

J-INTTECH

Journal of Information and Technology

Volume 07 Nomor 01, Bulan Juni Tahun 2019



STIKI

SEKOLAH TINGGI INFORMATIKA & KOMPUTER INDONESIA

Jl. Raya Tidar 100 Malang, 65146

Telp. (0341)560823, Fax (0341)562525

ISSN: 2303-1425 E-ISSN: 2580-720X

J-INTECH

Journal of Information and Technology

Volume 07 Nomor 01, Bulan Juni 2019



LEMBAGA PENELITIAN DAN PENGABDIAN KEPADA MASYARAKAT

STIKI

SEKOLAH TINGGI INFORMATIKA & KOMPUTER INDONESIA

Jl. Raya Tidar 100, Malang; Phone: 0341-560823; Fax: 0341-562525; <http://www.stiki.ac.id>; mail@stiki.ac.id

PENGANTAR REDAKSI

J-INTECH merupakan jurnal yang diterbitkan oleh Sekolah Tinggi Informatika dan Komputer Indonesia Malang guna mengakomodasi kebutuhan akan perkembangan Teknologi Informasi serta guna mensukseskan salah satu program DIKTI yang mewajibkan seluruh Perguruan Tinggi untuk menerbitkan dan mengunggah karya ilmiah mahasiswanya dalam bentuk terbitan maupun jurnal online.

Pada edisi ini, redaksi menampilkan beberapa karya ilmiah mahasiswa yang mewakili beberapa mahasiswa yang lain, yang dianggap cukup baik sebagai media pembelajaran bagi para lulusan selanjutnya.

Tentu saja diharapkan pada setiap penerbitan memiliki nilai lebih dari karya ilmiah yang dihasilkan sebelumnya sehingga merupakan nilai tambah bagi para adik kelas maupun pihak-pihak yang ingin studi atau memanfaatkan karya tersebut selanjutnya.

Pada kesempatan ini kami juga mengundang pihak-pihak dari PTN/PTS lain sebagai kontributor karya ilmiah terhadap jurnal J-INTECH, sehingga Perkembangan IPTEK dapat dikuasai secara bersama-sama dan membawa manfaat bagi institusi masing-masing.

Akhir redaksi berharap semoga dengan terbitnya jurnal ini membawa manfaat bagi para mahasiswa, dosen pembimbing, pihak yang bekerja pada bidang Teknologi Informasi serta untuk perkembangan IPTEK di masa depan.

REDAKSI

DAFTAR ISI

Sistem Informasi Penyedia Pemandu Wisata dengan Metode <i>Simple Additive Weighting (SAW)</i> Berbasis Android.....	01-16
<i>Rizky Fitriyanto, Eva Handriyantini, Johan Ericka W.P.</i>	
Rancangan Bangun Pelayanan Tambal Ban Online Menggunakan Algoritma <i>Dijkstra</i>	17-34
<i>Muhammad Henry Setiawan, Subari</i>	
Sistem Pendukung Keputusan Rumah Tangga Miskin (RTM) untuk Program Rastra Di Desa Lolo Menggunakan Metode <i>Weighted Product</i> Berbasis Web.....	35-43
<i>Fitri Dayanti, Sugeng Widodo</i>	
Sistem Pakar Diagnosa Penyakit Pada Hewan Ternak Sapi Menggunakan <i>Case Based Reasoning (CBR)</i> Berbasis Android.....	44-57
<i>Rachmania Indah Permata Sari, Diah Arifah Prastiningtyas, Subari</i>	
<i>Game Multiplayer "Mini Car Circuit"</i> Berbasis Android.....	58-66
<i>Syaifuddin Yudha Saputra, Subari</i>	
Sistem Penunjang Keputusan Simulasi Penentuan Akreditasi Sekolah Dasar Menggunakan Metode <i>Analytic Network Process (ANP)</i> Berbasis Web (Studi Kasus Korwil Dinas Pendidikan Kecamatan Pakisaji)	67-71
<i>Ahmad Syaifullah, Laila Isyriyah, Bagus Kristomoyo Kristanto</i>	
Penerapan Algoritma <i>Nearest Neighbor</i> Untuk Menentukan Rekomendasi Solusi Terhadap Layanan Kantor Teknologi Informasi STIKI Malang	72-79
<i>Yosua Kristanto, Diah Arifah Prastiningtyas, Meivi Kartikasari</i>	
Klasifikasi <i>E-book</i> Berbahasa Inggris Dengan Menggunakan Metode <i>K-Means Clustering</i> Studi Kasus Perpustakaan STIKI Malang	80-85
<i>Willyanto Sutikno, Jozua Ferjanus Palandi, Chaulina Alfianti Oktavia</i>	
Aplikasi Presensi Mahasiswa Berbasis Wifi <i>Direct Peer to Peer</i> pada Perangkat Android (Studi Kasus : STIKI Malang).....	86-89
<i>Khoirun Nafisah, Koko Wahyu Prasetyo</i>	

Penerapan Metode Jaringan Syaraf Tiruan Radial Basis Function Untuk Identifikasi
Jenis Mangga Berdasarkan Pola Daun 90-96
*Rhesal Mahadyanto, Diah Arifah Prastiningtyas, Febry Eka
Purwiantono*

ISSN: 2303-1425 E-ISSN: 2580-720X

J-INTTECH

Journal of Information and Technology
Volume 07 Nomor 01, Bulan Juni 2019

Pelindung

Yayasan Perguruan Tinggi Teknik Nusantara

Penasehat

Ketua STIKI

Pembina

Kepala Program Studi

Editor In Chief

Subari, S.Kom, M.Kom

Section Editor

Siti Aminah, S.Si, M.Pd

Layout Editor

Nira Radita, S.Pd., M.Pd

Tata Usaha/Administrasi

Muh. Bima Indra Kusuma

Rancangan Bangun Pelayanan Tambal Ban *Online* Menggunakan Algoritma *Dijkstra*

Muhammad Henry Setiawan¹, Subari²

^{1,2}Program Studi Teknik Informatika, Sekolah Tinggi Informatika & Komputer Indonesia (STIKI) Malang

¹own.cuit@gmail.com, ²subari@stiki.ac.id

ABSTRAK

Tidak selalu tersedianya tambal ban di sepanjang jalur yang kerap kali kita lalui sering kali menyusahkan pengguna sepeda motor yang mengalami kebocoran oleh karena itu di perlukan informasi yang dapat memudahkan para pengguna kendaraan mencari informasi secara cepat dan tepat. Dengan ilmu perkembangan teknologi saat ini telah dikenal adanya *geographical Information System (GIS)* adalah sistem informasi khusus yang mengelola data yang memiliki informasi spasial (*bereferensi keruangan*). informasi spasial adalah sebuah data yang berorientasi geografis, memiliki sistem koordinat tertentu sebagai dasar referensinya dan mempunyai dua bagian penting yang membuatnya berbeda dari data lain. Atau dalam arti yang lebih sempit adalah sistem komputer yang memiliki kemampuan untuk membangun, menyimpan, mengelola dan menampilkan informasi *bereferensi geografis*. Dengan lokasi tambal ban yang tidak ketahui, sehingga pengguna kendaraan yang mengalami kebocoran hanya bertanya pada seseorang di daerah sekitar ataupun hanya memprediksikan letak dan jaraknya saja. Dengan adanya algoritma *Dijkstra* akan mempermudah untuk pencarian tambal ban yang paling terdekat, cara kerja algoritma *Dijkstra* memakai strategi *greedy*, dimana pada setiap langkah dipilih sisi dengan bobot terkecil yang menghubungkan sebuah simpul yang sudah terpilih dengan simpul lain yang belum terpilih. Contoh penerapan algoritma *Dijkstra* adalah pencarian lintasan terpendek yang menghubungkan antara dua kota atau dua tempat yang berlainan. Dengan berkembangnya teknologi informasi saat ini proses pencarian dapat terbantu dengan mudah dan cepat.

Kata Kunci: Algoritma *Dijkstra*, Android, GPS, Pembelajaran GIS.

1. PENDAHULUAN

Pada sepeda motor fungsi ban memiliki peranan penting karena, karena ban adalah salah satu komponen kendaraan yang berfungsi sebagai pengendali arah kendaraan, sebagai penanggung berat beban kendaraan termasuk penumpangnya, sebagai penerus tenaga dari mesin, sebagai sistem peredam.

Kendala kebocoran ban kerap kali menghampiri para pengguna kendaraan bermotor, hal ini perlu menjadi perhatian khusus bagi para pengendara sepeda motor. Kebocoran ban dapat disebabkan oleh beberapa faktor diantaranya faktor eksternal seperti kondisi jalan, ban terkena benda-benda tajam yang terlindas secara tidak sengaja terlindas atau faktor internal yaitu kualitas ban dan kondisi kelayakan ban itu sendiri.

Tidak selalu tersedianya tambal ban di sepanjang jalur yang kerap kali kita lalui sering kali menyusahkan pengguna sepeda motor yang mengalami kebocoran oleh karena itu di perlukan informasi yang dapat memudahkan para pengguna kendaraan mencari informasi secara cepat dan tepat.

Dengan ilmu perkembangan teknologi saat ini telah dikenal adanya *geographical Information System (GIS)* adalah sistem informasi khusus yang mengelola data yang memiliki informasi spasial (*bereferensi keruangan*). informasi spasial adalah sebuah data yang berorientasi geografis, memiliki

sistem koordinat tertentu sebagai dasar referensinya dan mempunyai dua bagian penting yang membuatnya berbeda dari data lain. Atau dalam arti yang lebih sempit adalah sistem komputer yang memiliki kemampuan untuk membangun, menyimpan, mengelola dan menampilkan informasi *bereferensi geografis*.

Dengan lokasi tambal ban yang tidak ketahui, sehingga pengguna kendaraan yang mengalami kebocoran hanya bertanya pada seseorang di daerah sekitar ataupun hanya memprediksikan letak dan jaraknya saja. Dengan berkembangnya teknologi informasi saat ini proses pencarian dapat terbantu dengan mudah dan cepat.

2. ANALISA DAN PERANCANGAN SISTEM

Analisa Masalah

Aplikasi Tambal Ban ini akan berjalan diatas sistem operasi Android dengan menggunakan teknologi GPS sebagai penentu lokasi awal pengguna dan dengan menggunakan metode *client-server*, yang seluruh data koordinat lokasi tujuan, koordinat jalan berada di *server* dan proses perhitungan algoritma *Dijkstra* dilakukan oleh *server*. Hasil dari perhitungan akan dikirimkan ke perangkat pengguna dan ditampilkan berupa penunjuk arah (*direction*) rute terpendek menuju lokasi tujuan.

Aplikasi ini berbasis GIS mobile yang menerapkan peta dari Google (*Google Maps*) sebagai penunjuk lokasi koordinat. Berupa koordinat awal pengguna, koordinat lokasi tujuan maupun koordinat jalan. Seluruh data koordinat disesuaikan dengan koordinat di *Google Maps*. Aplikasi ini ditujukan untuk calon pengguna aplikasi yang ingin mengetahui lokasi tambal ban. Melalui aplikasi ini, calon penumpang dapat memperoleh informasi seperti rute terpendek menuju lokasi tujuan, informasi perkiraan biaya perjalanan, informasi perkiraan waktu perjalanan, informasi lalu-lintas terkini mengenai perkiraan titik yang akan di lewati, sehingga aplikasi mampu mencari rute terpendek tanpa, serta informasi nomor kontak bengkel ban yang ada di Kota Malang.

Identifikasi Masalah

Adanya beberapa permasalahan dalam media pembelajaran tentang belajar Matematika:

- a. menemukan lokasi tambal ban dengan menu Daftar aplikasi TBO dan Peta Lokasi;
- b. memperkirakan jarak antara pengguna dan tambal ban;
- c. aplikasi menyediakan informasi kontak untuk dihubungi;
- d. untuk mempermudah pengendara mobil dan bermotor yang bocor untuk mencari tambal ban terdekat.

Pemecahan Masalah

Dari identifikasi masalah diatas, dapat diberikan suatu pemecahan berupa pembuatan aplikasi game edukasi sebagai media pembelajaran berhitung. Kelebihan dari sistem ini antara lain:

- a. memudahkan pengguna kendaraan menemukan tambal ban;
- b. membantu untuk memperkirakan jarak menuju lokasi tambal ban;
- c. memudahkan pengguna dimana mengalami kendala;
- d. menjadi sarana promosi untuk pelayan tambal ban.

Perancangan Aplikasi

Tahap perancangan merupakan tahap dimana perancangan aplikasi mulai dilakukan. Perancangan tersebut antara lain perancangan struktur menu, perancangan *graphic use interface* aplikasi di perangkat android, perancangan *graphic use interface* untuk admin *database*, dan perancangan *database* aplikasi. Perancangan *interface* aplikasi di android ditujukan kepada pengguna aplikasi, sedangkan *interface* admin *database* ditujukan untuk admin *database* yang akan melakukan update kondisi lalu-lintas, serta update koordinat lokasi tujuan dan koordinat jalan.

Konsep Aplikasi

Perancangan untuk pengguna sistem pada perangkat android antara lain, antarmuka *Home*, antarmuka Navigasi, antarmuka Peta, antarmuka Navigasi Peta, antarmuka Pencarian, antarmuka Kategori Tujuan, antarmuka Nomor Kontak Bengkel Tambal, dan antarmuka Tentang Apps.

Tabel 1. Konsep Aplikasi

No	Kategori Pengguna	Hak Akses	Keterangan
1	Pengguna Aplikasi (<i>Use</i>)	a. Melakukan navigasi ke tempat tujuan yang disertai dengan informasi perkiraan biaya perjalanan yang diawali posisi pengguna saat menggunakan aplikasi b. Melakukan pencarian lokasi pengguna aplikasi dan tempat bengkel tambal ban c. Melihat lokasi-lokasi tujuan yang tertera di peta	Hak akses penuh
2	Admin	a. Melakukan <i>update database</i> , baik <i>database</i> koordinat jalan maupun koordinat lokasi dan info lalu lintas terkini.	Hak akses penuh

Perancangan

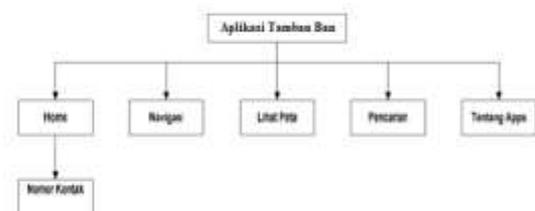
Tahap perancangan merupakan tahap dimana perancangan aplikasi mulai dilakukan. Perancangan tersebut antara lain perancangan struktur menu, perancangan *graphic use interface* aplikasi di perangkat android, perancangan *graphic use interface* untuk admin *database*, dan perancangan *database* aplikasi. Perancangan *interface* aplikasi di android ditujukan kepada pengguna aplikasi, sedangkan *interface* admin *database* ditujukan untuk admin *database* yang akan melakukan update kondisi lalu-lintas, serta update koordinat lokasi tujuan dan koordinat jalan.

Desain Sistem Aplikasi

Tahapan pada fase ini terbagi menjadi 2 yaitu perancangan aplikasi dan perancangan *database* aplikasi.

Perancangan Menu Sistem

Rancangan struktur menu merupakan tahapan untuk merancang bagaimana struktur menu yang akan dibangun. berikut struktur menu dari *system* yang akan dibangun yang dapat dilihat pada gambar di bawah ini.

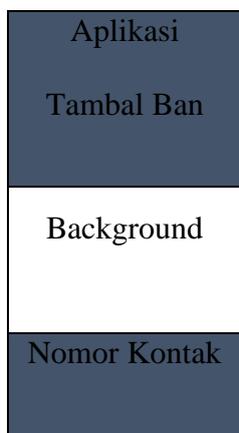


Gambar 1. Rancangan Struktur Menu

Antarmuka Aplikasi

Rancangan antarmuka pengguna aplikasi berfungsi sebagai landasan awal dalam merancang tampilan antarmuka sistem. Secara umum, background aplikasi yang akan digunakan nanti adalah berwarna biru dimana biru sangat identik dengan Kota Malang.

Pada analisa dan perancangan ini, antarmuka untuk pengguna sistem pada perangkat android antara lain, antarmuka *Home*, antarmuka Navigasi, antarmuka Peta, antarmuka Navigasi Peta, antarmuka Pencarian, antarmuka Kategori Tujuan, antarmuka Nomor Kontak Bengkel Tambal, dan antarmuka Tentang *Apps*.



Gambar 2. Antarmuka Halaman *Home*

3. IMPLEMENTASI DAN PEMBAHASAN

Implementasi

Tahapan ini merupakan tahap dimana aplikasi akan dibuat setelah melakukan analisa dan perancangan pada bab sebelumnya. Implementasi ini mencakup pembuatan aplikasi, pembuatan *use interface* untuk admin *database*, pembuatan *database* aplikasi, serta penerapan dari algoritma *Dijkstra*.

Spesifikasi Perangkat Keras

Berdasarkan rancangan yang ada, maka untuk membuat aplikasi Tambal Ban *Online* dibutuhkan beberapa perangkat keras. beberapa perangkat keras pendukung tersebut antara lain adalah:

- a. Telepon Genggam (*Handphone*)
Perangkat ini digunakan untuk menjalankan aplikasi ini dari sisi pengguna. Perangkat yang digunakan yaitu Redmi Note 4x dengan spesifikasi:
 - Sistem Operasi : MIUI versi 8.5.8.0 (NCFMIED)
 - CPU : Octa-core Max 2,0 GHz
 - Memori *internal* : 16 GB
 - Memori RAM : 3 GB

- Dimensi Layar : 1080 x 1920 px
- Konektivitas : GSM / HSPDA / LTE
- b. Unit komputer sebagai *server database* dengan spesifikasi sebagai berikut:
 - Sistem Operasi : Windows 10 Pro 64-bit
 - *Processor* : Intel Core i7-5500U, 2.40Ghz
 - *Memory* : 4,00 GB
 - *Hardisk* : 1 TB

Spesifikasi Perangkat Lunak

Dalam membangun aplikasi Tambal Ban *Online* ini dibutuhkan beberapa perangkat lunak sebagai berikut:

- a. Sistem Operasi
Sistem operasi yang digunakan untuk pemrograman adalah *Windows 10 Pro 64-bit*. Sedangkan sistem operasi untuk *server database* adalah *Linux CentOS 5.5 Final 32 bit*.
- b. Bahasa Pemrograman
Dalam hal ini yang digunakan adalah bahasa pemrograman *Java* dan *PHP*.
- c. Pemodelan UML
Argo UML, Microsoft Visio, Rational Rose

Langkah Pembuatan Program

Pada tahapan ini, akan dijelaskan mengenai proses pembuatan aplikasi, deskripsi pengkodean, dan jalannya sistem yang nanti akan dibuat.

Pembuatan Aplikasi

Tahapan ini merupakan tahapan dimana aplikasi yang telah dirancang, dianalisa, akan dibangun, lalu diuji kelayakannya untuk selanjutnya dioperasikan sebagaimana mestinya sesuai dengan fungsi dan kelayakannya. Berikut ini akan dijelaskan beberapa hal yang berhubungan dengan pembuatan aplikasi berdasarkan analisis dan perancangan yang telah dilakukan terhadap aplikasi Tambal Ban *Online* pada sistem operasi Android ini.

Lingkungan Pengembangan

Perangkat keras dan perangkat lunak merupakan hal yang selalu dibutuhkan untuk mengimplementasikan rancangan yang telah dibuat sebelumnya. Berikut ini akan dijelaskan mengenai lingkungan implementasi yang meliputi lingkungan pada perangkat keras dan perangkat lunak.

Mengenai lingkungan implementasi pengembang harus sesuai dengan kebutuhan sistem, maka dari itu diadakannya survei untuk menghasilkan aplikasi sesuai dengan kebutuhan.

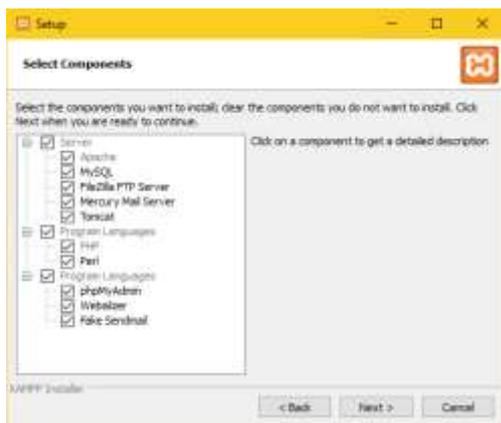
Batasan Dalam Pembuatan Aplikasi

Batasan dalam pembuatan aplikasi ini pada tugas akhir ini adalah:

- Menggunakan bahasa pemrograman Java sebagai bahasa pemrograman untuk mengembangkan aplikasi, yang didukung dengan aplikasi MySQL bahasa pemrograman PHP, serta aplikasi PhpMyAdmin sebagai *interface* untuk mengelola *database* pada *server*.
- Data yang digunakan dalam implementasi ini adalah berupa data simulasi lapangan yang berasal dari google map yang telah didigitasi sesuai dengan kebutuhan aplikasi.
- Aplikasi ini tidak mengembangkan dari sisi *website* sebagaimana aplikasi-aplikasi pada umumnya.
- Tahap pembuatan aplikasi menggunakan *Android Studio*, *Android SDK* version 8 dengan dukungan *Google APIs* 8 dan perangkat *Redmi Note 4x*.

XAMPP

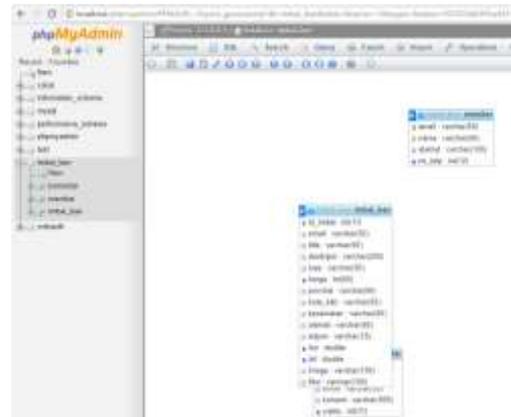
Pada penelitian ini, XAMPP digunakan sebagai *database server* karena XAMPP memiliki keunggulan dalam segi *interface* yang mudah dipelajari dan dioperasikan dalam membuat *database*. XAMPP adalah *server* yang berdiri sendiri dan bersifat lokal (*localhost*), terdiri dari beberapa program diantaranya Apache HTTP *Server*, dan MySQL *database*. Aplikasi ini dapat diunduh atau digunakan secara bebas (*open source*). XAMPP juga mendukung berbagai macam sistem operasi seperti Windows, Linux, MacOS, bahkan Solaris.



Gambar 3. XAMPP

Selanjutnya adalah membuat *database* dan tabel yang diperlukan pada penelitian ini. Untuk membuka halaman *interface* pembuatan *database* dengan mengetikkan pada *web browser* <http://localhost/phpmyadmin>. *Database* yang dibuat terdiri dari 2 buah tabel yakni tabel admin dan tabel

berita, untuk nama kolom dan tipe data bisa dilihat pada Gambar 4.



Gambar 4. *Interface* Pembuatan *Database*

Pembahasan Program

Tahapan implementasi pada aplikasi ini terbagi menjadi beberapa tahapan. Yaitu hasil implementasi perhitungan algoritma yang digunakan, hasil pengujian algoritma yang digunakan pada beberapa kasus tertentu, dan hasil implementasi pada aplikasi.

Hasil Implementasi Aplikasi

Pada tahapan hasil implementasi ini, seluruh *package source* dirubah menjadi sebuah file yang berformat .apk yang diberi nama *Tambal Ban Online*. aplikasi yang telah di install ke perangkat *Redmi Note 4x*.

Secara garis besar, seluruh antarmuka bisa diakses dengan menyentuh layar (*tap*) pada perangkat, kecuali untuk membuka menu aplikasi dengan menekan tombol menu pada perangkat (tidak menyentuh bagian layar aplikasi).

Terdapat beberapa antarmuka yang digunakan untuk interaksi antara pengguna. Antarmuka tersebut dijelaskan sebagai berikut:

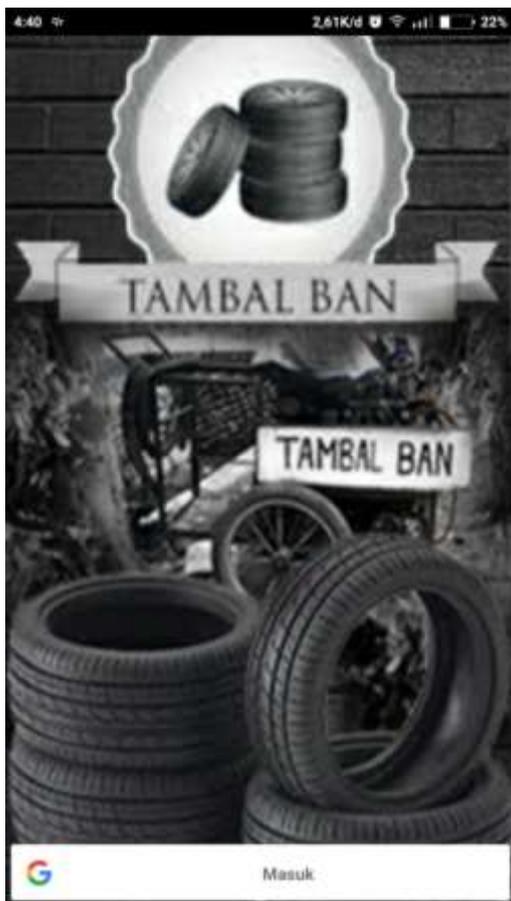
- Tampilan *Opening* Aplikasi.**
Merupakan tampilan awal yang dilihat pengguna ketika membuka aplikasi. Pada tampilan *opening* ini, diberi suara sebagai tanda bahwa aplikasi terbuka yang disertai dengan gambar tambal ban yang muncul. Menu ini juga terdapat tombol *login*.
- Tampilan *Home***
Adalah tampilan utama dari aplikasi TBO. pada tampilan ini, terdapat 1 menu yang bisa di akses dengan cara *tapping*, maka akan muncul 4 menu, yaitu Lihat Peta, Lihat Tambal Ban, Profil dan keluar (*log out*).
- Menu Lihat Peta**
Menu ini menampilkan seluruh lokasi tujuan yang terdapat di *database* yang ditandai

dengan gambar tertentu berdasarkan kategorinya.

- d. Menu Pilih Tambal Ban
Menu ini berisi daftar bengkel tambal ban yang ada di kota Malang. Pada menu ini, pengguna bisa melakukan ke seluruh lokasi tujuan dengan mengetik lokasi yang diinginkan pada *field* yang telah disediakan. Ketika lokasi tujuan didapat, pengguna bisa langsung diarahkan menuju lokasi tujuan.
- e. Menu Profil
Menu ini untuk identitas pengguna misalnya seperti nama *use*, *email* dan bisa mengedit atau menambah pelayan bengkel tambal ban.
- f. Menu *Logout*
Menu ini untuk keluar dari aplikasi tambal ban *Online*.

Halaman Login

Tampilan halaman *login* pada aplikasi TBO dapat dilihat pada Gambar 5.



Gambar 5. Menu Awal *Login*

Kode program untuk autentikasi pada halaman *login* admin adalah sebagai berikut:

```

GoogleSignInOptions signInOptions =
new
GoogleSignInOptions.Builder(GoogleSig
nInOptions.DEFAULT_SIGN_IN).requestEm
ail().build();
googleApiClient = new
GoogleApiClient.Builder(this).enableA
utoManage(this,this).addApi(Auth.GOO
GLE_SIGN_IN_API,signInOptions).build()
;

private void signIn(){
Intent intent =
Auth.GoogleSignInApi.getSignInIntent(
googleApiClient);
startActivityForResult(intent,REQ_COD
E);
}
    
```

Segmen Program 1. Kode Program Autentikasi *Login*

Halaman Home

Pada gambar 6 adalah menu awal saat menjalankan aplikasi, yaitu halaman *login*. Pada aplikasi tambal ban ini untuk *login* menggunakan gmail yang sudah ada di HP. Menu ini otomatis *login* gmail yang ada di HP untuk memudahkan dan mempercepat pengguna di saat sedang mengalami kendala pada kendaraannya.



Gambar 6. Tampilan Menu *Home*

```
private void filter(){
    Intent intent = new
    Intent(awal.this, HasilFilter.class);
    intent.putExtra("em",
    judulTxt.getText().toString());
    startActivity(intent);
}
private void signOut(){
    Auth.GoogleSignInApi.signOut(googleAp
    iClient).setResultCallback(new
    ResultCallback<Status>() {
        @Override
        public void onResult(@NonNull
        Status status) {
            updateUI(false);
        }
    });
}
private void direction(){
    Intent i = new Intent(awal.this,
    direction.class);
    startActivity(i);
}
private void profil(){
    Intent isi=new Intent(awal.this,
    com.example.henry.tmbalban.profil.cla
    ss);
    isi.putExtra("judul",
    kodeTxt.getText().toString()); //mengi
    rim nilai variabel
    isi.putExtra("id",
    judulTxt.getText().toString());
    startActivity(isi);
}
```

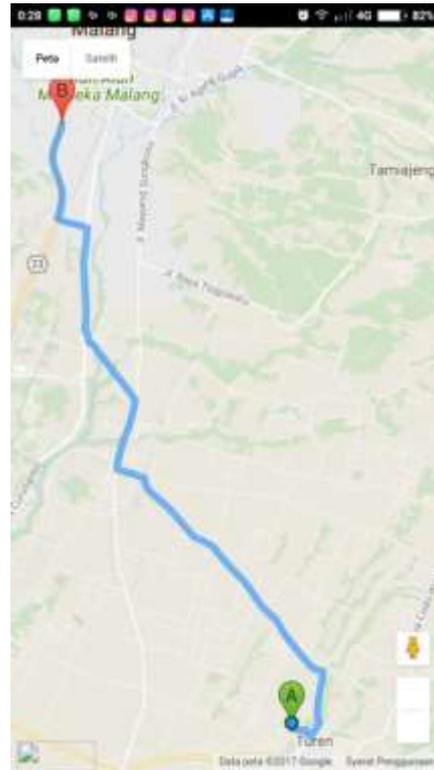
Segmen Program 2. Kode program Halaman Menu

Pada segmen program menu *Home* diatas, pada aplikasi menyediakan empat menu yaitu dari kiri atas terdapat menu lihat peta, pada menu lihat peta ini pengguna bisa melihat dimana posisi kita berada atau sedang mengalami masalah pada ban kendaraan kita. Menu ini kita bisa melihat posisi semua tambal ban yang ada di kota malang, dari sini terlihat lah lokasi tambal ban dan suda ada *direction* yang menunjukkan posisi tambal ban yang paling terdekat.

Menu yang kedua pojok atas kanan, ketika anda menekan tombol dan masuk pada menu yang kedua akan masuk pada menu pilih tambal, pada menu ini terdapat adalah rincian lokasi, deskripsi, nomor telepon, dan mengenai data profil pelayan tambal ban tersebut.

Untuk menu profil bawah kiri itu adalah menu dimana data pribadi untuk pengguna atau *use*, pada menu ini kalian bisa mengedit mengenai identitas misalnya seperti nama dan *email* bisa kalian cantumkan di menu ini.

Halaman Lihat Peta



Gambar 7. Halaman Lihat Peta

```
public class direction extends
    AppCompatActivity {
        WebView browser;
        double latPengguna=0;
        double lonPengguna=0;
        @Override
        protected void onCreate(Bundle
        savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.layout_directi
            on);
            browser=(WebView) findViewById(R.id.bro
            wser);
            browser.setWebChromeClient(new
            WebChromeClient());
            browser.setWebViewClient(new
            WebViewClient());
            browser.clearCache(true);
            browser.clearHistory();
            browser.getSettings().setJavaScriptEna
            bled(true);
            browser.getSettings().setJavaScriptCan
            OpenWindowsAutomatically(true);
            browser.addJavascriptInterface(new
            Object(){
                @JavascriptInterface
                public void ambilNilai(String
                inputan){
                    Toast.makeText(getApplicationContext()
                    , inputan, Toast.LENGTH_SHORT).show();
                }
                @JavascriptInterface
                public String ambilNilai2(String
```

```

inputan1,      String      inputan2){
    double      total=
Double.parseDouble(inputan1)+
Double.parseDouble(inputan2);
    return Double.toString(total);
}
public      void      posisiGPS(){
    try{
        GPSTracker      gps=      new
GPSTracker(this);

        if(gps.canGetLocation()){
            latPegguna=gps.getLatitude();
            lonPegguna=gps.getLongitude();
            Toast.makeText(this,
Double.toString(gps.getLatitude())+
+
Double.toString(gps.getLongitude()),
Toast.LENGTH_SHORT).show();
        }else{
            Toast.makeText(this,"Posisi Anda
tidak      ditemukan      !",
Toast.LENGTH_SHORT).show();
        }

    }catch(Exception      ex){
        Toast.makeText(this,"GPS      Anda
tidak      menyala      !",
Toast.LENGTH_SHORT).show();
    }
}
    
```

Segmen Program 3. Kode Program Lihat Peta

Pada segmen program 3 terdapat atribut *direction* untuk membantu menunjukan arah pada peta dimana titik lokasi awal kita berada kearah tujuan yang kita tuju. Sebelum menggunakan pustaka *Places di Google Maps JavaScript API*, terlebih dahulu pastikan *Google Places API Web Service* telah diaktifkan di *Google API Console*, dalam proyek yang sama dengan yang Anda persiapkan untuk *Google Maps JavaScript API*. Pada tampilan di gambar 8 ini menggunakan *webview* untuk menampilkan pada aplikasi TBO. Fungsi *Webview* pada Android – Pada setiap Android Mobile pasti memiliki *Android System Webview* atau Tampilan Web Sistem Android.

Halaman Menu Pilih Tambal Ban



Gambar 8. Halaman Menu Pilih Tambal Ban

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.
com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp" >
    <android.support.v7.widget.CardView

    android:layout_width="match_parent"

    android:layout_height="wrap_content"
        android:id="@+id/cv" >
        <RelativeLayout

    android:layout_width="match_parent"

    android:layout_height="wrap_content"
        android:padding="16dp" >
        <ImageView

    android:layout_width="wrap_content"

    android:layout_height="wrap_content"
        android:id="@+id/person photo"
    
```

```

android:layout_alignParentLeft="true"
android:layout_alignParentTop="true"
android:layout_marginRight="16dp" />
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
    android:id="@+id/person_name"
android:layout_toRightOf="@+id/person_
photo"
android:layout_alignParentTop="true"
    android:textSize="30sp"/>
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
    android:id="@+id/person_age"
android:layout_toRightOf="@+id/person_
photo"
android:layout_below="@+id/person_name
"/>
</RelativeLayout>
</android.support.v7.widget.CardView>
</LinearLayout>
    
```

Segmen Program 4. Kode Program Pilih Tambal Ban

Halaman Pemanggilan Tambal Ban



Gambar 9. Halaman Pemanggilan Tambal Ban

```

Bundle            isi;
String em,id, judul, image, deskripsi,
luas, harga, alamat, telpon, lat, lon,
garasi,baca,eml,prov,kec,kab;
CollapsingToolBarLayout
collapsingToolBarLayout;
ImageView          imagePreview;
ImageButton        img;
TextView           txtDeskripsi,    txtLuas,
txtTelpon,        txtHarga,        txtAlamat;
private           BottomSheetBehavior
mBottomSheetBehavior;
Toolbar            toolbar;
TextView           km;
private   SliderLayout  imageSlider;
HashMap<String, String> file_maps;
Button            btnMap,koment;
String            body;
WebView           browser;
double            latit;
double            longit;
ImageButton       imgbn;
ImageButton       aa;
LinearLayout fGarasi, fKolam;
    
```

Segmen Program 5. Kode Program Pemanggilan Tambal Ban

Pada segmen program diatas menggunakan *RecyclerView*, *RecyclerView* juga mempunyai animasi default sesuai standar *Google Material Design* saat menambahkan atau menghapus elemen. Sedangkan untuk mengatur posisi *item* pada *list*, *RecyclerView* menggunakan *Layout Managers*, dan ada beberapa jenis *layout managers* yang bisa digunakan.

Halaman Komentar Pengguna



Gambar 10. Halaman Komentar Pengguna

```

void requestData() {
    refreshLayout.setRefreshing(true);
    StringRequest request=new
StringRequest(Request.Method.POST,
url, new Response.Listener<String>() {
    @Override
    public void onResponse(String
response)
    {
refreshLayout.setRefreshing(false);
    List<DataRumah> listData=new
ArrayList<>();
    try
    {
        JSONObject jsonObject=new
JSONObject(response);
        JSONArray
JSONArray=jsonObject.getJSONArray("tmb
al_ban");
        if(jsonObject != null &&
JSONArray.length()>0) {
txtKosong.setVisibility(View.GONE);
        for (int i=0;
i<JSONArray.length(); i++){
refreshLayout.setVisibility(View.VISIB
LE);
txtKosong.setVisibility(View.GONE);
        JSONObject
data=jsonArray.getJSONObject(i);
        DataRumah rumah=new
DataRumah();
        rumah.id=data.getString("id_tmbal");
        rumah.judul=data.getString("email");
        rumah.deskripsi=data.getString("koment
");
        rumah.alamat=data.getString("waktu");
        listData.add(rumah);
        }
        adapterKomentar=new
AdapterKomentar(Komentar.this,
listData);
recyclerView.setAdapter(adapterKomenta
r);
recyclerView.setLayoutManager(new
LinearLayoutManager(Komentar.this));
        }else
        {
txtKosong.setVisibility(View.VISIBLE);
        }
        }catch (JSONException e){
        }
    }
}, new Response.ErrorListener()

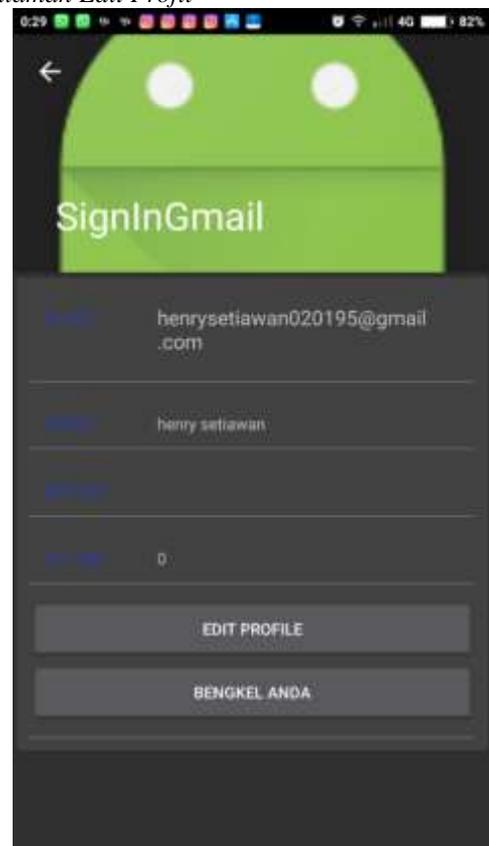
```

Segmen Program 6. Kode Program Komentar

Pada segmen program 6 ini adalah halaman menu komentar untuk pengguna aplikasi TBO, pengguna bisa menilai hasil kinerja bagi pelayan atau servis tambal ban ini. Di buatnya menu ini bermanfaat untuk mengoreksi bagi pelayan maupun pengguna.

Komentar di bahasa pemrograman merupakan suatu keterangan yang digunakan oleh programmer, untuk baris *Syntax* (aturan bahasa) tertentu. Tag Komentar ini sangat membantu sekali buat para kalian programmer, yang ingin menggunakan atau mengedit kembali suatu barisan *code* program. Tag Komentar ini umumnya tidak akan tampil di bagian *output*.

Halaman Edit Profil



Gambar 11. Halaman Edit Profil

```

@Override
protected void onCreate(Bundle
savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.profil);

    initComponents();
    new DownloadJSON1().execute();

    setSupportActionBar(toolbar);

    getWindow().getDecorView().setSystemUi
Visibility(

```

```
View.SYSTEM_UI_FLAG_LAYOUT_STABLE
|
View.SYSTEM_UI_FLAG_LAYOUT_FULLSCREEN
;

bengkel=(Button) findViewById(R.id.deta
il_rumah_btnMap);

prof=(Button) findViewById(R.id.simpan)
;
prof.setOnClickListener(new
View.OnClickListener() {
@Override
public void onClick(View v) {

textView=(TextView) findViewById(R.id.t
extView3);
a=textView.getText().toString();

textView1=(TextView) findViewById(R.id.
textView2);

b=textView1.getText().toString();

textView2=(TextView) findViewById(R.id.
textView);

c=textView2.getText().toString();
Intent isi=new
Intent(profil.this, profils.class);
isi.putExtra("email", email);
isi.putExtra("nama", a);
isi.putExtra("alamat", b);
isi.putExtra("telp", c);
finish();
startActivity(isi);
}
});
bengkel.setOnClickListener(new
View.OnClickListener() {
@Override
public void onClick(View v) {
Intent isi=new
Intent(profil.this,
profillFilter.class);
isi.putExtra("email", email);
startActivity(isi);
}
});

imageSlider.setPresetTransformer(Slide
rLayout.Transformer.Default);
imageSlider.setCustomAnimation(new
DescriptionAnimation());
int durationslider=5000;

imageSlider.setDuration(durationslider
);

collapsingToolBarLayout=(CollapsingToo
lbarLayout) findViewById(R.id.detail_ru
mah_collapse_toolbar);
```

```
getSupportActionBar().setDisplayHomeAs
UpEnabled(true);

imagePreview=(ImageView) findViewById(R
.id.detail_rumah_img_header)}
```

Segmen Program 7. Kode Program Edit Profil

Halaman Tambah Tambal Ban



Gambar 12. Tambah Tambal Ban

```
void requestData(){
refreshLayout.setRefreshing(true);
StringRequest request=new
StringRequest(Request.Method.POST,
url, new Response.Listener<String>() {
@Override
public void onResponse(String
response) {

refreshLayout.setRefreshing(false);
List<DataRumah> listData=new
ArrayList<>();
try {
JSONObject jsonObject=new
JSONObject(response);
JSONArray jsonArray=
jsonObject.getJSONArray("tmb
al_ban");
if(jsonArray != null &&
jsonArray.length()>0){

txtKosong.setVisibility(View.GONE);
for (int i=0;
```

```

i<jsonArray.length();          i++){
refreshLayout.setVisibility(View.VISIB
LE);

txtKosong.setVisibility(View.GONE);
JSONObject
data=jsonArray.getJSONObject(i);
DataRumah          rumah=new
DataRumah();

rumah.id=data.getString("id_tmbal");

rumah.eml=data.getString("email");

rumah.prov=data.getString("provinsi");

rumah.kec=data.getString("kecamatan");

rumah.judul=data.getString("title");

rumah.deskripsi=data.getString("deskri
psi");

rumah.luas=data.getString("luas");

rumah.kab_kota=data.getString("kota_ka
b");

rumah.alamat=data.getString("alamat");

rumah.telpon=data.getString("telpon");

rumah.harga=data.getString("harga");

rumah.garasi=data.getString("fitur");

rumah.lat=data.getString("lat");

rumah.lon=data.getString("lon");

rumah.image=data.getString("image");
listData.add(rumah);
}
adapterRumah=new
AdapterRumah1(profillFilter.this,
listData);

recyclerView.setAdapter(adapterRumah);

recyclerView.setLayoutManager(new
LinearLayoutManager(profillFilter.this
));
}else
{txtKosong.setVisibility(View.VISIBLE)
;
} }catch (JSONException e){

}
}, new Response.ErrorListener() {
@Override
public void
onErrorResponse(VolleyError error) {

```

```

refreshLayout.setRefreshing(false);
dialog(error.toString());
}
}){
@Override
protected Map<String, String>
getParams() throws AuthFailureError {
Map<String, String> map=new
HashMap<>();
map.put("email", email);
return map;
}
requestQueue.add(request);}

```

Segmen Program 8. Kode Program Tambah Tambal Ban

Hasil Implementasi Perhitungan Algoritma Dijkstra

Algoritma *Dijkstra* diterapkan pada graph berarah dan berbobot yang seluruh data dari graph nya telah direpresentasikan ke dalam bentuk tabel. Untuk itu diperlukan data tabel hasil dari representasi titik-titik koordinat jalan yang ada di kota Malang.

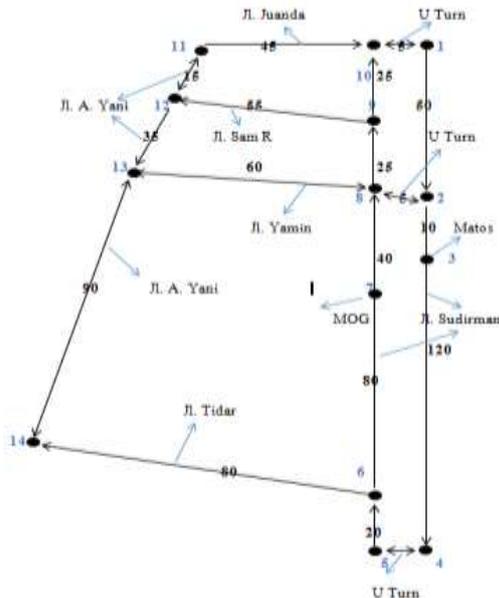
Cara Kerja Algoritma Dijkstra

Algoritma ini mencari panjang lintasan terpendek dari *node* asal ke *node* tujuan dalam sebuah *graph*. Langkah-langkah dalam menentukan lintasan terpendek pada algoritma *Dijkstra* yaitu:

- Pada awalnya inialisasikan *node* asal (V1) dan *node* tujuan (V2).
- Buat 2 buah *list*, *open list* dan *closed list*. Keduanya tidak ada data atau kosong, dan formatnya {*node* cabang - total bobot - *node* induk}.
- Masukkan V1 ke *open list*.
- Pilih 1 *node* dengan bobot terkecil pada *open list*, kemudian pindahkan ke dalam *closed list*. Sehingga bobot terkecil tadi tidak terdapat lagi di *open list*.
- Cari *node* yang bertetangga langsung dari *node* sebelumnya, yang masuk terakhir ke dalam *closed list*. Tambahkan bobot dengan *node* yang terkait, apabila sudah ada dalam *closed list* abaikan.
- Apabila dalam *open list* terdapat *node* yang sudah dibandingkan, lalu cari yang terkecil, dan perbaharui. Bila ternyata jumlah bobotnya sama dalam *node* yang sama, maka abaikan.
- Apakah data dalam *open list* kosong? jika belum ulangi langkah 4.
- Pencarian berhenti apabila *node* tujuan telah terdapat di *closed list*.
- Dalam *closed list* cari V2, telusuri jalur berdasarkan *node* induk sampai mengacu ke *node* asal (V1), dan balikkan urutan *node*. Pilih total bobot yg terkecil apabila ada *node* cabang yang lebih dari 1.

- j. Lintasan terpendek ditemukan bersama bobotnya.
- k. Algoritma Dijkstra digunakan untuk pencarian jalur terpendek dari suatu graph, sehingga akan didapatkan rute tempuh yang terpendek.

Pada Gambar 13 di bawah ini adalah contoh dari jalan yang ada di Kota Malang yang telah dirubah ke dalam bentuk graph berarah dan berbobot sesuai dengan kondisi arah lalu lintas jalan tersebut.



Gambar 13. Contoh Graph Jalan Kota Malang

Tabel 2 dibawah ini adalah hasil representasi dari graph jalan pada gambar 13 di atas. Tabel inilah yang nantinya yang akan digunakan oleh algoritma Dijkstra untuk mencari rute terpendek dari suatu node ke node yang lainnya.

Tabel 2. Graph Dijkstra

No	Nama Jalan	Titik Awal	Titik Akhir	Panjang	Kategori
1	Jl. Sudirman	1	2	50	Jalan
2	Matos	2	3	10	Mall
3	Jl. Sudirman	3	4	120	Jalan
4	U Turn	4	5	5	U Turn
5	U Turn	5	4	5	U Turn
6	Jl. Sudirman	5	6	20	Jalan
7	MOG	6	7	80	Mall
8	Jl. Tidar	6	14	80	Jalan
9	Jl. Sudirman	7	8	40	Jalan
10	Jl. Sudirman	8	9	25	Jalan
11	Jl. Sudirman	9	10	25	Jalan
12	U Turn	10	1	5	U Turn
13	U Turn	1	10	5	U Turn
14	Jl. A. Yani	14	13	90	Jalan
15	Jl. A. Yani	13	14	90	Jalan
16	Jl. A. Yani	13	12	35	Jalan
17	Jl. A. Yani	12	13	35	Jalan
18	Jl. A. Yani	11	12	15	Jalan
19	Jl. A. Yani	12	11	15	Jalan
20	Jl. Juanda	11	10	45	Jalan

No	Nama Jalan	Titik Awal	Titik Akhir	Panjang	Kategori
21	Jl. Sam Ratulangi	9	12	55	Jalan
22	Jl. M. Yamin	8	13	60	Jalan
23	Jl. M. Yamin	13	8	60	Jalan
24	U Turn	8	2	5	U Turn
25	U Turn	1	8	5	U Turn

Contoh Perhitungan Pencarian Algoritma Dijkstra

Berpedoman pada Gambar 13 diatas serta pada Tabel 2, maka akan dilakukan proses pencarian yang berasal dari node 14 menuju node 7. Berikut tahapan dalam menghitung rute terpendek menggunakan algoritma Dijkstra.

- a. Inisialisasi

Tabel 3. Inisialisasi

OPEN LIST	CLOSED LIST
(cabang-total bobot-induk)	(induk-total bobot-cabang)
14-0-null	----

Ini merupakan tabel perhitungan awal yaitu Inisialisasi titik awal ke dalam open list. Closed list kosong. Di sertai dua atribut seperti open list dan closed list untuk perhitungan pencarian algoritma Dijkstra.

- b. Iterasi 1

Tabel 3. Iterasi 1

OPEN LIST	CLOSED LIST
(cabang-total bobot-induk)	(induk-total bobot-cabang)
13-90-14	14-0-null

Iterasi 1, memilih total bobot terkecil pada open list dari iterasi sebelumnya, kemudian pindahkan ke closed list. dalam hal ini, 14-0-null adalah bobot terkecil dari open list pada iterasi sebelumnya, dan dimasukkan ke closed list. Kemudian, buka open list yang node induknya adalah node 14 berdasarkan graph atau tabel.

- c. Iterasi 2

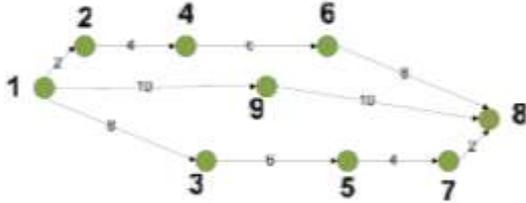
Tabel 4. Iterasi 2

OPEN LIST	CLOSED LIST
(cabang-total bobot-induk)	(induk-total bobot-cabang)
14-180-14	14-0-null
8-150-13	13-90-14
12-125-13	

Iterasi 2, memilih total bobot terkecil pada open list dari iterasi sebelumnya, kemudian pindahkan ke closed list. dalam hal ini, 13-90-14 adalah bobot terkecil dari open list pada iterasi sebelumnya, dan dimasukkan ke closed list. Kemudian, buka open list yang node induknya adalah node 13 berdasarkan graph atau tabel. Untuk iterasi selanjutnya dapat dilihat pada Lampiran C.

Hasil Pengujian Algoritma Dijkstra Pada Beberapa Bentuk Graph

1. Graph Tipe A



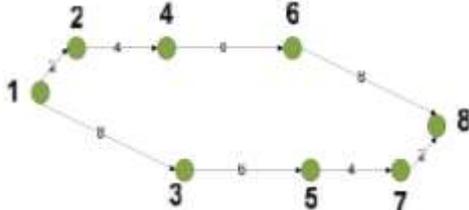
Gambar 14. Graph Tipe A

Tabel 5. Hasil Representasi Graph Tipe A

No	Node Asal	Node Tujuan	Panjang
1	1	2	2
2	1	3	8
3	2	4	4
4	3	5	6
5	4	6	6
6	5	7	4
7	6	8	8
8	7	8	2
9	1	9	10
10	9	8	10

Pada kondisi graph tipe A, setelah dilakukan pencarian menggunakan algoritma Dijkstra dari node asal 1 menuju node tujuan 8, maka didapatkan hasil rute terpendeknya adalah **1→9→10** dengan total bobot adalah **20**.

2. Graph Tipe B



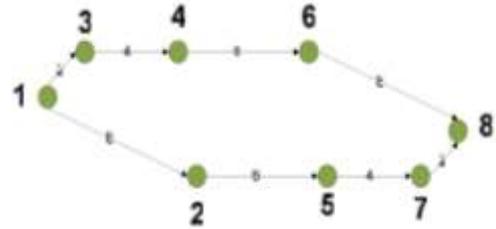
Gambar 15. Graph Tipe B

Tabel 6. Hasil Representasi Graph Tipe B

No	Node Asal	Node Tujuan	Panjang
1	1	2	2
2	1	3	8
3	2	4	4
4	3	5	6
5	4	6	6
6	5	7	4
7	6	8	8
8	7	8	2

Pada kondisi graph tipe B, setelah dilakukan pencarian menggunakan algoritma Dijkstra dari node asal 1 menuju node tujuan 8, maka didapatkan hasil rute terpendeknya adalah **1→2→4→6→8** dengan total bobot adalah **20**.

3. Graph Tipe C



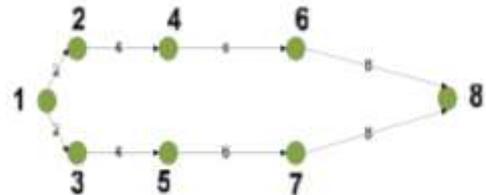
Gambar 16. Graph Tipe C

Tabel 7. Hasil Representasi Graph Tipe C

No	Node Asal	Node Tujuan	Panjang
1	1	3	2
2	1	2	8
3	3	4	4
4	2	5	6
5	4	6	6
6	5	7	4
7	6	8	8
8	7	8	2

Pada kondisi graph tipe C, setelah dilakukan pencarian menggunakan algoritma Dijkstra dari node asal 1 menuju node tujuan 8, maka didapatkan hasil rute terpendeknya adalah **1→3→4→6→8** dengan total bobot adalah **20**.

4. Graph Tipe D



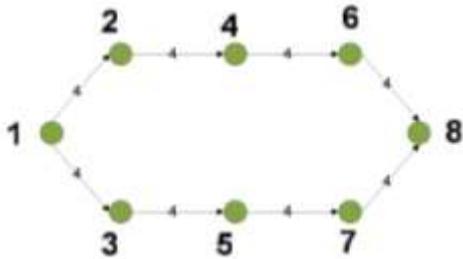
Gambar 17. Graph Tipe D

Tabel 8. Hasil Representasi Graph Tipe D

No	Node Asal	Node Tujuan	Panjang
1	1	2	2
2	1	3	2
3	2	4	4
4	3	5	4
5	4	6	6
6	5	7	6
7	6	8	8
8	7	8	8

Pada kondisi graph tipe D, setelah dilakukan pencarian menggunakan algoritma Dijkstra dari node asal 1 menuju node tujuan 8, maka didapatkan hasil rute terpendeknya adalah **1→2→4→6→8** dengan total bobot adalah **20**.

5. *Graph* tipe E



Gambar 18. *Graph* Tipe E

Tabel 9. Hasil Representasi *Graph* Tipe E

No	Node Asal	Node Tujuan	Panjang
1	1	2	4
2	1	3	4
3	2	4	4
4	3	5	4
5	4	6	4
6	5	7	4
7	6	8	4
8	7	8	4

Pada kondisi *graph* tipe E, setelah dilakukan pencarian menggunakan algoritma *Dijkstra* dari *node* asal 1 menuju *node* tujuan 8, maka didapatkan hasil rute terpendeknya adalah **1→2→4→6→8** dengan total bobot adalah **20**.

Implementasi Fitur Pemberitahuan (Alert) Aplikasi

Fitur *alert* berfungsi sebagai pemberitahuan kepada pengguna aplikasi apabila koordinat pengguna tidak berada pada rute perjalanan yang telah ditentukan sebelumnya.

a. Komponen Fitur *Alert*

Dalam penerapannya, fitur ini membutuhkan beberapa komponen seperti perangkat GPS, koneksi GPRS, dan *server*.

• GPS

Dalam menerapkan fitur ini, *gps* sangat berperan dalam menangkap koordinat pengguna. Koordinat pengguna dikirim ke *server* untuk terus dipantau. Untuk itu dibutuhkan sebuah perangkat GPS yang mampu mengirimkan serta menerima sinyal dari satelit dengan cepat. Dan mampu menangkap koordinat pengguna ketika pengguna sedang bergerak.

• Koneksi GPRS

Koordinat yang didapatkan oleh GPS, akan dikirimkan ke *server* untuk diolah melalui perangkat *mobile* pengguna. Untuk itu dibutuhkan koneksi GPRS yang memiliki respon dan koneksi yang cepat dalam pengiriman dan penerimaan paket data.

• *Server*

Selain untuk mendapatkan rute terpendek, *server* juga mengolah data koordinat pengguna untuk dilakukan penyesuaian

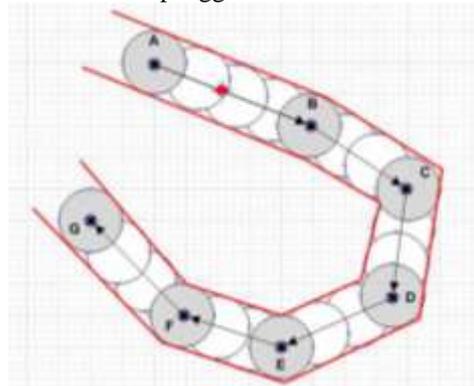
dengan data rute terpendek yang akan ditempuh. Apabila hasil pengolahan menunjukkan bahwa koordinat pengguna tidak berada pada jalur yang telah ditetapkan, maka *server* akan mengirimkan informasi kepada perangkat pengguna dan kemudian *alert* akan aktif.

b. Proses Kerja *Alert*

Dalam menentukan apakah koordinat pengguna yang diterima *server* berada pada jalur yang telah ditentukan sebelumnya atau tidak, maka dibutuhkan sebuah perhitungan yang mampu memberikan keputusan apakah pengguna berada dijalan atau tidak.

Perhitungan ini yakni, memulai dengan membuat daerah *range/cakupan* koordinat dari setiap titik koordinat jalan yang telah ditandai oleh di peta. Kemudian membandingkan apakah koordinat pengguna masih berada di daerah *range/cakupan* koordinat atau tidak.

Pada gambar 19 di bawah ini menjelaskan detail dari daerah *range/cakupan* koordinat sebagai batasan koordinat pengguna.



Gambar 19. Daerah *Range/Cakupan* Koordinat

Keterangan gambar :

- Titik hitam A, B, C, D, E, F, G adalah titik koordinat jalan yang ditandai dipeta.
- Garis panah berwarna hitam yang menghubungkan antar titik berwarna hitam merupakan arah tujuan yang harus diikuti pengguna. Garis ini merupakan sebuah rute terpendek untuk menuju lokasi tujuan.
- Lingkaran berwarna abu-abu merupakan daerah *range/cakupan* koordinat yang telah dihitung besarnya.
- Lingkaran berwarna putih merupakan daerah *range/cakupan* koordinat semu yang saling menghubungkan antara lingkaran abu-abu.
- Titik berwarna merah merupakan koordinat pengguna.
- Garis berwarna merah merupakan pinggir jalan. Ini juga menandai seberapa lebar jalan yang bisa dilalui.

Setelah rute terpendek didapat melalui perhitungan algoritma *Dijkstra* yang dilakukan *server*, maka akan ditentukan hasil perhitungan daerah *range/cakupan* koordinat sebagai batas koordinat yang tidak boleh dilalui pengguna hingga sampai ketujuan. Daerah *range/cakupan* koordinat ini juga termasuk daerah *range/cakupan* koordinat semu (pada gambar 19 adalah lingkaran yang berwarna putih). Tujuannya agar tidak terjadi putus daerah *range/cakupan* koordinat. Daerah ini adalah batasan yang boleh dilalui oleh pengguna.

Kemudian, aplikasi menangkap koordinat pengguna melalui perangkat GPS, lalu mengirimkan koordinat tersebut ke *server* untuk dilakukan pendeteksian posisi pengguna berdasarkan data daerah *range/cakupan* koordinat yang telah didapat. Apabila koordinat pengguna masih berada di daerah tersebut, maka *server* mengirimkan informasi bahwa koordinat pengguna masih berada di daerah *range/cakupan* koordinat dan *alert* tidak aktif. Sebaliknya, jika koordinat pengguna tidak berada di daerah *range/cakupan* koordinat, maka *server* mengirimkan informasi ke pengguna bahwa koordinat pengguna tidak berada pada daerah *range/cakupan* koordinat, dan *alert* akan aktif sehingga aplikasi memberitahukan informasi kepada pengguna bahwa posisi pengguna sedang tidak berada pada rute yang telah ditentukan sebelumnya.

Proses ini terus berlangsung hingga koordinat pengguna telah berada di daerah *range/cakupan* koordinat tujuan.

Pengujian

Tahapan pengujian merupakan tahapan dimana akan dilakukan pengujian terhadap aplikasi. Pengujian dilakukan dengan metode pengujian *blackbox*. Selain itu, dilakukan juga pengujian terhadap akses ke aplikasi pada lokasi tertentu, jam tertentu, serta menggunakan *provider* internet tertentu.

Testing dan Pengujian Aplikasi

Tahapan ini dilakukan untuk mengetahui apakah sistem yang dibangun telah sesuai dengan yang diharapkan. Tujuan utama dari *testing* dan pengujian sistem adalah untuk memastikan bahwa elemen-elemen atau komponen-komponen dari sistem telah berfungsi sesuai dengan yang diharapkan. Salah satu metode pengujian jenis ini dikenal dengan pengujian *blackbox*.

Pengujian Blackbox Aplikasi Tambal Ban Online

Pada tahap pengujian sistem ini, perangkat keras yang digunakan yaitu *smartphone Redmi Note 4x*. Sedangkan material pengujian untuk sistem ini V-21 menggunakan data koordinat yang

telah dimasukan oleh Admin *Database* ke *database* pada *server*. Pengujian yang akan dilakukan adalah pengujian akses ke sistem dan pengujian fungsionalitas sistem menggunakan metode *blackbox*. Hasil dari pengujian ini dapat dilihat pada tabel 11.

Tabel 10. Hasil Pengujian *Blackbox*

No	Komponen Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Keterangan
1	Halaman Utama Aplikasi (<i>login</i>)	Pengguna mengklik aplikasi di perangkat dan akan muncul icon gmail aplikasi menuju ke <i>Home</i> aplikasi	<i>Opening</i> aplikasi berjalan, setelah otomatis <i>login</i> gmail dan <i>Home</i> aplikasi berhasil ditampilkan	Benar
2	Menu Tujuan Pencarian	Pengguna mengklik tombol menu pada perangkat android dan memilih menu Pencarian kemudian akan menampilkan lokasi peta tambal ban berada	Kategori Pencarian berhasil ditampilkan	Benar
3	Daftar Kategori Tujuan	Pengguna memilih kategori yang diinginkan dan akan ditampilkan daftar lokasi tujuannya	Daftar lokasi tujuan berhasil ditampilkan	Benar
4	Memilih 1 Daftar Lokasi Tujuan	Pengguna memilih salah satu dari daftar lokasi tujuan dan akan muncul tampilan Tujuan	Tampilan Tujuan ditampilkan	Benar
5	Menu Go (pemanggilan)	Pengguna mengklik tombol GO. Maka pemanggilan tambal ban akan di proses	Koordinat didapat dan muncul <i>direction</i> rute terpendek menuju dan waktu yang di tempuh	Benar
6	Menu Lihat Peta	Pengguna menekan tombol menu pada perangkat android dan	Peta lokasi tambal ban berhasil dimunculkan	Benar

No	Komponen Pengujian	Hasil Yang Diharapkan	Hasil Pengujian	Keterangan
7	Menu Pencarian Tambal Ban	memilih menu Lihat Peta. akan ditampilkan peta yangtelah ditandai dengan icon maka akan ditampilkan <i>direction</i> rute terpendek menuju lokasi tujuan. Pencarian. dan akan menampilkan seluruh daftar tambal ban yang akan di pilih lokasi tujuan.	Daftar menu lokasi tambal ban berhasil ditampilkan	Benar
8	Menu Nomor Tambal Ban	Pengguna mengklik tombol Nomor Kontak tambal ban pada aplikasi dan akan menampilkan <i>message</i> untuk mengirim <i>Lan-Lot</i> posisi anda berada.	Daftar nomor Kontak pemanggilan berhasil ditampilkan	Benar
9	Men Log Out	Pengguna menekan tombol menu pada perangkat android dan memilih menu Log Out maka akan keluar dari aplikasi.	Keluar dari aplikasi berhasil	Benar

Pengujian Akses Aplikasi Tambal Ban Online

Pengujian yang dilakukan bertujuan untuk mengetahui proses hasil dari sistem, yaitu memperlihatkan aplikasi Tambal Ban *Online* yang telah dibangun bisa diakses melalui berbagai perangkat Android dan berbagai versi Android. Hasil dari pengujian dapat dilihat halaman *Home* telah tampil, dan semua menu serta fitur dapat digunakan.

Pada tabel 12 di bawah ini menjelaskan pengujian akses ke aplikasi di beberapa lokasi yang berbeda.

Tabel 11. Pengujian Akses Aplikasi Di beberapa Lokasi Berbeda

1	Tanggal Lokasi Awal Lokasi Tujuan Pengujian	15/10/2017 Turen Sukun Koordinat Go Lihat Peta Lihat Tambal	Hasil Berhasil Berhasil Berhasil Berhasil	
	Tampilan Hasil Pengujian			
2	Tanggal Lokasi Awal Lokasi Tujuan Pengujian	30/10/2017 Tidar Mergan Koordinat Go Lihat Peta Lihat Tambal	Hasil Berhasil Berhasil Berhasil Berhasil	
	Tampilan Hasil Pengujian	Hasil 		

Tabel 12. Pengujian Aplikasi Pada Beberapa Perangkat

No	Merk Perangkat	Sistem Operasi	Hasil Pengujian Aplikasi
1	Redmi Note 4x	MIUI v8.5.8.0 (NCFMIED)	Berjalan dengan lancar

2	Redmi 4x	MIUI v8.5.8.0 (NCFMIED)	Berjalan dengan lancar
3	Samsung J7 Prime	v8.1 (Oreo)	Berjalan dengan lancar

Pengujian Fitur Alert

Berdasarkan hasil analisa dari proses kerja fitur ini, maka hingga saat ini fitur *alert* masih belum dapat terealisasi karena disebabkan oleh beberapa faktor yaitu:

- a. Kemampuan perangkat GPS dalam mendapatkan koordinat pengguna ketika sedang bergerak. Setiap perangkat *mobile* memiliki kemampuan yang berbeda-beda dalam penggunaan GPS karena didukung oleh spesifikasi perangkat itu sendiri seperti prosesor, memory, dan tipe GPS yang digunakan.
- b. Perhitungan daerah *range*/cakupan koordinat berdasarkan rute terpendek yang telah ditentukan sebelumnya masih belum terpecahkan. Hal ini terjadi karena penentuan daerah *range* ini berhubungan dengan data koordinat *longitude* dan *latitude* yang langsung merepresentasikan objek di muka bumi yang memiliki tingkat akurasi tinggi. Selain itu, penentuan daerah *range*/cakupan koordinat semu sebagai pembantu agar seluruh daerah *range*/cakupan koordinat dapat saling terhubung masih belum ditemukan solusinya.

4. KESIMPULAN

Setelah dilakukan beberapa pengujian terhadap aplikasi yang telah dibangun, maka dapat ditarik kesimpulan dari hasil pengujian tersebut. Berikut kesimpulannya:

- a. Aplikasi yang dijalankan di beberapa perangkat Android yang berbeda merk, versi Android, dan berbeda operator dapat berjalan dengan lancar.
- b. Lamanya pencarian rute terpendek oleh *server* disebabkan oleh beberapa faktor, diantaranya koneksi internet pada perangkat android, serta posisi awal pengguna menuju lokasi tujuan. Semakin jauh dari tujuan maka akan membutuhkan waktu yang tidak sebentar dalam menentukan rute terpendeknya.
- c. Penggunaan perangkat GPS untuk menentukan koordinat pada perangkat pengguna sangat dominan. Apabila koordinat pengguna berhasil didapat, maka aplikasi bisa berjalan sebagaimana mestinya. Apabila tidak, maka aplikasi tidak dapat membantu pengguna untuk menentukan *direction* rute terpendek.
- d. Rute terpendek yang dilalui dari lokasi awal ke lokasi tujuan belum tentu sama dengan rute terpendek dari lokasi tujuan ke lokasi awal. Hal ini biasa dilihat dari hasil pengujian pada tabel

4.7 point 7 dan 8, ini disebabkan oleh peraturan jalan yang ada di Kota Malang. Sehingga menghasilkan rute yang berbeda dari sebelumnya dengan jarak tempuh dan waktu tempuh yang berbeda.

5. REFERENSI

- [1] Adiwijaya, M. Kodrat, I.S. Christyono, Yuli. 2015. Perancangan Game Edukasi Platform Belajar Matematika Berbasis Android Menggunakan Construct2. *Jurnal TEKNIK UNDIP*. 36 (II): 24-39.
- [2] Amrullah, Ahmad Zuli. 2013. *Analisis dan Perancangan Game Petualangan "Jumper" Berbasis Android*. Yogyakarta: STMIK Amikom.
- [3] Aprilianti, Y. Lestari, Uning. Iswahyudi, Catur. 2013. Aplikasi Mobile Game Edukasi Matematika Berbasis Android Application Of Education Mobile Games for Math Based on Android. *Jurnal SCRIPT*. 1 (I): 89-97.
- [4] Arsyad, Azhar. 2011. *Media Pembelajaran Matematika Asyik*. Jakarta: PT Grafindo Persada.
- [5] Audacity. About Audacity. *Online*: <http://www.audacityteam.org/about/>. Diakses tanggal 15 Oktober 2016.
- [6] Busran dan Fitriyah. 2015. Perancangan Permainan Edukasi Belajar Membaca pada Anak Prasekolah Berbasis Android. *Jurnal TEKNOIF*. 3 (I): 62-69.
- [7] *Corona SDK The Ultimate 2D Development Platform*. *Online*: <http://www.coronalabs.com/products/coronasdk/>. Diakses tanggal 19 Oktober 2016, dari
- [8] Dewobroto, Wiryanto. 2005. *Aplikasi Rekayasa Konstruksi dengan Visual Basic 6.0*. Jakarta: Elex Media Komputindo.
- [9] Eric Haughee. 2013. *Instant Sublime Text Starter*. Birmingham: Packt Publishing.
- [10] Febrianto, Luky. 2014. *Analisis Dan Perancangan Game Android Kopi Luwak*. Yogyakarta: STMIK Amikom.
- [11] Bakri, Hasrul. 2010. Langkah-langkah Pengembangan Pembelajaran Multimedia Interaktif. *Jurnal MEDTEK*. 2 (I): 1-8.

- [12] Hergenhahn. B.R. & Olson M.H. 2008. *Theories of Learning Edisi Ketujuh (Tribowo B.S. Terjemahan)*. New Jersey: Pearson Education.
- [13] Rifa, Iva. 2012. *Koleksi Game Edukatif di Dalam dan Luar Sekolah*. Yogyakarta: FlashBooks.
- [14] Sibero, Ivan C. 2010. *Membuat Game 2D*. Yogyakarta: MediaKom.
- [15] Jailani, F. Sandro, Dedek dan Irawan, Bambang. 2012. Rancang Bangun Aplikasi Edugame Matematika Berbasis Android. *Jurnal TEKNOMATIKA*. 2 (III): 1-10.
- [16] Madcoms. 2005. *Seri Panduan Lenangkap Adobe Photoshop CS*. Yogyakarta: ANDI.
- [17] Martono. K.T. 2015. Pengembangan Game dengan Menggunakan Game Engine Game Make. *Jurnal Sistem Komputer*. 5 (I): 23.
- [18] Fernandez, Michelle M. 2012. *Corona SDK Mobile Game Development Beginner's Guide. 2nd. Ed.* Birmingham: Packt Publishing.
- [19] Hamid, Moh. Sholeh. 2011. *Moetode Edutainment*. Yogyakarta: DIVA Press.
- [20] Safaat H, Nazruddin. 2014. *Android Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Bandung: Informatika Bandung.
- [21] Ngafifi, Muhamad. 2014. Kemajuan Teknologi Dan Pola Hidup Manusia Dalam Perspektif Sosial Budaya. Wonosobo: *Jurnal Pembangunan Pendidikan*.
- [22] Nursyaidah. 2015. *Thariqah Ilmiah Vol.02 02 Juli 2015 Gaya Mengajar Guru Terhadap Motivasi Belajar Siswa*. Padang: Thariqah Ilmiah.
- [23] Sulaeman, Usep. 2011. *Pembuatan Personal Computer Game Edukasi Matematika Untuk Anak TK*. UNIKOM, Bandung: Undergraduate Theses.
- [24] Tinarbuko, Sumbo. 2003. *Semiotika Analisis Tanda Pada Karya Desain komunikasi Visual*. Jakarta: Jurnal Nirmana.