

IMPLEMENTASI ALGORITMA *BRUTE FORCE* DAN FITUR *LOCATION BASED SERVICE* (LBS) PADA APLIKASI KUMPULAN DOA HARIAN BERBASIS ANDROID

Fenty E.M.A¹, Anif Hanifa², Novi Riyanto³

^{1,2,3} Program Studi Teknik Informatika, Fakultas Sains dan Teknologi,
UIN Syarif Hidayatullah Jakarta, Indonesia

¹fentyema@gmail.com

²anifhanifa@yahoo.com

³nooveice@gmail.com

Abstrak: Bagi seorang muslim, doa adalah amaliah yang selalu dilakukan dalam kehidupan sehari-hari. Perkembangan penggunaan *smartphone* khususnya yang berbasis android dan mendukung fitur *Location Based Service* (LBS), membuat peneliti tertantang untuk membuat aplikasi kumpulan doa harian yang dapat memberikan tuntunan doa yang harus dibaca saat berada pada lokasi tertentu. Aplikasi ini juga dilengkapi fitur pencarian doa yang dikembangkan menggunakan algoritma *brute force*. Penggunaan algoritma ini dinilai cukup memadai untuk aplikasi ini karena mudah diaplikasikan dalam pencocokan *string*. Metode pengembangan perangkat lunak yang digunakan adalah *Rapid Application Development* (RAD).

Kata Kunci: Algoritma *Brute Force*, *Rapid Application Development* (RAD), *Location Based Service* (LBS)

Abstract: As A Moslem prayer is daily worship. *Smartphone* usefulness especially *Android Based one* and support *Location Based Services* (LBS) features made researcher want to made daily Prayer application which could give lead which one to read in certain situation. This application also equipped with prayers search menu which developed using *Brute Force Algorithm*. This algorithm was enough for this application because it easy to implemented in this application in string matching. This Application development methods is *Rapid Application Development* (RAD)

Keywords : *Brute Force Algorithm*, *Rapid Application Development* (RAD), *Location Based Services* (LBS).

Dengan banyaknya pengguna *smartphone* Android di Indonesia maka memudahkan para muslim untuk membaca doa-doa harian Islam tanpa harus terbatas oleh ruang dan waktu, baik dalam perjalanan, saat berbelanja maupun ketika bekerja. Salah satu fitur yang menarik tersebut adalah teknologi GPS (*Global Positioning System*). Saat ini hampir semua *smartphone* menggunakan perangkat GPS, untuk memberikan informasi posisi pengguna berdasarkan lokasi.

Teknologi ini menjadi penting saat ini karena dapat dimanfaatkan untuk navigasi (penunjuk arah) serta pelacakan. Layanan berbasis lokasi atau *Location Based Service* (LBS) merupakan layanan informasi yang dapat diakses melalui perangkat *mobile* melalui jaringan seluler dan memiliki kemampuan untuk memanfaatkan lokasi posisi perangkat *mobile*. Pengertian yang sama juga diberikan oleh *Open Geospatial Consortium* mengenai LBS yaitu sebuah layanan *Internet*

I. PENDAHULUAN

Android adalah sebuah sistem operasi untuk *smartphone* berbasis Linux. Berdasarkan data IDC (*International Data Corporation*) sebagai situs layanan analisa data terpercaya memberikan informasi bahwa sistem operasi paling populer di Indonesia saat ini adalah Android dengan cakupan 52% dari total pengguna *smartphone* di Indonesia.

Protocol (IP) nirkabel yang menggunakan informasi geografi untuk memberikan layanan kepada pengguna perangkat *mobile*. Setiap layanan aplikasi yang memanfaatkan posisi terminal *mobile*. *Location Base Service* (LBS) adalah sebuah nama umum untuk sebuah nama baru dimana informasi lokasi menjadi parameter utamanya pengertian lain juga diberikan, bahwa LBS sebenarnya salah satu nilai tambah dari layanan seluler *Global System for Mobile Communication* (GSM).

LBS bukanlah sistem, tetapi merupakan layanan yang menggunakan sistem tambahan penunjang sistem GSM. Jadi jelas, bisa jadi ada beberapa opsi sistem yang dapat mengirim layanan sistem ini dengan teknologi yang bervariasi. Tetapi pada dasarnya, sistem-sistem tersebut menggunakan prinsip dasar yang sama, yaitu Triangulasi. Jadi, prinsipnya tidak jauh beda dengan sistem GPS, hanya saja fungsi satelit digantikan oleh BTS. Perbedaan lain antara LBS dan GPS adalah pemroses posisi. Pada peralatan GPS, pengguna yang mengolah dan mengukur posisi. Sistem *backend* satelit hanya memberikan info posisi satelit, kecepatan, dan waktu.

Aplikasi kumpulan doa harian yang dikembangkan memanfaatkan *Location Based Service* (LBS) untuk menentukan lokasi pengguna dan aplikasi akan menampilkan doa yang harus dibaca sesuai lokasi pengguna berada. Selain itu, aplikasi memberi fitur untuk mencari suatu bacaan doa yang akan dibaca atau dipelajari oleh pengguna. Dalam mencari doa pada suatu aplikasi kumpulan doa, kita akan mengalami kesulitan jika doa-doa dalam aplikasi tersebut berjumlah puluhan atau ratusan. Tentu tidak efektif jika harus mengurutkan satu per satu doa untuk mendapatkan doa yang ingin kita baca. Akan lebih baik jika kita mampu mendapatkan informasi doa yang

diinginkan dalam waktu singkat. Permasalahan ini sudah dapat diatasi oleh algoritma pencocokan *string*. Algoritma pencocokan *string* yang dapat dijadikan alternatif untuk pencarian ini adalah algoritma *Brute Force*. Algoritma *Brute Force* bersifat lempang dalam melakukan komputasi, sehingga mudah dimengerti [3].

II. PENGEMBANGAN APLIKASI MENGUNAKAN METODE *RAPID APPLICATION DEVELOPMENT* (RAD)

Pengembangan aplikasi yang digunakan adalah metode *Rapid Application Development* (RAD), dengan tahapan-tahapan dalam RAD berikut [3]:

1. Fase Perencanaan Syarat-syarat
2. Fase Workshop Desain
3. Fase Implementasi

A. Fase Perencanaan Syarat-syarat

Pada fase ini dilakukan identifikasi tujuan dibuatnya aplikasi serta syarat-syarat informasi yang dibutuhkan.

1. Tujuan Aplikasi (*Application Goals*)

Aplikasi kumpulan doa harian harus memiliki kemampuan:

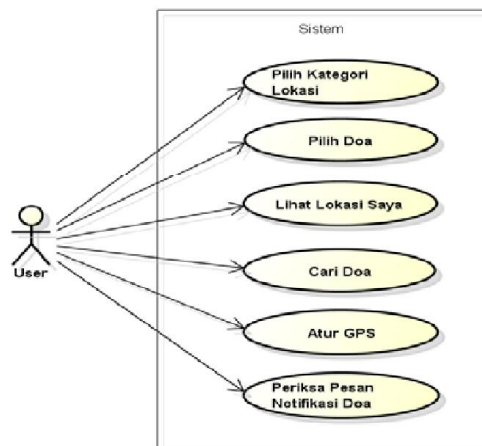
- a) Menampilkan daftar doa-doa yang dapat diamalkan berdasarkan kategori tempat seperti masjid, sekolah, pasar, terminal/stasiun/ bandara/pelabuhan, rumah makan, pemakaman dan rumah sakit.
- b) Menampilkan rincian tiap doa berupa ayat-ayat doa dalam huruf arab, lafal doa dalam tulisan latin, dan arti doa dalam Bahasa Indonesia;
- c) Menampilkan posisi pengguna dalam layanan *google map*, pengguna dapat melihat dimana lokasi saat ini dalam *google map* dengan mengaktifkan GPS dalam *smartphone* dan *smartphone* harus terhubung dengan internet;

- Jurnal Pseudocode, Volume 2 Nomor 1, September 2014, ISSN 2355 – 5920

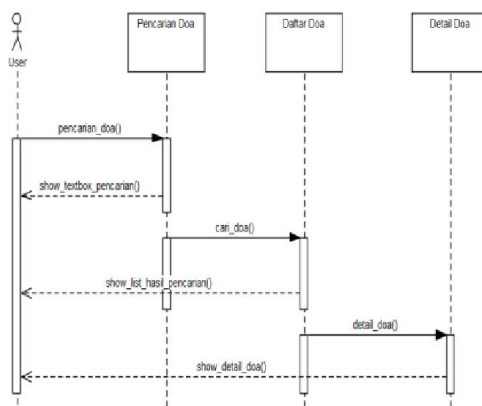
- ## 2. Syarat Aplikasi (*Application Requirements*)

Sedangkan implementasinya dapat diinstal pada *smartphone* dengan sistem operasi mulai Android froyo (versi 2.2).

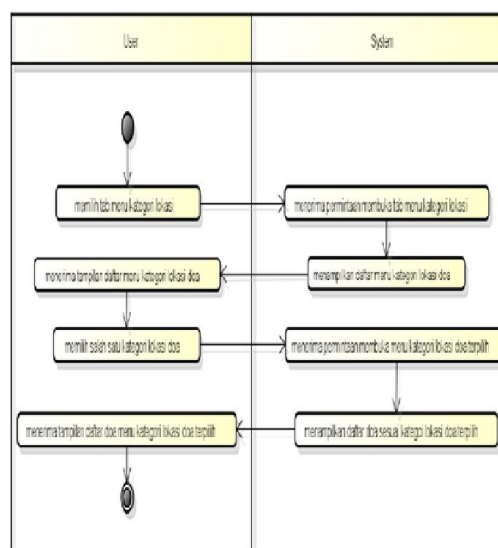
B. Fase Workshop Design



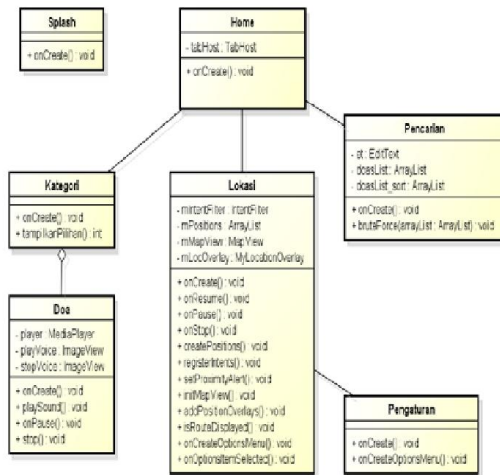
Gambar 1. Diagram *Use Case* Aplikasi Kumpulan Doa Harian Berbasis Android



Gambar 2. Diagram *Activity* Kumpulan Doa Harian Berbasis Android



Gambar 3. Diagram *Sequence* Pencarian Doa



Gambar 4. Diagram Class Aplikasi Kumpulan Doa

Dalam perancangan algoritma Brute Force pencarian doa, peneliti menggunakan *pseudocode* sebagai berikut [3] :

```

Procedure BruteForceSearch(
  input m, n : integer,
  input P : array[1..m] of char,
  input T : array[1..n] of char,
  output idx: integer)
  {Mencari kecocokan pattern P di
  dalam teks T. Jika ditemukan P di dalam
  T, lokasi awal kecocokan disimpan di
  dalam peubah idx..
  Masukan: pattern P yang
  panjangnya m dan teks T yang
  panjangnya n. Teks T direpresentasikan
  sebagai string (array of integer)
  Keluaran: posisi awal kecocokan
  (idx). Jika P tidak ditemukan, idx = -1.}

  Deklarasi
  s, j : integer
  ketemu : boolean

  Algoritma:
  s ← 0
  ketemu ← false
  while (s ≤ n-m) and (not ketemu) do
    j ← 1
    while (j ≤ m) and (P[j] =
    T[s+j]) do
      j ← j+1
    endwhile
    {j > m or P[j] ≠ T[s+j]}
    if j = m then {kecocokan string
    ditemukan}
      ketemu ← true
    else
      s ← s+1 {geser
      pattern satu karakter ke

```

```

      kanan teks}
    endif
  endfor
  {s > n-m or ketemu}
  if ketemu then
    idx ← s+1 {catatan: jika
    indeks array dimulai dari 0, idx ← s}
  else
    idx ← -1
  endif

```

Langkah selanjutnya adalah merancang *user interface* dari aplikasi kumpulan doa harian sebagai berikut:

icon 1 Kategori Lokasi	icon 2 Tab 2	icon 3 Tab 3
Icon Lokasi 1	Icon Lokasi 2	Icon Lokasi 3
Icon Lokasi 4	Icon Lokasi 5	Icon Lokasi 6
Icon Lokasi 7	Icon Lokasi 8	

Gambar 6: Rancangan Tampilan Tab Kategori Lokasi

icon 1 Tab 1	icon 2 Lokasi Saya	icon 3 Tab 3
Google Map		

Gambar 7: Rancangan Tampilan Tab “Lokasi Saya”

icon 1 Tab 1	icon 2 Tab 2	icon 3 Pencarian Doa
Textbox Pencarian Doa		
Doa 1		
Doa 2		
Doa 3		

Gambar 8: Rancangan Tampilan Tab “Pencarian Doa”

Nama Kategori Lokasi
Doa 1
Doa 2
Doa 3
Doa 4
Doa 5
Doa 6

Gambar 9: Rancangan Tampilan Tab “Daftar Doa”

Judul Doa
Ayat Doa
Lafal Doa
Arti Doa

Gambar 10: Rancangan Tampilan Tab “Rincian Doa”

C. Fase Implementasi

Pada fase ini terdiri dari implementasi hardware dan software yang digunakan, penerapan algoritma Brute Force dan pembuatan antar muka aplikasi. Dalam proses implementasi ini ada beberapa langkah-langkah yang dilakukan, yaitu:

1. Memasukkan kode program (*coding*), tahap ini dilakukan dengan menggunakan program pengembang aplikasi Android, yaitu *Android Developer Tools* (ADT), *Android SDK* (*Software Development Kit*);
2. Melakukan proses pemaketan dengan menggunakan fasilitas yang disediakan oleh *Android SDK*;
3. Menguji program dengan menggunakan *handset smartphone* Android, serta melakukan *debugging* atau perbaikan program jika perlu.

Perangkat lunak yang digunakan dalam membangun aplikasi ini adalah sebagai berikut:

1. *Java Development Kit* (JDK);
2. *Android SDK Emulator*;
3. *Eclipse IDE*.

Perangkat keras yang digunakan dalam membangun aplikasi ini adalah sebagai berikut:

1. Komputer dengan *processor* Intel Core i3 2,1 GHz
2. Memory 2 GB
3. *Harddisk* kosong 50 GB
4. Monitor 14"
5. *Mouse* dan *keyboard*
6. Satu unit *smartphone*

Sistem Android mengenal beberapa teknik penyimpanan data (*data store*). Teknik yang umum digunakan adalah sebagai berikut [4]:

1. *Shared Preference*, yaitu menyimpan data beberapa nilai (*value*) dalam bentuk *group key* yang dikenal dengan *preference*;

2. *Files*, yaitu menyimpan data dalam *file*, dapat berupa menulis ke *file* atau membaca dari *file*;
3. *SQLite Databases*, yaitu menyimpan data dalam bentuk *Databases*;
4. *Content Providers*, yaitu menyimpan data dalam bentuk *content providers service*.

Dalam penelitian ini peneliti menggunakan teknik *Files* untuk menyimpan data doa-doa dan koordinat lokasi fasilitas publik di Jakarta. Dan format *file* yang digunakan oleh peneliti adalah *JSON* (*JavaScript Object Notation*). Ada beberapa alasan kenapa peneliti menggunakan *file JSON* ini, yaitu:

1. *Font Arabic* yang nantinya akan ditampilkan pada halaman rincian doa tidak bisa disimpan di dalam *default databases* Android yaitu *SQLite Databases*, kecuali untuk Android versi 3.0 dan setelahnya;
2. *Font Arabic* bisa disimpan dalam format *JSON*;
3. Format *JSON* didukung oleh Android sehingga memudahkan pengambilan data.

```
[
  {
    "title": "Doa Menuju Masjid",
    "arabic": "اللَّهُمَّ اجْعَلْ فِي قَلْبِي نُورًا
    وَفِي بَصَرِي نُورًا وَفِي سَمْعِي نُورًا وَعَنْ يَمِينِي
    نُورًا وَعَنْ شِمَائِي نُورًا وَفَوْقِي نُورًا وَتَحْتِي نُورًا
    وَأَمَامِي نُورًا وَخَلْفِي نُورًا وَعَظَمَ لِي نُورٌ",
    "spell": "ALLAAHUMMAJ'AL FII QALBII
    NUURAN WA FII BASHARII
    NUURAN WA FII SAM'II NUURAN
    WA'AYYAMIINII NUURAN WA
    AYYASAARII NUURAN WA
    FAWQII NUURON WA TAHTII
    NUURAN WA AMAAMII NUURAN
    WA KHOLFII NUURON WA
    AZHZHOMLII NUURON.",
    "indo": "Wahai Tuhanku,
    Jadikanlah hatiku bercahaya, lisanku
    bercahaya, pendengaranku bercayaha,
    penglihatankupun bercahaya,
    belakanku bercahaya, hadapan
    bercahaya dan atas bawahku
    bercahaya. Wahai Tuhanku, berikanlah
```

```

aku cahaya"
    },
    {
        "title": "Doa Masuk Masjid",
        "arabic": "اللَّهُمَّ افْتَحْ لِي أَبْوَابَ رَحْمَتِكَ",
        "spell": "ALLAAHUMMAFTAH LII ABWAABA RAHMATIKA",
        "indo": "Wahai Tuhanku, bukakanlah pintu rahmat-Mu untukku"
    },
    {
        "title": "Doa Keluar Masjid",
        "arabic": "اللَّهُمَّ إِنِّي أَسْأَلُكَ مِنْ فَضْلِكَ",
        "spell": "Alloohumma innii asaluka min fadhlik",
        "indo": "Yaa Allah aku mohon kepada-Mu akan karunia-Mu"
    }
]

```

Koordinat lokasi fasilitas publik di Jakarta, penggunaan format JSON sebagai berikut:

```

[
    {
        "id": 0,
        "latitude": -6.306421,
        "longitude": 106.754587,
        "nama": "Masjid UINJKT"
    },
    {
        "id": 0,
        "latitude": -6.20128,
        "longitude": 106.831725,
        "nama": "Masjid Sunda Kelapa"
    },
    {
        "id": 0,
        "latitude": -6.235248,
        "longitude": 106.799469,
        "nama": "Masjid Agung Al-Azhar/Jaksel"
    }
]

```

Format JSON ini untuk beberapa file penyimpanan data doa meliputi:

- File* doas.json , yaitu *data store* untuk pencarian doa;
- File* masjid_doa.json, yaitu *data store* untuk doa-doa di masjid;
- File* sekolah_doa.json, yaitu *data store* untuk doa-doa di sekolah/kampus;
- File* kantor_doa.json, yaitu *data store* untuk doa-doa di kantor/tempat kerja;
- File* pasar_doa.json, yaitu *data store* untuk doa-doa di pasar/pusat perbelanjaan;
- File* terminal_doa.json, yaitu *data store* untuk doa-doa di terminal/stasiun/pelabuhan/bandara;
- File* rumah_makan_doa.json, yaitu *data store* untuk doa-doa di rumah makan;
- File* rumah_sakit_doa.json, yaitu *data store* untuk doa-doa di rumah sakit;
- File* makam.json, yaitu *data store* untuk doa-doa di pemakaman;

Untuk menampilkan data JSON di dalam Android digunakan *Class JSON*, berikut ini cara pengambilan data doa di Android:

```

BufferedReader jsonReader = new
BufferedReader(new
InputStreamReader(this.getResources().
getAssets().open("doas.json")));
StringBuilder jsonBuilder = new
StringBuilder();
for (String line = null; (line =
jsonReader.readLine()) != null;){

    jsonBuilder.append(line).append("\n");
}
//parse json
JSONTokener tokener = new
JSONTokener(jsonBuilder.toString());
JSONArray jsonArray = new
JSONArray(tokener);

for(int index=0; index <
jsonArray.length(); index++){
    //set both value into the listview
JSONObject jsonObject =

```

```

JSONArray.getJSONObject(index);
    String title =
JSONObject.getString("title");
    String arabic =
JSONObject.getString("arabic");
    String spell =
JSONObject.getString("spell");
    String indo =
JSONObject.getString("indo");

    // creating new HashMap
    HashMap<String, String> map = new
    HashMap<String, String>();

    //then each child node to Hashmap
    key => value
        map.put("title", title);
        map.put("arabic",
arabic);

        map.put("spell", spell);

        map.put("indo", indo);
        //adding HashList to
        ArrayList

        doasList.add(map);
    }

```

Antarmuka yang sudah dirancang sebelumnya selanjutnya dibuat *working prototype*-nya untuk selanjutnya dilakukan *coding* sehingga dapat dijalankan sesuai dengan skenario sebelumnya. Antarmuka yang diimplementasikan antara lain:

1. Tampilan *Splash Screen*

Setelah menekan *icon launcher* aplikasi ‘Doa Dimana?’ pada layar *smartphone* Android, maka yang pertama kali akan ditampilkan adalah *splash screen* yang berupa animasi teks dan gambar.



Gambar 11. Tampilan *Splash Screen*

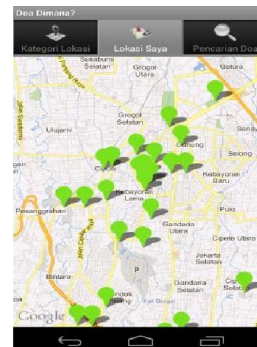
2. Tampilan Menu Utama

Tampilan menu utama terdiri dari tiga tab menu utama, yaitu tab menu kategori lokasi, tab menu lokasi saya, dan tab menu pencarian doa. Adapun tampilannya seperti gambar berikut:



Gambar 12. Tampilan Menu Utama

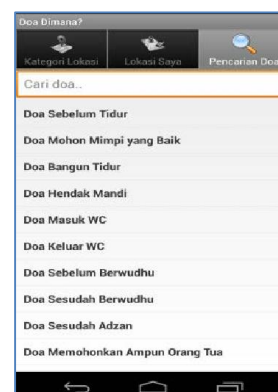
3. Tampilan Tab Menu “Lokasi Saya”



Gambar 13. Tampilan Tab Menu “Lokasi Saya”

Pada menu ini terdapat fitur *location based service* yang memberikan penjelasan posisi atau lokasi dimana pengguna berada.

4. Tampilan Tab Menu “Pencarian Doa”



Gambar 14. Tampilan Tab Menu “Pencarian Doa”

5. Tampilan Tab Menu “Doa Per Kategori”



Gambar 15. Tampilan Tab Menu “Doa Per Kategori”

6. Tampilan Menu “Detail Doa”



Gambar 16. Tampilan Menu “Detail Doa”

7. Tampilan Pesan “Notifikasi Doa”



Gambar 17. Tampilan “Notifikasi Doa”

8. Tampilan Pengaturan GPS



Gambar 18. Tampilan Pengaturan GPS

III. IMPLEMENTASI ALGORITMA

BRUTE FORCE

Cara kerja algoritma *Brute Force* pencarian doa pada aplikasi “Doa Dimana?” ini (sesuai dengan *pseudocode* di atas) akan dijelaskan dengan contoh sebagai berikut :

1. Pertama, aplikasi akan membaca data doa yang berada dalam file JSON yang berisi judul doa, selanjutnya disebut sebagai teks. Contoh : Doa masuk masjid;

T = Doa masuk masjid

m = 16

2. Kemudian aplikasi menangkap karakter atau string yang diketikkan oleh pengguna aplikasi, dalam bahasan ini disebut sebagai *pattern*. Contoh : masjid;

P = masjid

n = 6

3. Kemudian *pattern* disejajarkan dengan teks pada posisi paling kiri sehingga karakter pertama pada *pattern* sejajar vertikal dengan karakter pertama pada teks;

T = D o a m a s u k m a s j i d

P = m a s j i d

4. Pada langkah pertama (s = 0) didapatkan karakter “m” pada *pattern* sejajar dengan karakter “D” pada teks;

s = 0 D o a m a s u k m a s j i d
m a s j i d

D ≠ m

5. Karena terdapat ketidakseamaan maka geser *pattern* satu langkah ke kanan terhadap teks (s = 1). Didapat karakter “o” pada teks sejajar dengan karakter “m” pada *pattern*.

s = 1 D o a m a s u k m a s j i d
m a s j i d

o ≠ m

6. Ditemukan ketidakcocokan di sini, geser kembali satu langkah ke kanan terhadap teks.
7. Lakukan pergeseran *pattern* hingga ditemukan kecocokan antar karakter *pattern* dan karakter teks.

8. Di langkah kelima ($s = 4$), ditemukan kecocokan karakter “m” pada *pattern* dengan karakter “m” pada teks;

$$s = 4$$

D	o	a	m	a	s	u	k	m	a	s	j	i	d
			m	a	s	j	i	d					

$m = m$

$P[j] = T[s+j] \Rightarrow P[1] = T[5]$

maka nilai $j = j+1 \Rightarrow j = 2$

$$s = 4$$

D	o	a	m	a	s	u	k		m	a	s	j	i	d
			m	a	s	j	i	d						

$a = a$

$P[2] = T[6]$

maka nilai $j = 2+1 \Rightarrow j = 3$

$$s = 4$$

D	o	a	m	a	s	u	k		m	a	s	j	i	d
			m	a	s	j	i	d						

$s = s$

$P[3] = T[7]$

maka nilai $j = 3+1 \Rightarrow j = 4$

$j \neq u$

$P[4] \neq T[8]$

$$s = 4$$

D	o	a	m	a	s	u	k		m	a	s	j	i	d
			m	a	s	j	i	d						

9. Karena pada langkah kelima ($s = 4$), dengan nilai $j=4$ ditemukan ketidakcocokan antara karakter keempat “j” pada *pattern* dengan karakter “u” pada teks. Maka geser *pattern* satu karakter ke kanan terhadap teks ($s = 5$), sehingga karakter pertama *pattern* “m” sejajar dengan karakter keenam teks yaitu karakter “a”;

$$s = 5$$

	D	o	a	m	a	s	u	k	m	a	s	j	i	d
				m	a	s	j	i	d					

$m \neq a$

10. Pada langkah keenam ($s = 5$) hingga langkah kesepuluh ($s = 9$), tidak ditemukan kecocokan;

11. Pada langkah kesebelas ($s = 10$), ditemukan kecocokan sebagai berikut:

$$s = 10$$

									m	a	s	j	i	d
									m	a	s	j	i	d

$m = m$

$P[j] = T[s+j] \Rightarrow P[1] = T[11]$

maka nilai $j = j+1 \Rightarrow j = 2$

$$s = 10$$

									m	a	s	j	i	d
									m	a	s	j	i	d

$a = a$

$P[2] = T[12]$

maka nilai $j = 2+1 \Rightarrow j = 3$

$$s = 10$$

									m	a	s	j	i	d
									m	a	s	j	i	d

$s = s$

$P[3] = T[13]$

Maka nilai $j = 3+1 \Rightarrow j = 4$

$$s = 10$$

									m	a	s	j	i	d
									m	a	s	j	i	d

$j = j$

$P[4] = T[14]$

Maka nilai $j = 4+1 \Rightarrow j = 5$

$$s = 10$$

									m	a	s	j	i	d
									m	a	s	j	i	d

$i = i$

$P[5] = T[15]$

Maka nilai $j = 5+1 \Rightarrow j = 6$

$$s = 10$$

									m	a	s	j	i	d
									m	a	s	j	i	d

$d = d$

$P[6] = T[16]$

Maka nilai $j = m$

Dan nilai *variable* ketemu = true

12. Jadi pada langkah kedua belas ini ($s = 10$), *pattern* = masjid ditemukan dalam teks = Doa masuk masjid .

Berikut ini rangkuman langkah-langkah di atas:

Teks : Doa masuk masjid

Pattern : masjid

	D	o	a		m	a	s	u	k		m	a	s	j	i	d	
s=0	m	a	s	j	i	d											
s=1		m	a	s	j	i	d										
s=2			m	a	s	j	i	d									
s=3				m	a	s	j	i	d								
s=4					m	a	s	j	i	d							
s=5						m	a	s	j	i	d						
s=6							m	a	s	j	i	d					
s=7								m	a	s	j	i	d				
s=8									m	a	s	j	i	d			
s=9										m	a	s	j	i	d		
s=10											m	a	s	j	i	d	

Langkah selanjutnya, algoritma Brute Force pencarian *string* di atas diubah menjadi *code* dalam bahasa pemrograman Java Android sebagai berikut:

```
// method string match dengan metode
brute force

private void
bruteForce(ArrayList<HashMap<String,
String>> doasList_r){
    String s =
    et.getText().toLowerCase();
    int p =
    et.getText().length();

    doasList_sort.clear();
    if(p==0){
        ListAdapter
        adapter = new SimpleAdapter(this,
        doasList_r,

        R.layout.list_item, new String[]
        {"title", "arabic", "spell", "indo"}, new
        int[] {R.id.title, R.id.arabic,
        R.id.spell, R.id.indo});
        setListAdapter(adapter);
    } else {
        for(int i=0;i<
        doasList_r.size();i++){ //looping max
        panjang doaslist
            //Log.d("BRUTE", "jml_array_doa =
            "+doasList_r.size());

            HashMap<String, String> map =
            doasList_r.get(i);

            for(Entry<String, String>
            mapEntry: map.entrySet()) {
                String key =
                mapEntry.getKey();
                String value =
                mapEntry.getValue().toLowerCase();
                if(key=="title"){
```

```
int t =
value.length();

for(int
// j=mewakili
karakter pada text
int
//
k=0;
k=mewakili karakter pada pattern

while((k<p)&&(value.charAt(j+k)==s.char
At(k)))
k++;

if(k==p){
doasList_sort.add(map);

break; // jika pattern sudah
ditemukan sekali, pencocokan karakter
selesai.
}

// tanpa harus mencocokkan hingga
karakter terakhir (per judul doa)
}
}
}
ListAdapter
adapter = new SimpleAdapter(this,
doasList_sort,

R.layout.list_item, new String[]
{"title", "arabic", "spell", "indo"}, new
int[] {R.id.title, R.id.arabic,
R.id.spell, R.id.indo});

setListAdapter(adapter);

//selecting single listview item
ListView
listView = getListView();

//launcing new
screen(view doa detail) on selecting single
listitem

listView.setOnItemClickListener(
new OnItemClickListener() {

public
void onItemClick(AdapterView<?> parent,
View view, int position,

long id) {

// TODO Auto-generated method
stub

//geting values from selected
```

listitem

```

        String title = ((TextView)
view.findViewById(R.id.title)).getText().to
String();

        String arabic = ((TextView)
view.findViewById(R.id.arabic)).getText()
.toString();

        String spell = ((TextView)
view.findViewById(R.id.spell)).getText().t
oString();

        String indo = ((TextView)
view.findViewById(R.id.indo)).getText().t
oString();
        //starting new item

        Intent in = new
Intent(getApplicationContext(),
DoaActivity.class);

        in.putExtra("title", title);

        in.putExtra("arabic", arabic);
in.putExtra("spell", spell);

        in.putExtra("indo", indo);

        startActivity(in);

    }
    });
}

```

IV. KESIMPULAN

Aplikasi kumpulan doa harian dapat dikembangkan dengan mengikuti metode pengembangan system *Rapid Application Development* (RAD) yang terdiri dari tiga fase yaitu perencanaan syarat, *workshop design*, dan implementasi. Pada menu pencarian doa, digunakan algoritma *Brute Force* untuk mencocokkan string. Algoritma ini cukup sederhana untuk diimplementasikan pada aplikasi ini.

REFERENSI

- [1] Ali Al-Kasyaf, Amrin, 2012, *Kitab Doa Sehari-hari Terlengkap*, Diva Press, Jogjakarta
- [2] Kendall, K.E., dan Kendall, J.E. 2008. *System Analisis and Design 7th Edition*. New Jersey: Prentice Hall.
- [3] Munir, R. 2004. *Strategi Algoritmik, Algoritma Pencarian String (String Matching)*. Departemen Teknik Informatika Institut Teknologi Bandung: Bandung.
- [4] Safaat H, Nazruddin. 2012. *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC berbasis Android Edisi Revisi*. Informatika: Bandung.