pada menu bermain menggunakan Algoritma Levenshtein Distance :

Contoh :

String awal = MAHAESA

String target = MAAESA

Penyelesaian :

$s =$ MAHAESA

$t =$ MAAESA

$m =$ 7

$n =$ 6

- a. Mencari jarak *string* awal ke *string* target
1. Matriks ukuran (7 x 6)

	-	M	A	H	A	E	S	A
-	0	1	2	3	4	5	6	7
M	1							
A	2							
A	3							
E	4							
S	5							
A	6							

2. Cell d [1,1] d
 = (i-1, j-1)
 = (1-1, 1-1)
 = 0

	-	M	A	H	A	E	S	A
-	0	1	2	3	4	5	6	7
M	1	0						
A	2							
A	3							
E	4							
S	5							
A	6							

- a. Cell d [1,2]
- $d(i - 1, j) + 1$
 $d(1 - 1, 2) + 1 = 3$
 - $d(i, j - 1) + 1$
 $d(1, 2 - 1) + 1 = 1$
 - $d(i - 1, j - 1) + \text{cost}$
 $d(1 - 1, 2 - 1) + 1 =$
 2 Nilai terkecil adalah 1

	-	M	A	H	A	E	S	A
-	0	1	2	3	4	5	6	7
M	1	0	1					
A	2							
A	3							
E	4							
S	5							
A	6							

4. Cell d [1,3]

$$\bullet d(i - 1, j) + 1$$

$$=d(1 - 1, 3) + 1 = 4$$

$$\bullet d(i, j - 1) + 1$$

$$=d(1, 3 - 1) + 1 = 2$$

$$\bullet d(i - 1, j - 1) + \text{cost}$$

$$=d(1 - 1, 3 - 1) + 1 = 3 \text{ Nilai terkecil adalah } 2$$

5.KESIMPULAN DAN SARAN

5.1 KESIMPULAN

Adapun kesimpulan yang diperoleh dari penelitian ini, adalah sebagai berikut :

1. Aplikasi Asmaul Husna ini memberikan kemudahan bagi setiap orang dalam belajar dan mengingat nama Allah SWT dimanapun dan kapanpun.
2. Penggunaan algoritma *Levenshtein Distance* pada pengecekan ejaan jawaban dan algoritma *Knuth Morris Pratt* pada pencocokan jawaban yang terdapat dalam aplikasi Asmaul Husna ini dapat memudahkan *user* untuk mengingat kembali apa yang telah dipelajari tentang asmaul husna dan kemudian dapat mengulanginya secara langsung dan dilengkapi dengan gambar terkait.
3. Pada aplikasi Asmaul Husna ini terdapat menu belajar, bermain, petunjuk dan profil. Algoritma *Levenshtein Distance* dan Algoritma *Knuth Morris Pratt* digunakan pada menu bermain dimana saat *user* menginputkan jawaban kemudian sistem akan melakukan proses pengecekan ejaan dan pencocokan jawaban hingga menghasilkan *output* berupa score jawaban yang benar.

5.2 SARAN

Agar sistem ini dapat bermanfaat dengan baik untuk sekarang maupun akan datang, maka penulis memberikan saran sebagai berikut :

1. Diharapkan penelitian berikutnya dapat memaparkan dalil berupa ayat-ayat al-quran mengenai 99 nama-nama asmaul husna yang lebih banyak.
2. Diharapkan penelitian terkait berikutnya dapat mencari metode untuk penyajian Asmaul Husna yang lebih efektif agar pengguna tidak bosan dalam mempelajari Asmaul Husna yang hanya berupa teks dan gambar.
3. Menambahkan multimedia seperti *audio* dan *video* agar Asmaul Husna dapat disajikan dengan cara yang lebih menyenangkan.
4. Memperindah tampilannya sehingga akan lebih menarik lagi untuk digunakan.

DAFTAR PUSTAKA

Andi. 2013. Pemograman Android dengan APP Inventor. Kerjasama Andi dan Wahana Komputer: Yogyakarta dan Semarang.

Aprilianti, Yunis, dkk., 2013. Aplikasi Game Edukasi Matematika Berbasis Android Application Of Education Mobile Games For Math Based On Android. Teknik Informatika, Institut Sains dan Teknologi AKPRIND Yogyakarta. Yogyakarta. Jurnal SCRIPT. ISSN : 2338-6304 Vol.1 No.1
[Diakses pada situs
<http://journal.akprind.ac.id/index.php/script/article/download/54/39>
Selasa, 6 September 2016]

Ariyati, Sri, dan Titik Misriati. 2016. Perancangan Animasi Interaktif Pembelajaran Asmaul Husna. Teknik Komputer AMIK BSI. Bekasi dan Jakarta. Jurnal Teknik Komputer AMIK BSI. ISSN : 2442-2436 Vol. II No.1
[Diakses pada situs
<http://ejournal.bsi.ac.id/ejurnal/index.php/jtk/article/download/369/278>
8 Jumat, 2 September 2016]

A.S., Rosa, dan M. Shalahuddin. 2013. Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek. Informatika: Bandung.
Damarullah, Wandy, dkk. 2013. Aplikasi Pengenalan dan Pembelajaran Bahasa Korea (Hangeul) Berbasis Android. Teknik Informatika, Institut Sains & Teknologi AKPRIND Yogyakarta. Yogyakarta. Jurnal SCRIPT. ISSN : 2338-6304 Vol. 1 No. 1

[Diakses pada situs

<http://download.portalgaruda.org/article.php?article=181516&val=6283&title=>

APLIKASI%20PENGENALAN%20DAN%20PEMBELAJARAN%20BAHASA%20KOREA%20(HANGEUL)%20BERBASIS%20ANDROID Jumat, 2 Desember 2016]

Dryad, Pandu. 2016. Perkembangan Versi-versi OS Android.

[Diakses pada situs <http://www.techijau.com> Jumat, 2 Desember 2016]

Haviluddin. 2011. Memahami Penggunaan UML (Unified Modelling Language). Program Studi Ilmu Komputer FMIPA Mulawarman. Samarinda.

[Diakses pada situs <https://informatikamulawarman.com/2011/10/01-jurnal-informatika-mulawarman-feb-2011.pdf> Kamis, 1 September 2016]

Hutabarat, Rahmi Nanipaima. 2016. Aplikasi Ensiklopedia Provinsi Aceh Berbasis Android Menggunakan Algoritma Apostolico-Crochemore.

Program Studi Teknik Informatika. Universitas Malikussaleh: Lhokseumawe.

Junedy. S, Richard. 2014. Perancangan Aplikasi Deteksi Kemiripan Isi Dokumen Teks Dengan Menggunakan Metode Levenshtein Distance. Teknik Informatika STMIK Budi Darma Medan. Medan. Jurnal Pelita Informatika Budi Darma. ISSN : 2301-9425 Vol.VII No.2

[Diakses pada situs <http://pelita>

informatika.com/berkas/jurnal/28.%20richard%20jjune.pdf Kamis, 1 September 2016]

Kurniawan. Tony. 2011. Pengembangan Aplikasi Helpdesk Pada JICT (Jakarta Internasional Container Terminal). Universitas Islam Negeri Syarif Hidayatullah. Jakarta.

[Diakses pada situs

<http://repository.uinjkt.ac.id/dspace/bitstream/123456789/250/1/101434-TONI%20KURNIAWAN-FST.PDF> Kamis, 1 September 2016]

Kusumaningati, Imam FR. 2012. Ngandroid Hidup Menjadi Mudah dan Menyenangkan Dengan Android. PT Elex Media Komputindo: Jakarta.

Pradana, Pandu. 2013. Mengenal Android Lebih Dekat. Skripta Media Creative: Yogyakarta.

Rossaria, Maya, dkk. 2015. Implementasi Algoritma Pencocokan String Knuth Morris Pratt Dalam Aplikasi Pencarian Dokumen Digital Berbasis Android. Teknik Informatika Universitas Bengkulu. Bengkulu. Jurnal Rekursif. ISSN 2303-0755 Vol. 3 No.2

[Diakses pada situs

<http://download.portalgaruda.org/article.php?article=406637&val=7008&title=>

Implementasi%20Algoritma%20Pencocokan%20String%20Knuth%20Morris-

Pratt%20dalam%20Aplikasi%20Pencarian%20Dokumen%20Digital%20Berbasis%20Android Jumat, 2 September 2016]

Satria, Dedi. 2014. Aplikasi Ensiklopedia Digital Phylum Mollusca Berbasis Android Dengan Metode String Matching. Program Studi Teknik Informatika. Politeknik Negeri Lhokseumawe: Lhokseumawe.

Soleh, Moch. Yusup. 2011. Implementasi Algoritma Knuth Morris Pratt dan Booyer-Moore dalam Aplikasi Search Engine Sederhana. Teknik Informatika Institut Teknologi Bandung. Bandung.

[Diakses pada situs

[http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2010-](http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2010-2011/Makalah2010/MakalahStima2010-096.pdf)

2011/Makalah2010/MakalahStima2010-096.pdf Jumat, 2 September 2016]