

Pengembangan Aplikasi *Traffic Light E-Tilang* Menggunakan Google Geofencing API Berbasis Android

Ditya Enandini Palupi¹, Muhammad Aminul Akbar², Adam Hendra Brata³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹enandiniditya@yahoo.com, ²muhammad.aminul@ub.ac.id, ³adam@ub.ac.id

Abstrak

Tingkat pelanggaran lalu lintas di Indonesia semakin meningkat. Hal ini terbukti karena meningkatnya angka kecelakaan yang ada di kota Surabaya. Dalam rangka mengurangi penurunan pelanggaran lalu lintas jalan, maka diperlukan aturan atau pembinaan kepada masyarakat untuk pentingnya menaati peraturan lalu lintas yang sudah ditetapkan oleh pemerintah. Dinas Perhubungan Kota Surabaya merupakan salah satu lembaga atau instansi yang menyediakan prasarana dan sarana transportasi untuk mendukung kegiatan pembangunan daerah serta meningkatkan kesejahteraan masyarakat, salah satunya dengan memfasilitasi kamera CCTV. Kamera tersebut di fungsikan untuk menerapkan sistem tilang elektronik yang beroperasi sejak awal September 2017. Dengan adanya bantuan CCTV tersebut, sistem dapat membantu mengidentifikasi jenis pelanggaran, pelaku, hingga lokasi pelanggaran berada. Penulis melakukan survei kepada 30 responden terkait kebutuhan-kebutuhan yang dibutuhkan untuk pengguna kendaraan yang belum mengetahui kapan serta dimana melakukan pelanggaran tersebut. Dengan latar belakang tersebut dibuatlah sebuah aplikasi sistem pemantau *traffic light* untuk mempermudah pengguna kendaraan dalam menemukan informasi pelanggaran dan memberikan pengguna sebuah peringatan yang memasuki daerah *traffic light*. Implementasi dilakukan dengan Android Studio. Pengujian dilakukan dengan melewati suatu daerah dan hasil yang didapatkan yaitu hasil jarak berdasarkan kecepatan kendaraan.

Kata kunci: *traffic light*, pengguna, notifikasi

Abstract

The level of traffic violations in Indonesia is increasing. This is evidenced by the increasing number of accidents in the city of Surabaya. In order to reduce the reduction of road traffic violations, it is necessary to have rules or guidance to the public for the importance of obeying the traffic rules set by the government. Surabaya Transportation Department is one of institutions or agencies that provide infrastructure and transportation facilities to support local development activities and improve the welfare of the community, one of them by facilitating CCTV cameras. The camera is enabled to implement an electronic ticketing system that operates since the beginning of September 2017. With the help of CCTV, the system can help identify the types of violations, offenders, to where the offense is located. The authors conducted a survey to 30 respondents terkait necessary needs for vehicle users who do not know when and where the conduct of the offense. Against this background, a traffic light monitoring system application was created to make it easier for vehicle users to locate infringement information and give users a warning that enters the traffic light area. Implementation is done with Android Studio. Testing is done by passing an area and the result of distance based on vehicle speed.

Keywords: *traffic light*, users, notifications

1. PENDAHULUAN

Tingkat pelanggaran lalu lintas di Indonesia semakin meningkat. Hal ini terbukti karena meningkatnya angka kecelakaan yang ada di kota besar (Mohammad Ridwan, 2017). Tingkat

jumlah kendaraan yang beredar dari tahun ketahun semakin meningkat. Menurut Raydian Kokrosono selaku Kasat Lantas Polresta Surabaya AKBP, hal ini dapat memicu pengaruh keamanan lalu lintas yang semakin sering terjadi dan pelanggaran lalu lintas yang menimbulkan

kecelakaan lalu lintas serta kemacetan lalu lintas (Wahyudin, 2014).

Untuk mengatasi permasalahan tersebut maka, kota Surabaya mempunyai sistem CCTV yang dapat membantu mengidentifikasi jenis pelanggaran, pelaku, hingga lokasi pelanggaran berada. Setelah mendata pelaku tersebut, petugas Dinas Perhubungan Kota Surabaya akan meneruskannya ke bagian penegakan hukum. Namun, para pelaku pelanggar lalu lintas belum mengetahui kapan serta dimana melakukan pelanggaran tersebut. Para pelaku pelanggar lalu lintas baru akan mengetahui pelanggaran mereka ketika membayar pajak kendaraan tersebut. Jenis pelanggaran yang terekam oleh CCTV yaitu kejadian dimana pelanggar menerobos lampu merah, melanggar marka jalan, dan salah jalur.

Salah satu penyebab terjadinya pelanggaran lalu lintas adalah kurangnya waspada para pengguna kendaraan ketika hendak melewati *traffic light*. Pada umumnya, pengguna kendaraan bermotor sering menggunakan kecepatan yang tinggi sehingga menyebabkan terjadinya kecelakaan dan secara tidak langsung perilaku tersebut melanggar peraturan lalu lintas. Maka dari itu, perlunya sebuah aplikasi yang mampu memberikan peringatan dini kepada pengguna untuk mengurangi kecepatan ketika hendak melewati *traffic light* dan memberikan informasi secepat mungkin agar para pelanggar mengetahui jenis pelanggaran dan berusaha tidak mengulangnya kembali. Dengan bantuan *Google Geofencing API*, maka suatu aplikasi dapat memulai interaksi dialog dengan pengguna jika perangkat *mobile* memasuki atau keluar dari area yang telah ditentukan (Garzon & Deva, 2014).

Berdasarkan permasalahan yang telah dijelaskan tersebut, akan dilakukan penelitian skripsi di Dinas Perhubungan Kota Surabaya dengan mengangkat judul “Pengembangan Aplikasi *Traffic Light* E-Tilang Menggunakan *Google Geofencing API* Berbasis Android” dengan menggunakan sistem android yang saat ini sudah banyak dipakai oleh masyarakat Indonesia, akan sangat mudah untuk membantu serta memudahkan masyarakat dalam mengetahui jenis pelanggaran, waktu pelanggaran serta lokasi pelanggaran yang telah dilanggar secara mudah diakses dengan menggunakan aplikasi yang ada di *smartphone* berupa android.

2. LANDASAN KEPUSTAKAAN

2.1 Dinas Perhubungan

Dinas Perhubungan kota Surabaya adalah salah satu instansi pemerintah yang mempunyai tugas pokok mengatur segala urusan transportasi dan perhubungan seperti tata letak rambu-rambu lalu lintas, membuat marka jalan, perparkiran, penetapan trayek dan tarif angkutan umum, memeriksa uji kelayakan kendaraan umum. Dinas Perhubungan sendiri mempunyai sistem cerdas untuk mendukung manajemen transportasi dengan pemanfaatan teknologi (informasi, komunikasi, sensor, kontrol dan komputerisasi) untuk membangun sistem informasi dan manajemen transportasi secara otomatis yang biasanya disebut *Intelligent Transportation System* (ITS).

Dinas Perhubungan kota Surabaya mulai melakukan pemberlakuan CCTV e-tilang sejak bulan September 2017. CCTV e-tilang merupakan penindakan setiap pelanggaran lalu lintas dengan alat bantu kamera CCTV yang terletak pada titik-titik strategis yang sudah ditentukan. Di Kota Surabaya saat ini sedang berjalan 2 titik strategis yang menerapkan CCTV e-tilang, namun di tahun 2018 ini akan ada penambahan 15 unit CCTV. Selain itu, Dinas Perhubungan kota Surabaya juga akan memasang kamera CCTV khusus untuk mengukur kecepatan kendaraan yang melintas di daerah yang rawan mengebuk seperti di Kenjeran dan Middle East Ring Road (Peni Widiarti, 2018).

Mekanisme cara kerja CCTV e-tilang ini yaitu pertama, data pelanggar akan terekam di CCTV, data tersebut berisikan nomor polisi pelanggar, jenis pelanggaran, serta tanggal kejadian pelanggaran. Kemudian, data yang sudah terekam oleh CCTV akan dikirimkan kepada pihak kepolisian untuk ditindak lanjuti. Jenis pelanggaran yang biasanya terekam oleh CCTV yaitu, melanggar lampu merah, melanggar marka jalan, melawan arah, dan melebihi batas garis saat lampu merah. Tujuan CCTV e-tilang ini adalah untuk mengedukasi masyarakat agar tertib berlalu lintas sehingga mengurangi angka kecelakaan.

2.2 Android Studio

Android Studio adalah sebuah IDE (*Integrated Development Environment*) yang digunakan untuk pengembangan aplikasi Android (Hafiz Herdi, 2014). Android

Studio merupakan IDE resmi untuk Android. Android Studio memiliki fitur editor kode cerdas (*Intelligent Code Editor*) yang memiliki kemampuan penyelesaian kode, optimalisasi, dan analisis kode yang canggih. Selain itu fitur *New Project Wizards* membuat proses memulai proyek baru menjadi jauh lebih mudah bahkan dapat mengimpor contoh kode Google dari GitHub. Berbagai modul baru digunakan dalam Android Studio ini, salah satunya adalah pengembangan aplikasi multi layar yang memudahkan pengembangan untuk membangun sebuah aplikasi untuk ponsel dan tablet Android, Android Wear, Android TV, Android Auto, dan Android Google Glass. Fitur-fitur Android Studio (Google Inc, 2017) Antara lain sebagai berikut:

1. Dibuat dengan *IntelliJ IDEA Community Edition*, JAVA IDE populer karya JetBrains.
2. Sistem pembuatan *Gradle* yang fleksibel.
3. Dapat membangun berbagai generasi dan varian APK.
4. Dukungan template bertambah untuk Google Services dan aneka tipe perangkat.
5. Editor layout yang lengkap dengan dukungan pengeditan tema.
6. Lint Tools untuk solusi kinerja, kegunaan, kompatibilitas versi, dan masalah lain.
7. ProGuard dan kemampuan app-signing.
8. Dukungan bawaan untuk *Google Cloud Platform*, mempermudah integrasi *Google Cloud Messaging* dan *App-Engine*.

Java dalam ilmu komputer, merupakan bahasa pemrograman berorientasi objek yang diperkenalkan pada tahun 1995 oleh Sun Microsystems Inc., yang saat Java diciptakan, dipimpin oleh James Gosling (Adi Nugroho, 2008). Bahasa pemrograman java tercipta berawal dari sebuah perusahaan Sun Microsystems yang ingin membuat sebuah bahasa pemrograman yang dapat berjalan di semua *device* tanpa harus terikat oleh platform yang digunakan oleh *device* tersebut, terlaksanalah sebuah proyek yang dipelopori oleh Patrick Naughton, James Gosling, Mike Sheridan dan Bill Joy pada tahun 1991, maka terciptalah bahasa pemrograman java yang awalnya bernama "Oak".

Java adalah sebuah teknologi dimana pada teknologi tersebut mencakup java sebagai bahasa pemrograman yang memiliki sintaks dan aturan pemrograman tersendiri, juga mencakup

java sebagai platform dimana teknologi ini memiliki *virtual machine* dan *library* yang diperlukan untuk menulis dan menjalankan program yang ditulis dengan bahasa pemrograman java (Isak Rickyanto, 2003). Alasan terbesar dalam pembuatan bahasa pemrograman java adalah keinginan akan terbentuknya suatu bahasa pemrograman yang bisa berjalan di berbagai perangkat tanpa harus terikat oleh *platform*, sehingga java ini bersifat *portable* dan *platform independent* (tidak tergantung mesin atau sistem operasi).

3. METODOLOGI

Alur dari penelitian ini dapat dijabarkan menjadi studi literatur, pengumpulan data, analisis kebutuhan dari pengguna, perancangan sistem sesuai kebutuhan pengguna, pembuatan *prototype* dan diujikan ke pengguna, evaluasi pengguna (jika kebutuhan pengguna belum tercapai pada fase *prototype*, maka akan dilakukan analisis kebutuhan kembali, jika kebutuhan telah sesuai maka proses akan berlanjut ke fase berikutnya), implementasi, pengujian dan analisis, serta pengambilan kesimpulan dan saran..

3.1. Analisis Kebutuhan

Pada tahap analisis kebutuhan dilakukan untuk mendapatkan semua kebutuhan yang diperlukan dari sistem yang akan dibangun. Analisis kebutuhan dilakukan dengan mengidentifikasi semua kebutuhan (*requirement*) perangkat lunak dan siapa saja yang terlibat di dalamnya. Analisis juga dilakukan untuk mengetahui kondisi lapangan yang ada sehingga dapat diketahui implementasi perangkat lunak yang akan digunakan. Kebutuhan-kebutuhan yang ditentukan akan dikategorikan menjadi kebutuhan fungsional dan kebutuhan non-fungsional. Kebutuhan-kebutuhan yang ditentukan berasal dari wawancara dengan pihak yang terlibat.

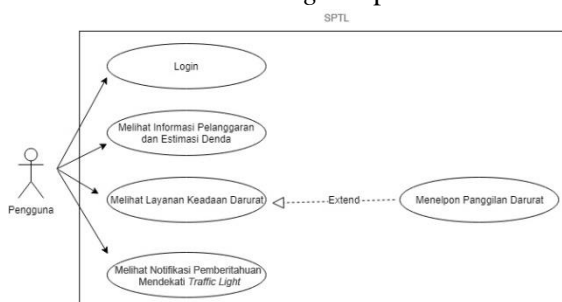
Pada tahap identifikasi aktor dihasilkan sebuah aktor yang berinteraksi dengan sistem, aktor tersebut adalah pengguna. Keterangan mengenai aktor pengguna dapat dilihat pada Tabel 1.

Tabel 1 Identifikasi Aktor

Nama Aktor	Deskripsi Aktor
Pengguna	Merupakan seseorang yang mempunyai hak akses untuk

	menggunakan semua fitur dari aplikasi yang telah dibuat.
--	--

Use case diagram merupakan diagram yang berfungsi menunjukkan hal-hal apa saja yang dapat dilakukan oleh aktor berdasarkan kebutuhan fungsional yang telah di deskripsikan. Kebutuhan fungsional yang didapatkan sebanyak empat kebutuhan. Berdasarkan kebutuhan fungsional yang telah di deskripsikan maka dibuat *use case diagram* pada Gambar 1.



Gambar 1 Use Case Diagram Sistem

3.2. Perancangan

Dalam tahap perancangan dibuat beberapa perancangan agar dapat digunakan untuk acuan dalam implementasi seperti perancangan diagram. Perancangan diagram yang dibangun adalah *sequence diagram* & *class diagram*. Setelah pembangunan diagram tersebut, selanjutnya merupakan perancangan algoritme atau perancangan komponen dan perancangan *user interface* (UI).

3.2.1 Perancangan Algoritme

Dalam perancangan algoritme dibuat sebuah algoritme yang berkaitan dengan implementasi. Pada bab ini akan dibahas salah satu algoritme dalam proses *login*.

Tabel 2 Perancangan Algoritme Login

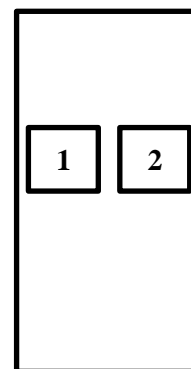
Start
Menekan tombol Informasi Pelanggaran
Menampilkan halaman input data berupa Form input
Input Nopol Kendaraan
Memindai kode QR
Jika Nopol Kendaraan == null
Sistem menampilkan pesan "Field is empty"
Selain itu
Membuka activity Scan
Jika, kode QR != nopol
Sistem menampilkan

notifikasi "Data tidak di temukan"
Selain itu
Sistem menampilkan halaman informasi pelanggaran
End

3.2.2 Perancangan Antarmuka

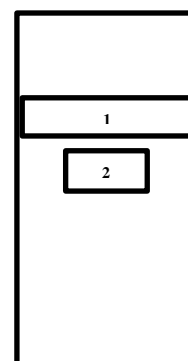
Perancangan antarmuka merupakan kerangka pembuatan sebuah aplikasi. Tujuan dari perancangan antarmuka untuk membuat sebuah tampilan dari aplikasi yang sederhana agar memudahkan pengguna dalam mengoperasikan aplikasi. Pada bagian ini akan dijelaskan antarmuka sistem *traffic light* e-tilang.

Antarmuka pada halaman menu utama menjelaskan bahwa pada halaman menu utama terdapat dua kategori utama yaitu untuk informasi pelanggaran dan panggilan layanan darurat. Pada Gambar 2 ditunjukkan perancangan antarmuka halaman menu utama.



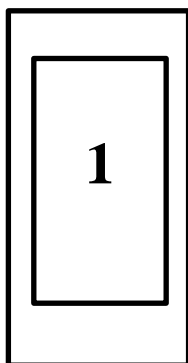
Gambar 2 Perancangan Antarmuka Halaman Menu Utama

Antarmuka pada halaman *input* nomor polisi menjelaskan bahwa pada halaman menu utama terdapat *form field* untuk mengisi nomor polisi kendaraan dan *button* untuk mengarahkan menuju halaman pindai kode. Pada Gambar 3 ditunjukkan perancangan antarmuka halaman *input* nomor polisi.



Gambar 3 Perancangan Antarmuka Input Nomor Polisi

Antarmuka pada halaman pindai kode menjelaskan bahwa pada halaman pindai kode terdapat *scanner camera* untuk memindai kode QR melalui kamera ponsel. Pada Gambar 4 ditunjukkan perancangan antarmuka halaman pindai kode.



Gambar 4. Perancangan Antarmuka Pindai Kode

3.3 Implementasi

Tahap implementasi akan dijelaskan mengenai implementasi yang dilakukan dalam pembuatan sistem. Sebelum implementasi dijelaskan spesifikasi yang digunakan dalam implementasi sistem. Spesifikasi tersebut berupa spesifikasi sistem yang digunakan, serta batasan dalam implementasi.

3.3.1 Implementasi Algoritme

Pada implementasi algoritme, pembahasan terkait dengan implementasi yang ada pada tahap perancangan algoritme.

Tabel 3 Kode Program Login

```
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.imageButton2:
            String mytext =
            inputnopol.getText().toString();

            if (TextUtils.isEmpty(mytext))
            {
                Toast.makeText(Input_nopol.this,
                "Fields is empty",
                Toast.LENGTH_LONG).show();
            } else {
                Intent intent = new
                Intent(Input_nopol.this,
                Scan_qr.class);
                intent.putExtra("Nopol",
                mytext);
                startActivity(intent);
            }
            break;
    }
}
```

```
public void handleResult(Result
result) {
    final String myResult =
    result.getText();
    final String tempNopol = nopol;
    this.nopol = null;

    if
    (myResult.equalsIgnoreCase(tempNopol)
    ){
        Intent intent = new
        Intent(Scan_qr.this,
        Info_pelanggaran.class);
        intent.putExtra("Nopol",
        myResult);
        startActivity(intent);
    }else{
        Toast.makeText(Scan_qr.this,
        "Data tidak cocok",
        Toast.LENGTH_LONG).show();
        finish();
    }
}
```

3.3.2 Implementasi Antarmuka

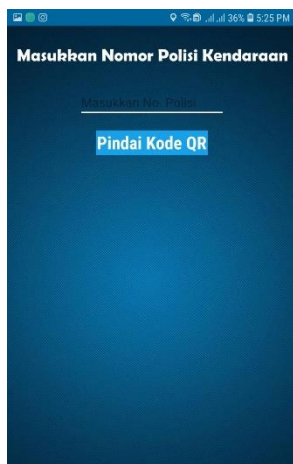
Implementasi antarmuka pada aplikasi sistem *traffic light* dibuat dengan bantuan pengolah gambar Adobe Photoshop. Implementasi antarmuka merupakan implementasi dari perancangan antarmuka sebelumnya.

Gambar 5 merupakan implementasi dari perancangan antarmuka halaman menu utama yang terdapat pada Gambar 2.



Gambar 5 Implementasi Antarmuka Menu Utama

Gambar 6 merupakan implementasi dari perancangan antarmuka halaman *input* nomor polisi kendaraan yang terdapat pada Gambar 3.



Gambar 6 Implementasi Antarmuka Input Nomor Polisi

Gambar 7 merupakan implementasi dari perancangan antarmuka halaman pindai kode yang terdapat pada Gambar 4.



Gambar 7 Implementasi Antarmuka Pindai Kode

4. Hasil dan Pembahasan

Dalam proses pengujian, terdapat dua pengujian yaitu pengujian validasi dan pengujian performa.

4.1 Pengujian Validasi

Pengujian validasi dilakukan untuk mengetahui bahwa fungsi-fungsi yang dibangun telah sesuai dengan kebutuhan fungsional. Dalam pengujian validasi dibuat kasus uji yang mengacu pada kebutuhan fungsional. Kasus uji dibuat untuk masing-masing kebutuhan fungsional. Hasil yang didapatkan dari pengujian validasi akan dianalisis untuk mendapatkan kesimpulan apakah aplikasi telah memenuhi kebutuhan

fungsional yang telah dideskripsikan atau belum.

Tabel 4 Hasil Pengujian Performa

Nomor Kasus Uji	Hasil yang Diharapkan	Status Validasi
SPTL_TEST_VAL_1	Aplikasi dapat menampilkan hasil pencarian informasi jenis pelanggaran dengan masukan nomor polisi kendaraan dan pemindaian kode QR yang dilakukan oleh pengguna.	Valid
SPTL_TEST_VAL_2	Aplikasi dapat menampilkan hasil informasi pelanggaran seperti jenis pelanggaran, waktu pelanggaran, lokasi pelanggaran dan estimasi denda.	Valid
SPTL_TEST_VAL_3	Aplikasi dapat menampilkan notifikasi di titik yang sesuai mendekati <i>traffic light</i> dengan tepat.	Valid
SPTL_TEST_VAL_4	Aplikasi dapat menampilkan nomor pada <i>dial pad</i> sesuai dengan tombol yang di tekan.	Valid

4.2 Pengujian Performa

Pengujian performa dilakukan dengan melakukan tiga percobaan terhadap aplikasi. Pengujian ini dilakukan untuk melihat respon sistem aplikasi pemantau *traffic light* dengan parameter jarak dan kecepatan kendaraan yang telah dibangun.

Tabel 5. Hasil Pengujian Performa

Pengujian Ke-	Kecepatan Kendaraan (km)	Jarak Radius (m)
1	20	84
2	40	40
3	60	22

5. Kesimpulan

Berdasarkan hasil analisis kebutuhan, perancangan, implementasi, dan pengujian pengembangan aplikasi pemantau *traffic light* e-tilang kota Surabaya berbasis android, maka dapat disimpulkan sebagai berikut:

1. Berdasarkan hasil analisis dan pemodelan kebutuhan didapatkan seorang aktor yang memiliki empat kebutuhan fungsional dan satu kebutuhan non fungsional, kebutuhan fungsional tersebut dimodelkan dalam *use case diagram* serta *use case scenario*.
2. Hasil dari tahapan perancangan menghasilkan perancangan basis data yang digambarkan dengan *entity relationship diagram*, perancangan arsitektur yang digambarkan dengan *sequence diagram* sebanyak empat kebutuhan fungsional dan lima *class diagram* yang terbentuk. Perancangan algoritma dibuat sebanyak dua algoritma dari fitur utama. Terakhir adalah perancangan *user interface* yang dibuat sebanyak lima perancangan dari kebutuhan fungsional.
3. Implementasi yang telah dilakukan menghasilkan implementasi basis data yang dikonversikan ke dalam JSON, implementasi *class diagram* menggunakan bahasa Java dengan *engine* Android Studio, implementasi *user interface* dilakukan dengan aplikasi Adobe Photoshop kemudian diimplementasikan kedalam *engine* Android Studio.
4. Pengujian dilakukan dengan *black box testing* untuk pengujian validasi kebutuhan fungsional dan pengujian akurasi dilakukan untuk memastikan akurasi dari fungsi sistem pemantau *traffic light*.

6. Daftar Pustaka

Garzon, S. R. & Deva, B., (2014). *Geofencing 2.0: Taking Location-based Notifications to the Next Level*.

UBICOMP.

Hadi, S. (2017). *Tilang CCTV Mampu Mengurangi Pelanggaran Lalu Lintas di Surabaya*. Tersedia di : <http://news.metrotvnews.com/read/2017/10/26/778715/tilang-cctv-mampu-mengurangi-pelanggaran-lalu-lintas-di-surabaya>

Herdi, H. (2014). *Belajar Membuat Aplikasi Android Menggunakan Android Studio*. Tersedia di : <http://www.twoh.co/2014/09/28/belajar-membuat-aplikasi-android-menggunakan-android-studio.html>

Larasati, W. A. (2017). Pelanggaran Lalu Lintas di Kota Surabaya Menurun Setelah Tilang Lewat CCTV. Tersedia di : <http://news.liputan6.com/read/3094520/pelanggaran-lalu-lintas-di-kota-surabaya-menurun-setelah-tilang-lewat-cctv>

Nugroho, A. (2008). *Pemrograman Java Menggunakan IDE Eclipse Callisto*. Andi Offset. Yogyakarta.

Rachman, A. F. (2015). Android Kuasai Asia Tenggara, di Indonesia Paling Juara. Tersedia di : <https://inet.detik.com/consumer/d-3054169/android-kuasai-asia-tenggara-di-indonesia-paling-juara>

Ridwan, M. (2017). *Denda Pelanggaran Lalu Lintas di Surabaya Rp. 10 Miliar*. Tersedia di : <https://www.lensaIndonesia.com/2017/12/28/denda-pelanggaran-lalu-lintas-di-surabaya-rp10-miliar.html>

Sommerville, Ian. (2011). *Software Engineering 9th edition*. Pearson Education, Inc.

Trenggono, E. P., Sukmaaji, A., & Taufik, V. M. (2012). *Rancang Bangun Sistem Informasi Kontrol Kondisi Lalu Lintas Dengan Kamera Pemantau CCTV Berbasis GIS*. 7-10.

Wahyudin. (2014). *Kendaraan di Surabaya Tambah 17 Ribu Lebih Sebulan*. Tersedia di : <http://www2.jawapos.com/baca/artikel/9796/kendaraan-di-surabaya-tambah-17-ribu-lebih-sebulan>

Widiarti, P. (2018). *Tahun Ini, CCTV e-Tilang di Surabaya Ditambah di 15 Titik*. Tersedia di : <http://kabar24.bisnis.com/read/20180104/78/723509/tahun-ini-cctv-e-tilang-surabaya-ditambah-di-15-titik>