

PEMANFAATAN GOOGLE SPEECH RECOGNATION DAN TEXT TO SPEECH DALAM PENCARIAN BUKU BERBASIS ANDROID

Bambang Yuwono (1), Yudha Satria (2) Hidayatulah Himawan (3)

Jurusan Teknik Informatika, UPN "Veteran" Yogyakarta

Jl. Babarsari No 2, Tambakbayan, Yogyakarta

Email : bambangy@gmail.com

Abstrak

Proses Pencarian buku sebagian besar masih dilakukan secara manual, Pencari buku pustaka harus bertanya dulu ke petugas kemudian petugas mencari lewat database komputer setelah data buku yang dicari ditemukan baru petugas memberi tahu pencari buku pustaka letak buku yang sedang dicari. Dengan memanfaatkan Speech Recognition (Speech Input) dan Text-to-Speech API pada sistem operasi Android, proses pencarian dapat dilakukan dengan lebih mudah. Perangkat Lunak (software) yang digunakan untuk membangun aplikasi ini adalah Android Studio dan sistem operasi Android versi 5.0 (Lollipop). Tujuan dari penelitian ini adalah untuk menghasilkan aplikasi yang berfungsi untuk membantu pengguna dalam mencari buku dengan melakukan penginputan dan pengecekan suara. Metode grapple digunakan untuk melaksanakan penelitian ini.

Kata kunci : *Text to speech, Google Speech, Recognition, Android*

1. PENDAHULUAN

Peningkatan profesionalisme dan kualitas pendidikan serta pelayanan yang ditawarkan suatu lembaga pendidikan merupakan salah satu daya tarik yang menjadi perhatian masyarakat sebagai salah satu kriteria dalam memilih suatu lembaga pendidikan. UPN Veteran Yogyakarta merupakan salah satu perguruan tinggi negeri yang mempunyai prestasi yang baik di Yogyakarta. Dalam melakukan penelitian di UPN Veteran Yogyakarta menemukan permasalahan yaitu pada bagian perpustakaan Fakultas Teknologi Industri (FTI) belum adanya sistem informasi dalam pencarian buku di perpustakaan tersebut. Pencarian buku pustaka pada perpustakaan FTI UPN Veteran Yogyakarta masih dilakukan secara manual. Pencarian buku pustaka harus bertanya dulu ke petugas perpustakaan kemudian petugas perpustakaan tersebut mencari data buku melalui *database* komputer, setelah data buku yang dicari ditemukan baru petugas memberi tahu pencari buku pustaka letak buku yang sedang dicari. Oleh karena itu, sangat perlu aplikasi pemanfaatan *Google Speech Recognition* dan *Text To Speech* dalam pencarian buku perpustakaan berbasis Android.

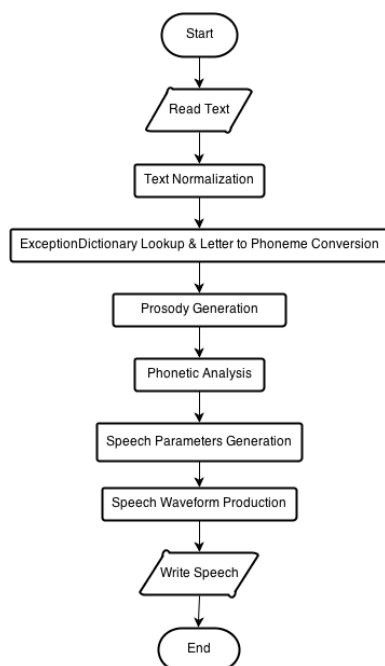
2. TINJAUAN PUSTAKA

Penelitian "Aplikasi Pemanfaatan *Google Speech Recognition* dan *Text To Speech* dalam Pencarian Buku Perpustakaan Berbasis Android" merupakan penelitian yang memanfaatkan teknologi dari sebuah sistem operasi pada smartphone, yakni Android. Penelitian ini memanfaatkan fasilitas *Google Speech Recognition* dan *Text To Speech* yang sudah tersedia pada Android. *Text To Speech* merupakan teknologi yang mengkonversi teks menjadi suara, sedangkan *Google Speech Recognition* merupakan teknologi yang mengkonversi suara menjadi teks. Penelitian ini dilaksanakan berdasarkan pada studi pustaka dari penelitian sebelumnya, yakni "Pembangunan Aplikasi Berbasis Suara (*Voice Based Application*) Pada Perpustakaan Dengan Menggunakan Teknologi *Voicexml*" oleh mahasiswa Jurusan Teknik Informatika Universitas Atma Jaya Yogyakarta, Andrianto Ficka Sanggara (2007).

3. METODE PENELITIAN

Perancangan sistem menggunakan metode GRAPPLE (*Guidelines for Rappid Application Engineering*). Metode ini merupakan pemodelan proses pengembangan perangkat lunak (*software*) yang menitikberatkan pada aksi-aksi yang dilakukan pada sejumlah tahap, dimana setiap tahap akan menghasilkan *output* berorientasi objek (Schmuller, 1999). Terdapat lima tahapan dalam metode GRAPPLE ini, antara lain *Requirement Gathering, Analysis, Design, Development, Deployment*.

Flowchart Text-to-Speech



Gambar 1. Flowchart Text to Speech (Pelton, 1992)

Text to speech (TTS) diartikan sebagai proses perubahan teks menjadi audio digital dan diucapkan. Pengucapan ini dapat berupa pengiriman audio digital tersebut ke pengeras suara computer atau menyimpan hasil perubahan tersebut untuk diputar nanti. Dalam mengubah teks menjadi audio (Akhmad, A.,2012). Pada bagian *text to phoneme* berfungsi untuk mengubah input teks menjadi rangkaian kode-kode bunyi yang direpresentasikan dengan kode fonem, durasi serta *pitch*-nya. *Text to phoneme* ditunjukkan pada bagian *text normalization*, *exception dictionary lookup & letter to phoneme conversion*, *prosody generation*, dan *phonetic analysis*. Sedangkan bagian *phoneme to speech* akan menerima input berupa kode-kode fonem serta *pitch* dan durasi hasil dari bagian sebelumnya. Berdasarkan kode-kode tersebut bagian *phoneme to speech* akan menghasilkan bunyi sinyal ucapan yang sesuai dengan teks yang ingin diucapkan. Bagian *phoneme to speech* ditunjukkan pada *speech parameters generation* dan *speech waveform production*.

Tahapan-tahapan konversi teks menjadi suara adalah :

a. *Text Normalization*, Tahap ini berfungsi untuk mengubah semua teks yang ingin diucapkan menjadi teks yang lengkap memperlihatkan cara pengucapannya.

b. *Exception Dictionary Lookup & Letter to Phoneme Conversion* , Pada tahap ini dilakukan konversi dari teks yang sudah lengkap mempresentasikan kalimat yang ingin diucapkan menjadi kode-kode fonem. Konversi dilakukan dengan dua cara, yaitu *Exception Dictionary Lookup* dan *Letter to Phoneme Conversion*. Konversi yang tidak teratur ditangani oleh bagian *Exception Dictionary Lookup*, sedangkan *Letter to Phoneme Conversion* menangani kondisi yang teratur.

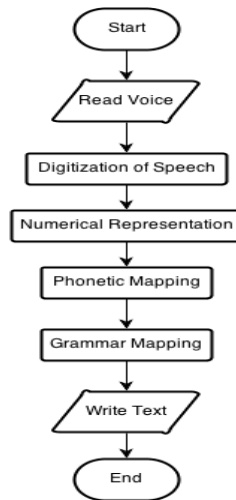
c. *Prosody Generator* , Bagian ini melengkapi setiap unit fonem yang dihasilkan dengan data durasi serta *pitch*-nya. Data durasi serta *pitch*-nya diperoleh berdasarkan kombinasi antara tabel atau database serta model prosodi. Hasil dari bagian ini sudah cukup menghasilkan informasi dari ucapan yang diinginkan.

d. *Phonetic Analysis*, Tahap ini adalah tahap penyempurnaan, yaitu melakukan perbaikan di tingkat bunyi. Setiap teks dikonversi menjadi fonemnya. 15

e. *Speech Parameters Generation*, Tahap ini adalah tahap pencarian padanan fonem dari kata. Pada tahap ini kata akan ditelaah konteksnya, dan dicari padanan fonemnya, misalnya untuk kata tanya akan berbeda pengucapannya dengan kata perintah.

f. *Speech Waveform Production*, Tahap ini adalah tahap terakhir, yaitu membangkitkan bentuk gelombang atau bentuk yang dapat didengar dari data fonem dengan menggunakan elemen audio dan aturan penggabungan elemen-elemen fonem berdasarkan sistem logika dan kecerdasan buatan.

Flowchart Speech Recognition



Gambar 2. Flowchart Speech Recognition (Shukla dkk, 2002)

Gambar 2 adalah predefined process dari speech recognition. Speech recognition terdiri dari empat proses utama yaitu (Shukla dkk, 2002) :

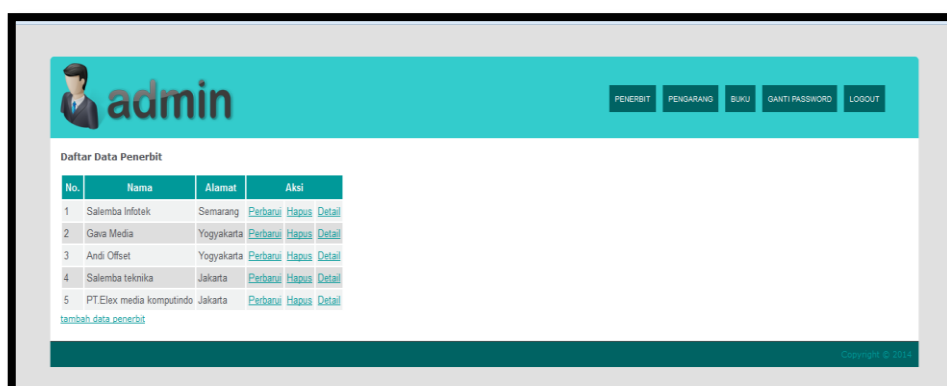
- Digitization of Speech*, Tahap ini adalah tahap pertama dalam *speech recognition*, yaitu menangkap suara pengguna dan mengubahnya menjadi format digital.
- Numerical Representation*, Tahap ini mentransformasi data digital hasil *digitization of speech* menjadi susunan nomor, yang mengandung kombinasi *pitch* dan informasi detail mengenai tingkat *pitch* pada setiap sesi.
- Phonetic Mapping*, Pada tahap ini susunan nomor yang dihasilkan pada proses sebelumnya dipasangkan dengan fonem yang paling mendekati nilainya. Tahap ini adalah tahap yang paling penting karena akan secara langsung berpengaruh pada keakuratan sistem pengenalan suara. Hasil dari *phonetic mapping* ini terlihat pada *string* fonem yang setiap *string*-nya memiliki nilai tersendiri sesuai dengan *text* yang telah diucapkan oleh pengguna.
- Grammar Mapping*, Pada tahap terakhir ini, susunan fonem dipasangkan lagi dengan *grammar library* yang tersedia untuk memilih nilai yang paling mendekati dari *text* yang dilafalkan oleh *user*.

4. HASIL DAN PEMBAHASAN

Berdasarkan hasil dari perancangan dengan menggunakan metode GRAPPLE dalam membangun aplikasi, dilakukanlah implementasi Aplikasi Pemanfaatan *Google Speech Recognition dan Text-to-Speech* Dalam Pencarian Buku Perpustakaan Berbasis Android. Selain itu juga dilakukan implementasi terhadap aplikasi halaman web admin yang dibangun menggunakan bahasa pemrograman PHP.

Halaman Penerbit

Halaman yang menampilkan daftar data penerbit. Halaman ini dapat menambah data penerbit serta dapat memperbaharui dan menghapus data penerbit. pada menu ini terdapat beberapa pilihan diantaranya: Nama Penerbit, Alamat, Aksi (Perbaharui,Hapus,detail).



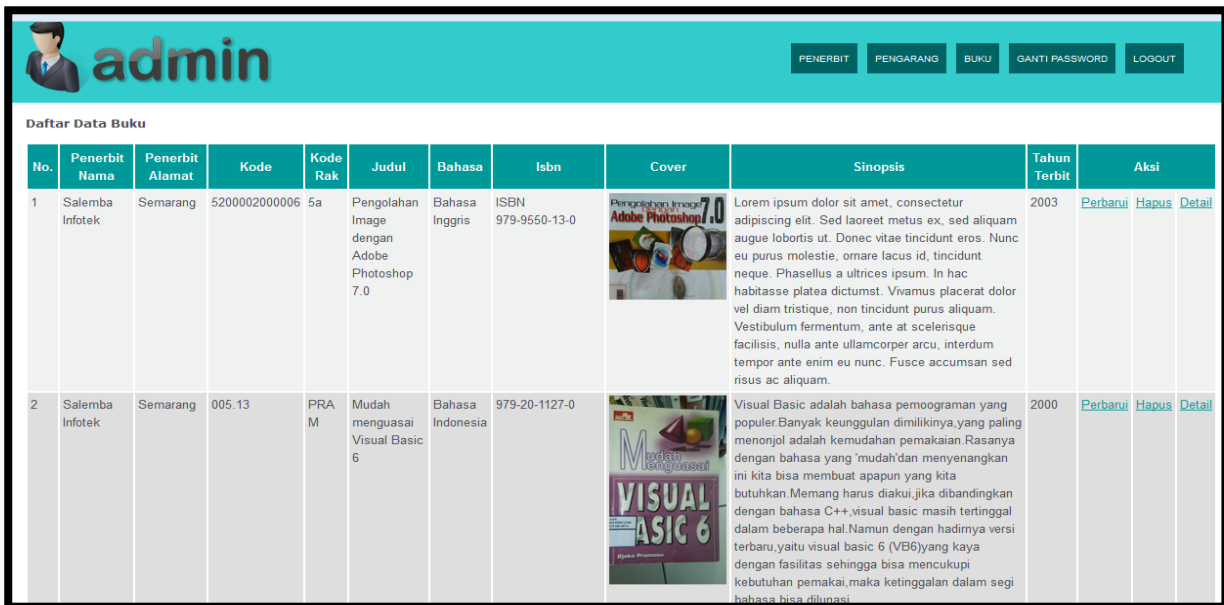
Gambar 3. Tampilan Menu Penerbit

```
?>
<tr class="<?php echo $odd_even;?>">
  <td><?php echo $no;?></td>
  <td><?php echo $row->penerbit_nama;?></td>
  <td><?php echo $row->penerbit_alamat;?></td>
  <td><a href="<?php echo base_url().'penerbit/index/ubahdata/'. $row->id_penerbit;?>">Perbarui</a></td>
  <td><a href="<?php echo base_url().'penerbit/updatestatus/'. $row->id_penerbit;?>/0" onclick="return confirm('Benar Anda ingin Menghapus Data ini?')">Hapus</a></td>
  <td><a href="<?php echo base_url().'penerbit/details/'. $row->id_penerbit;?>">Detail</a></td>
</tr>
```

Modul Program 1. Source Code penerbit_data.php

Halaman Buku

Halaman yang menampilkan data buku. Halaman ini dapat menambah data penerbit serta dapat memperbahruhi dan menghapus data buku.

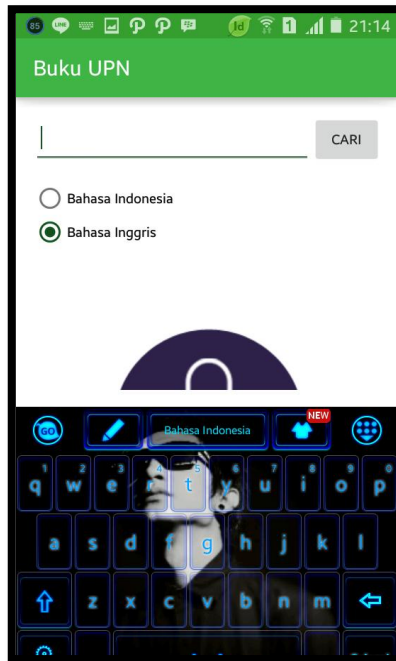


Gambar 4. Tampilan Menu Buku

```
?>
<tr class="<?php echo $odd_even;?>">
  <td><?php echo $no;?></td>
  <td><?php echo $row->penerbit_nama;?></td>
  <td><?php echo $row->penerbit_alamat;?></td>
  <td><?php echo $row->buku_kode;?></td>
  <td><?php echo $row->buku_kode_rak;?></td>
  <td><?php echo $row->buku_judul;?></td>
  <td><?php echo $bahasaArray[$row->buku_bahasa];?></td>
  <td><?php echo $row->buku_isbn;?></td>
  <td></td>
  <td><?php echo $row->buku_sinopsis;?></td>
  <td><?php echo $row->buku_tahun_terbit;?></td>
  <td><a href="<?php echo base_url().'buku/index/ubahdata/'. $row->id_buku;?>">Perbarui</a></td>
  <td><a href="<?php echo base_url().'buku/updatestatus/'. $row->id_buku;?>/0" onclick="return confirm('Benar Anda ingin Menghapus Data ini?')">Hapus</a></td>
  <td><a href="<?php echo base_url().'buku/details/'. $row->id_buku;?>">Detail</a></td>
</tr>
```

Modul Program 2. Source Code buku_data.php

Tampilan Menu Pencarian



Gambar 5. Tampilan Menu Pencarian

```
public class PencarianActivity extends AppCompatActivity {
    private static final int TAG_SPEECH_CODE = 100;
    Context mContext;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_pencarian);
        mContext = this;
        try {
            String[] kata = new SemuaKata().execute().get();
            AutoCompleteTextView actv1 = (AutoCompleteTextView) findViewById(R.id.actvPencarian1);
            ArrayAdapter<String> adapter = new ArrayAdapter<>(this,
                android.R.layout.simple_dropdown_item_1line, kata);
            actv1.setAdapter(adapter);
        } catch (InterruptedException | ExecutionException ignored) {}
        final RadioGroup rg1 = (RadioGroup) findViewById(R.id.rgPencarian1);
        rg1.check(R.id.rbPencarian1);
        Button btn1 = (Button) findViewById(R.id.btnPencarian1);
        btn1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                AutoCompleteTextView actv1 = (AutoCompleteTextView) findViewById(R.id.actvPencarian1);
                String keyword = actv1.getText().toString();
                String bahasa = "id_ID";
                Buku b = new Buku();
                switch (rg1.getCheckedRadioButtonId()) {
                    case R.id.rbPencarian1:
                        b.setBahasa("in");
                        bahasa = AplikasiHelper.bahasa(b);
                        break;
                    case R.id.rbPencarian2:
                        b.setBahasa("en");
                        bahasa = AplikasiHelper.bahasa(b);
                        break;
                }
                if (!keyword.equalsIgnoreCase("")) {
                    Intent i = new Intent(mContext, HasilPencarianActivity.class);
                    i.putExtra(HasilPencarianActivity.TAG_KEYWORD, keyword);
                    i.putExtra(HasilPencarianActivity.TAG_BAHASA, bahasa);
                    startActivity(i);
                }
            }
        });
    }
}
```

Modul Program 3.1 Source Code PencarianActivity.java

```
}    }    });  
    ImageButton ibtn1 = (ImageButton) findViewById(R.id.ibtnPencarian1);  
    ibtn1.setOnClickListener(new View.OnClickListener() {  
        @Override  
        public void onClick(View v) {  
            Intent intent = new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH);  
            intent.putExtra(RecognizerIntent.EXTRA_LANGUAGE_MODEL,  
                RecognizerIntent.LANGUAGE_MODEL_FREE_FORM);  
            String bahasa = "id_ID";  
            Buku b = new Buku();  
            switch (rg1.getCheckedRadioButtonId()) {  
                case R.id.rbPencarian1:  
                    b.setBahasa("in");  
                    bahasa = AplikasiHelper.bahasa(b);  
                    break;  
                case R.id.rbPencarian2:  
                    b.setBahasa("en");  
                    bahasa = AplikasiHelper.bahasa(b);  
                    break;  
            }  
        }  
    });  
    @Override  
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
        super.onActivityResult(requestCode, resultCode, data);  
        switch (requestCode) {  
            case TAG_SPEECH_CODE: {  
                if (resultCode == RESULT_OK && null != data) {  
                    ArrayList<String> result = data.getStringArrayListExtra(RecognizerIntent.EXTRA_RESULTS);  
                    String hasil = result.get(0);  
                    AutoCompleteTextView actv1 = (AutoCompleteTextView) findViewById(R.id.actvPencarian1);  
                    actv1.setText(hasil);  
                    Button btn1 = (Button) findViewById(R.id.btnPencarian1);  
                    btn1.performClick();  
                }  
                break;  
            }  
        }  
    }  
    private class SemuaKata extends AsyncTask<String, Void, String[]> {  
        private static final String TAG_RESULT = "result";  
        private static final String TAG_RESULT_TRUE = "true";  
        private static final String TAG_DATA = "data";  
    }  
}
```

Modul Program 3.2 Lanjutan Source Code PencarianActivity.java

5. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan, terciptalah sebuah Aplikasi Pencarian Buku Perpustakaan FTI Upn "Veteran" Yogyakarta dengan memanfaatkan *Text-to-Speech* dan *Speech Recognition* berbasis Android. Aplikasi ini dapat mempermudah *user* dalam mencari buku dengan mengkonversi kata menjadi suara yang diucapkan dalam Bahasa Indonesia dan Bahasa Inggris untuk mencari buku yang tersedia. Dengan demikian, aplikasi ini bermanfaat untuk meningkatkan memudahkan *user* dalam mencari buku.

DAFTAR PUSTAKA

Arry, 2008, *Text To Speech*, <https://kupalima.wordpress.com/2008/08/16/text-to-speech-dan-pemodelan-intonasi/>, (diakses pada tanggal 22 januari 2016).

Akhmad, A. *Teknologi Pemrosesan Bahasa Alami sebagai Teknologi Kunci untuk Meningkatkan Cara Interaksi antara Manusia dengan Mesin*. [Online]

Unduh:

http://www.itb.ac.id/focus/focus_file/Pidato%20Ilmiah%20pada%20Sidang%20Terbuka%20PMB%202004.pdf

Jogiyanto Hartono, 1999, *Analisis & Desain Sistem Informasi*, Penerbit Andi, Yogyakarta.

Kadir, A., 2008. *Perkenalan System Informasi*, Andi, Yogyakarta.

Kristanto, Harianto. Ir. 1994. *Konsep dan Perancangan Database*. Yogyakarta: Andi Offset

Nugroho, Riant. 2003.*KEBIJAKAN PUBLIK Formulasi, Implementasi, dan Evaluasi*, Jakarta. Elex Media Komputindo

Schmuller, Joseph. 1999..*Sams Teach Yourself UML in 24 Hours*. Pearson Education, Inc.

Tessy.,2011. *Menggunakan UML*, http://id.wikipedia.org/wiki/Diagram_alir, (diakses pada tanggal 20 September 2015).

Yuniar. Supardi, 2011. *Semua Bisa Menjadi Programmer VB6 Hingga VB 2008 Case Study*. Jakarta: PT Elex Media Komputindo.