

TUGAS AKHIR

**PENGEMBANGAN PERANGKAT LUNAK BERBASIS
ANDROID UNTUK MENGHITUNG KECEPATAN,
VOLUME, DAN KEPADATAN RUAS JALAN (STUDI
KASUS JALAN SILIWANGI YOGYAKARTA)
*DEVELOPMENT OF SOFTWARE BASED ON
ANDROID TO CALCULATE SPEED, VOLUME, AND
DENSITY
(CASE STUDY: SILIWANGI St, YOGYAKARTA)***

**Diajukan kepada Universitas Islam Indonesia Yogyakarta untuk Memenuhi
Persyaratan Memperoleh Derajat Sarjana Teknik Sipil**



**Kenzta Fazahudiya
14511187**

**PROGRAM STUDI TEKNIK SIPIL
FAKULTAS TEKNIK SIPIL DAN PERENCANAAN
UNIVERSITAS ISLAM INDONESIA
2021**

TUGAS AKHIR

PENGEMBANGAN PERANGKAT LUNAK BERBASIS ANDROID UNTUK MENGHITUNG KECEPATAN, VOLUME, DAN KEPADATAN RUAS JALAN (STUDI KASUS JALAN SILIWANGI YOGYAKARTA) DEVELOPMENT OF SOFTWARE BASED ON ANDROID TO CALCULATE SPEED, VOLUME, AND DENSITY (CASE STUDY: SILIWANGI St, YOGYAKARTA)

Disusun oleh:

Kenzta Fazahudiya
14511187

Telah diterima sebagai salah satu persyaratan untuk memperoleh derajat Sarjana
Teknik Sipil

Diuji pada tanggal: 30 November 2021

Oleh Dewan Penguji:

Pembimbing

Prima Juanita Romadhona, S.T., M.Sc.

NIK: 135111103

Penguji 1

Miftahul Fauziah, S.T., M.T., Ph.D.

NIK: 955110103

Penguji 2

Rizki Budi Utomo, S.T., M.T.

NIK: 045110406

Mengesahkan,

Ketua Program Studi Teknik Sipil



Dr. Ir. Sri Amini Yuni Astuti., M.T.

NIK: 885110101

PERNYATAAN BEBAS PLAGIASI

Saya menyatakan dengan sesungguhnya bahwa laporan Tugas Akhir yang saya susun sebagai syarat untuk memenuhi salah satu persyaratan pada Program Studi Teknik Sipil Universitas Islam Indonesia merupakan hasil karya saya sendiri. Adapun bagian-bagian tertentu dalam penulisan laporan Tugas Akhir yang saya kutip dari hasil karya orang lain telah dituliskan dalam sumbernya secara jelas sesuai dengan norma, kaidah, dan etika penulisan karya ilmiah. Apabila di kemudian hari ditemukan seluruh atau sebagian laporan Tugas Akhir ini bukan hasil karya saya sendiri atau adanya plagiasi dalam bagian-bagian tertentu, saya bersedia menerima sanksi, termasuk pencabutan gelar akademik yang saya sandang sesuai dengan perundang-undangan yang berlaku.

Yogyakarta, 15 Oktober 2021

Yang membuat pernyataan,



Kenzta Fazahudiya
(14511187)

KATA PENGANTAR

Puji syukur penulis ucapkan kepada Tuhan Yang Maha Esa karena penulis dapat menyelesaikan Tugas Akhir yang berjudul PENGEMBANGAN PERANGKAT LUNAK BERBASIS *ANDROID* UNTUK MENGHITUNG KECEPATAN, VOLUME, DAN KEPADATAN RUAS JALAN. Tugas Akhir ini merupakan salah satu syarat akademik dalam menyelesaikan studi tingkat strata satu di Prodi Teknik Sipil, Fakultas Teknik Sipil dan Perencanaan, Universitas Islam Indonesia Yogyakarta.

Dalam penyusunan Tugas Akhir ini banyak hambatan yang dihadapi penulis, namun berkat saran, kritik, serta dorongan semangat dari berbagai pihak, alhamdulillah Tugas Akhir ini dapat diselesaikan. Berkaitan dengan ini, penulis ingin mengucapkan terima kasih yang sedalam-dalamnya kepada:

1. Ibu Prima Juanita Romadhona, ST, M.Sc. selaku dosen pembimbing yang selalu memberi ilmu, nasihat, dan semangat.
2. Bapak Rizki Budi Utomo, S.T., M.T. dan Ibu Miftahul Fauziah, S.T., M.T., Ph.D. selaku dosen penguji 1 dan dosen penguji 2 Tugas Akhir.
3. Ibu Dr. Ir. Sri Amini Yuni Astuti., M.T. selaku Ketua Jurusan Teknik Sipil, Fakultas Teknik Sipil dan Perencanaan, Universitas Islam Indonesia, Yogyakarta.
4. Bapak dan Ibu penulis yang telah berkorban sangat banyak baik material maupun spiritual hingga selesainya Tugas Akhir ini.

Akhirnya Penulis berharap agar Tugas Akhir ini dapat bermanfaat bagi berbagai pihak yang membacanya.

Yogyakarta, Oktober 2021

Kenzta Fazahudiya
14511187

DAFTAR ISI

Halaman Judul	i
Halaman Pengesahan	ii
PERNYATAAN BEBAS PLAGIASI	iii
KATA PENGANTAR	iv
DAFTAR ISI	v
DAFTAR TABEL	vii
DAFTAR GAMBAR	viii
ABSTRAK	x
<i>ABSTRACT</i>	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	1
1.3 Tujuan Penelitian	2
1.4 Manfaat Penelitian	2
1.5 Batasan Penelitian	2
BAB II TINJAUAN PUSTAKA	4
2.1 Perbandingan Penghitung Kendaraan Otomatis dengan Manual	4
2.2 Perbandingan Penelitian Terdahulu dengan Penelitian yang Dilakukan	8
BAB III LANDASAN TEORI	12
3.1 Pengertian Umum	12
3.2 Survei Lalu Lintas	12
3.3 Kecepatan	13
3.4 Arus dan Komposisi Lalu Lintas	14
3.5 Satuan Mobil Penumpang	14
3.6 Kapasitas	15
3.7 Hubungan Kecepatan-Arus-Kepadatan Menurut Greenshields	17

3.8 Sistem Operasi <i>Android</i>	17
BAB IV METODOLOGI PENELITIAN	18
4.1 Lokasi dan Waktu Penelitian	18
4.2 Objek Penelitian	19
4.3 Analisis Data	19
4.4 Tahapan Penelitian	19
4.5 Metode Pengembangan Perangkat Lunak	20
4.6 Bagan Alir Penelitian	21
BAB V HASIL DAN PEMBAHASAN	23
5.1 Pengembangan Perangkat Lunak “ <i>Survey App</i> ”	23
5.2 Pengambilan Data Survei	36
5.3 Penerapan Perangkat Lunak “ <i>Survey App</i> ” dalam Membantu Perhitungan Kapasitas Ruas Jalan	39
5.4 Kelebihan dan Kekurangan Perangkat Lunak Android “ <i>Survey App</i> ”	45
BAB VI KESIMPULAN DAN SARAN	47
6.1 Kesimpulan	47
6.2 Saran	47
DAFTAR PUSTAKA	49
LAMPIRAN	50

DAFTAR TABEL

Tabel 2.1	Perbandingan Penelitian Terdahulu dengan Penelitian yang Dilakukan	9
Tabel 3.1	Ekivalensi Mobil Penumpang untuk Ruas Jalan Tak Terbagi	15
Tabel 5.1	Hasil Perhitungan Kecepatan, Volume, dan Kepadatan Menggunakan Perangkat Lunak <i>Survey App</i>	37
Tabel 5.2	Kalibrasi Data Greenshields untuk Survei dengan “ <i>Survey App</i> ”	42



DAFTAR GAMBAR

Gambar 3.1	Pemodelan Dasar Greenshields	17
Gambar 4.1	Peta Lokasi Penelitian	18
Gambar 4.2	Detail Lokasi Penelitian	18
Gambar 4.2	Bagan Alir Penelitian	22
Gambar 5.1	Bagan Alir Kinerja “ <i>Survey App</i> ”	26
Gambar 5.3	Halaman Utama “ <i>Survey App</i> ”	28
Gambar 5.4	<i>Input</i> Data Survei Kecepatan	29
Gambar 5.5	Perhitungan Waktu Tempuh Kendaraan	30
Gambar 5.6	<i>Output</i> Survei Kecepatan	31
Gambar 5.7	<i>Input</i> Data Survei Volume	32
Gambar 5.8	Perhitungan Jumlah Kendaraan	33
Gambar 5.9	<i>Output</i> Survei Volume	34
Gambar 5.10	Perhitungan Kepadatan	35
Gambar 5.11	<i>Output</i> Perangkat Lunak “ <i>Survey App</i> ”	36
Gambar 5.12	Diagram Pencar dan Persamaan Orde 2 dari Hubungan antara Kepadatan dengan Volume dari Survei dengan “ <i>Survey App</i> ”	40
Gambar 5.13	Diagram Pencar dan Persamaan Linear dari Hubungan antara Kepadatan dengan Kecepatan dari Survei dengan “ <i>Survey App</i> ”	40
Gambar 5.14	Diagram Pencar dan Persamaan Orde 2 dari Hubungan antara Volume dengan Kecepatan dari Survei dengan “ <i>Survey App</i> ”	41
Gambar 5.15	Hubungan Antara Kecepatan, Volume, dan Kepadatan dengan Model <i>Greenshields</i> dari Survei dengan “ <i>Survey App</i> ”	43

ABSTRAK

Dalam penelitian kinerja ruas jalan, tahapan survei di lapangan seringkali memerlukan peralatan survei yang sangat banyak dan waktu yang cukup lama. Di era dengan teknologi maju seperti sekarang, dibutuhkan suatu perangkat lunak, dalam hal ini perangkat lunak berbasis *Android*, yang dapat mempermudah surveyor dalam pengambilan data di lapangan, terutama kecepatan, volume, dan kepadatan kendaraan di ruas jalan.

Penelitian ini dilakukan dengan cara mengembangkan suatu perangkat lunak berbasis *Android* yang kemudian diuji langsung di ruas Jalan Siliwangi, Daerah Istimewa Yogyakarta untuk mendapatkan data yang diperlukan. Analisis meliputi nilai kecepatan kendaraan, volume, serta kepadatan ruas jalan. Kemudian nilai kecepatan, volume, dan kepadatan yang diperoleh dari survei menggunakan perangkat lunak ini kemudian digunakan untuk perhitungan kapasitas ruas jalan dengan model *Greenshields* dan metode MKJI 1997. Hasil perhitungan kapasitas dari kedua metode tersebut kemudian dibandingkan untuk mengetahui apakah perangkat lunak ini dapat digunakan untuk survei yang sebenarnya.

Hasil yang diperoleh dari penelitian ini berupa nilai kecepatan, volume, dan kepadatan yang diperoleh menggunakan perangkat lunak *Android*. Waktu yang dibutuhkan untuk pelaksanaan survei juga berkurang, dari yang sebelumnya membutuhkan waktu 90 – 120 menit menjadi 65 – 70 menit. Hasil kapasitas berdasarkan pemodelan *Greenshields* adalah sebesar 4323,672 smp/jam, sedangkan kapasitas yang diperoleh dari metode MKJI 1997 adalah 3762 smp/jam. Adanya selisih nilai kapasitas ruas Jalan Siliwangi antara pemodelan *Greenshields* dengan metode MKJI 1997 disebabkan karena jangka waktu survei yang hanya dilakukan 1 jam untuk keperluan pengujian perangkat lunak. Namun karena prinsip kerja perangkat lunak yang hanya menggantikan peran perangkat survei konvensional, perangkat lunak ini dapat digunakan untuk survei di lapangan.

Kata Kunci: *Android*, kecepatan, kepadatan, ruas jalan, survei, volume.

ABSTRACT

Data collection on the field requires a lot of instruments and takes too much time. So, in this day and age when technologies are far more sophisticated, a kind of software, in this case is for Android OS, is needed to help surveyors in collecting data in the field, especially the speed, volume, and vehicle density of road segments.

This research was conducted by developing an Android software that can be used for collecting speed, volume, and density of a road segment. This software is later be tested on Siliwangi Street, Yogyakarta. The output of the software, which are the speed, volume, and density on road segment, are used to determine the road segment capacity based on Greenshields' Model and MKJI 1997, which are then compared to see if the software is ready to be used on a real survey.

The results of this study is the output of the software, which are speed, volume, and density on road segment. Time spent on gathering data with Android software is also decreased, from 90 – 120 minutes to just 65 – 70 minutes. There are differences in the capacity on Siliwangi Street between Greenshields model and MKJI. The capacity based on Greenshields model is 4323,672 pcu/hr, and the capacity based on MKJI 1997 is 3762 pcu/hr. This is caused by the duration of the survey conducted in this research, which was just 1 hour for testing purposes. But, because this software is designed to mimick the function of conventional tools, it is able to be used on a real survey.

Keywords: *Android, density, road segment, speed, survey, volume*

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam melakukan survei lalu lintas di lapangan dibutuhkan waktu, biaya, serta perlengkapan survei tidak sedikit. Oleh karena itu, di era globalisasi yang menuntut pekerjaan yang cepat, tepat, dan efisien seperti saat ini, dibutuhkan suatu teknologi berupa perangkat lunak yang dapat mempermudah pelaksanaan survei di lapangan.

Berdasarkan permasalahan yang telah dikemukakan di atas, maka pada Tugas Akhir ini dilakukan penelitian mengenai **“PENGEMBANGAN PERANGKAT LUNAK BERBASIS ANDROID UNTUK MENGHITUNG KECEPATAN, VOLUME, DAN KEPADATAN RUAS JALAN”** untuk mempermudah pelaksanaan survei. Pengembangan perangkat lunak ini dibuat seiring dengan berkembangnya dunia teknologi informasi di Indonesia yang menuntut seluruh pekerjaan agar dapat dilakukan dengan cepat.

Perangkat lunak ini akan diuji dengan cara melaksanakan survei kecepatan, volume, dan kepadatan pada ruas Jalan Siliwangi, Kabupaten Sleman, Daerah Istimewa Yogyakarta. Hasil dari perangkat lunak ini akan digunakan untuk memenuhi data yang dibutuhkan dalam perhitungan kapasitas ruas jalan raya dengan metode *Greenshields* dan persamaan yang ditetapkan oleh Direktorat Jenderal Bina Marga.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang dikemukakan di atas, maka dirumuskanlah permasalahan penelitian ini sebagai berikut ini.

1. Bagaimana perangkat lunak berbasis *Android* untuk perhitungan kecepatan, volume, dan kepadatan di ruas jalan raya perkotaan akan dikembangkan?
2. Bagaimana kegunaan perangkat lunak berbasis *Android* ini dalam bidang ketekniksipilan?

3. Apakah perangkat lunak *Android* ini layak digunakan dalam survei di lapangan?

1.3 Tujuan Penelitian

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut ini.

1. Mengetahui bagaimana perangkat lunak berbasis *Android* untuk perhitungan kecepatan, volume, dan kepadatan di ruas jalan dikembangkan.
2. Mengetahui kegunaan perangkat lunak *Android* dengan menghitung kapasitas berdasarkan metode *Greenshields* dan *MKJI 1997* menggunakan data kecepatan, volume, dan kepadatan dari perangkat lunak.
3. Mengetahui apakah perangkat lunak *Android* ini layak digunakan dalam survei di lapangan dengan membandingkan nilai kapasitas ruas jalan antara metode *Greenshields* dengan metode *MKJI 1997*.

1.4 Manfaat Penelitian

Adapun manfaat yang ingin dicapai dalam penelitian ini adalah sebagai berikut ini.

1. Memberi informasi mengenai pengembangan perangkat lunak *Android* untuk membantu pelaksanaan survei kecepatan, volume, dan kepadatan ruas jalan.
2. Dapat dijadikan bahan referensi bila ada pengembangan lebih lanjut dari perangkat lunak yang digunakan dalam penelitian ini.

1.5 Batasan Penelitian

Dalam penelitian yang dilakukan, ada beberapa lingkup penelitian yang dibatasi untuk mencapai maksud dan tujuan sebagai berikut ini.

1. Melakukan pengembangan perangkat lunak berbasis sistem operasi *Android* untuk pelaksanaan survei kecepatan, volume, dan kepadatan ruas jalan.
2. Penelitian ini mengambil studi kasus pada Jalan Siliwangi, Kabupaten Sleman, Daerah Istimewa Yogyakarta, yang merupakan jalan antar kota.
3. Kendaraan yang disurvei dalam penelitian ini adalah kendaraan ringan (LV) dan kendaraan berat (HV).

4. Perhitungan kapasitas di ruas Jalan Siliwangi dilakukan dengan metode *Greenshields* dan metode dari Direktorat Jenderal Bina Marga, dengan data survei yang dihasilkan dari perangkat lunak *Android* yang dikembangkan dalam penelitian ini.



BAB II

TINJAUAN PUSTAKA

2.1 Penelitian Serupa yang Pernah Dilakukan

Pal'ó, et. Al (2019) melakukan penelitian mengenai perbandingan alat hitung kendaraan otomatis dengan perhitungan jumlah kendaraan secara manual. Penelitian ini mengambil studi kasus di Kota Košice dan Desa Palárikovo, Slovakia. Tujuan dari penelitian ini antara lain untuk mengetahui tingkat akurasi dari kedua metode perhitungan jumlah kendaraan. Perhitungan jumlah kendaraan secara otomatis dilakukan menggunakan alat yang bernama Sierzega SR04, sedangkan perhitungan kendaraan secara manual dilakukan melalui rekaman video.

Perhitungan jumlah kendaraan secara manual memiliki sebuah keuntungan yaitu kemungkinan kesalahan hitung yang sedikit, bahkan pada umumnya terjadi di bawah 1%. Namun metode perhitungan ini memiliki kekurangan yaitu tingkat kesalahan dalam menentukan jenis kendaraan yang lebih tinggi, yaitu sekitar 4-5%. Hal ini disebabkan karena kesulitan surveyor dalam menentukan jenis kendaraan berdasarkan panjang kendaraan melalui rekaman video.

Sierzega SR04 memiliki kelemahan dalam kemudahan instalasi, di mana alat ini hanya bisa digunakan di daerah tertentu serta dibutuhkan beberapa ketentuan yang harus dipenuhi agar alat ini bisa bekerja secara optimal. Namun selain mampu menghitung jumlah kendaraan, alat ini dapat mendeteksi panjang kendaraan secara akurat sehingga alat ini dapat menentukan jenis kendaraan yang dihitung.

Dari penelitian yang dilakukan, perbedaan hasil perhitungan Sierzega SR04 dengan hasil perhitungan manual di Kota Košice adalah sebesar 2%, sedangkan untuk perhitungan yang dilakukan di Desa Palárikovo diperoleh perbedaan sebesar 3%. Hal ini menunjukkan bahwa Sierzega SR04 memiliki tingkat akurasi yang sangat bagus.

Romadhona dan Chasanah (2017) melakukan penelitian dengan membuat perangkat lunak bernama PRIFACOUNTER. Tujuan dari pembuatan perangkat lunak ini adalah untuk mempermudah pelaksanaan survei kinerja lalu lintas. Melalui alamat *prifacounter.uir.ac.id*, perangkat lunak ini dapat diakses melalui *PC*, *smartphone*, atau *tablet*. Untuk gawai dengan sistem operasi *Android* dipermudah dengan adanya aplikasi PRIFACOUNTER yang berbentuk *web view*, yang berarti aplikasi tersebut secara otomatis mengakses alamat web PRIFACOUNTER ketika dibuka.

Sejauh ini, PRIFACOUNTER dapat digunakan untuk perhitungan volume pada ruas jalan dan simpang tidak bersinyal. Aplikasi ini bekerja dengan cara memasukkan data survei dan dilanjutkan dengan memasukkan jumlah kendaraan yang lewat sesuai dengan tipe kendaraan tersebut, sehingga diperoleh kinerja lalu lintas berupa volume, kapasitas, serta derajat kejenuhan. Perhitungan yang dilakukan oleh aplikasi ini adalah berdasarkan Manual Kapasitas Jalan Indonesia (MKJI).

Dalam perhitungan volume lalu lintas dibutuhkan beberapa variabel meliputi tipe jalan serta jenis kendaraan. Sedangkan untuk perhitungan kapasitas dibutuhkan beberapa variabel meliputi tipe jalan, lebar jalan, jenis pemisah arah, lebar bahu, kerib, serta ukuran kota. Perhitungan derajat kejenuhan dilakukan melalui pembagian nilai volume dengan kapasitas.

Beberapa langkah dilakukan dalam penggunaan aplikasi PRIFACOUNTER antara lain pengisian data masukan, perhitungan volume lalu lintas, perhitungan kapasitas, serta perhitungan derajat kejenuhan. Data masukan yang dimaksud terdiri atas kondisi umum jalan, kondisi geometrik jalan, hambatan samping, serta data survei. Sedangkan perhitungan volume dilakukan berdasarkan jenis kendaraan yang lewat meliputi kendaraan ringan atau mobil penumpang, kendaraan berat, serta sepeda motor. Kapasitas ruas jalan dihitung dengan cara mengalikan kapasitas dasar dengan faktor-faktor penyesuaian kapasitas. Hasil perhitungan volume dan kapasitas digunakan untuk menentukan nilai derajat kejenuhan.

Dari penelitian yang telah dilakukan dapat disimpulkan bahwa aplikasi PRIFACOUNTER dapat digunakan secara individu melalui telepon seluler dengan sistem operasi *Android* untuk menentukan kinerja lalu lintas berupa volume, kapasitas, serta derajat kejenuhan. Aplikasi yang dapat diakses bagi semua orang mampu menurunkan jangka waktu pelaksanaan survei.

Romadhona dan Chasanah (2017) juga melakukan validasi dalam mengembangkan aplikasi PRIFACOUNTER. Validasi dilakukan dengan membandingkan hasil perhitungan aplikasi PRIFACOUNTER dengan hasil perhitungan survei manual.

Validasi data dilakukan menggunakan program SPSS. Metode yang digunakan dalam pengujian hipotesis adalah *Paired Sample T-Test* dan *Sign Test*. Adapun hipotesis yang digunakan dalam penelitian ini adalah sebagai berikut.

H_0 = Tidak terdapat perbedaan yang signifikan antara data yang diambil dengan perhitungan manual dengan data dari aplikasi PRIFACOUNTER

H_1 = Terdapat perbedaan yang signifikan antara data yang diambil dengan perhitungan manual dengan data dari aplikasi PRIFACOUNTER

H_0 diterima bila terdapat perbedaan sebesar 0,05.

Dari penelitian yang dilakukan diperoleh seluruh data hasil *Paired T-Test* dan *Sign Test* menunjukkan angka di atas 0,05, serta persentase perbedaan yang diperoleh adalah $\leq 1\%$. Sehingga dapat disimpulkan bahwa H_0 diterima, yaitu tidak ada perbedaan yang signifikan antara perhitungan volume kendaraan dengan cara manual dengan perhitungan menggunakan aplikasi PRIFACOUNTER.

Yulianto dan Yuliansyah (2015) melakukan penelitian dengan membuat aplikasi penghitung kendaraan menggunakan teknologi *Radio Frequency Identification* (RFID). Aplikasi ini dibuat untuk mengurangi adanya permasalahan dalam proses perhitungan yang diakibatkan oleh kurangnya kelincahan serta kondisi fisik dari surveyor.

Penelitian ini menghasilkan rancangan sistem penghitung kendaraan dengan memanfaatkan RFID serta perangkat lunak untuk mengelola data perhitungan kendaraan dengan teknologi RFID dan *webcam* yang dapat menjadi informasi status kemacetan serta memudahkan saat pendokumentasian perhitungan kendaraan bermotor.

Pengujian aplikasi dilakukan dengan metode *blackbox test* dan *whitebox test*. Berdasarkan hasil *blackbox test*, setiap fungsi yang terdapat pada perangkat lunak berfungsi sesuai rancangan. Sedangkan pada *whitebox test*, semua logika struktur kontrol berdasarkan *test case* yang terdapat pada kode program berjalan dengan baik. Hal ini membuktikan bahwa logika struktur kontrol aplikasi sudah berjalan dengan benar.

Bellucci dan Cipriani (2010) melakukan sebuah penelitian dengan membandingkan 7 jenis sistem yang biasa digunakan dalam perhitungan kendaraan secara otomatis. Adapun 7 sistem yang digunakan adalah induksi elektromagnetik, pendeteksi beban kendaraan menggunakan sensor *piezoelectric*, pendeteksi beban kendaraan menggunakan sensor *quartz*, rekaman video, metode *double* (gabungan radar dan laser), metode *triple* (gabungan radar, sensor pasif, dan sensor *ultracoustic*), serta teknologi laser.

Penelitian ini bertujuan untuk mengetahui sistem yang tepat untuk digunakan dalam beberapa kasus tertentu. Selain itu, penelitian ini diharapkan dapat meningkatkan ketelitian dalam survei lalu lintas seperti pengambilan data statistik, desain jalan, serta manajemen lalu lintas. Penelitian ini dilakukan dalam jangka waktu satu tahun di ruas jalan raya S. S. 1 Aurelia yang terletak pada 23 km dari kota Roma, Italia.

Faktor utama yang memengaruhi efektivitas masing-masing sistem adalah sensitivitas terhadap lingkungan sekitar, seperti cahaya dan hujan. Penelitian ini menghasilkan kesimpulan bahwa tingkat ketelitian yang tinggi dicapai oleh sistem yang menggunakan teknologi gabungan, sedangkan sistem yang hanya menggunakan satu jenis teknologi menghasilkan performa yang rendah. Namun

tingkat ketelitian dan reliabilitas tidak selalu berbanding lurus, sehingga kekuatan suatu sistem juga menjadi pertimbangan agar alat tersebut dapat bekerja dengan presisi.

2.2 Perbandingan Penelitian Terdahulu dengan Penelitian yang Dilakukan

Beberapa penelitian mengenai perhitungan kecepatan, volume, dan kepadatan ruas jalan raya telah dibahas pada penelitian terdahulu. Perbandingan penelitian terdahulu dengan penelitian yang dilakukan dapat dilihat pada Tabel 2.1 berikut.



Tabel 2.1 Perbandingan Penelitian Terdahulu dengan Penelitian yang Dilakukan

Judul	Peneliti	Tujuan	Lokasi Penelitian	Metode Penelitian	Hasil Penelitian
<i>The Comparison of Automatic Traffic Counting and Manual Traffic Counting</i>	Pal'o et. Al (2019)	Mengetahui tingkat akurasi dari metode perhitungan jumlah kendaraan secara otomatis maupun manual	Kota Košice dan Desa Palárikovo, Slovakia	Penghitung kendaraan otomatis Sierzega SR04 dan rekaman video	Perbedaan hasil perhitungan Sierzega SR04 dengan hasil perhitungan manual di Kota Košice adalah sebesar 2%, sedangkan untuk perhitungan yang dilakukan di Desa Palárikovo diperoleh perbedaan sebesar 3%, yang membuktikan bahwa Sierzega SR04 memiliki tingkat akurasi yang sangat bagus.
<i>Prifacounter: The Innovation of Traffic Counter Application for Android OS</i>	Romadhona dan Chasanah (2017)	Membuat perangkat lunak berbasis Android untuk mempermudah pelaksanaan survey kinerja lalu lintas	Jalan Parangtritis, Yogyakarta	Metode <i>Research and Development</i>	Aplikasi PRIFACOUNTER dapat digunakan secara individu melalui telepon seluler dengan sistem operasi Android mampu menurunkan jangka waktu pelaksanaan survey serta dapat menentukan kinerja lalu lintas berupa volume, kapasitas, dan derajat kejenuhan.

Sumber: Pal'o et. Al (2019), Romadhona dan Chasanah (2017)

Lanjutan Tabel 2.1 Perbandingan Penelitian Terdahulu dengan Penelitian yang Dilakukan

Judul	Peneliti	Tujuan	Lokasi Penelitian	Metode Penelitian	Hasil Penelitian
<i>Prifacounter Validation: Technology Development Innovation Of Traffic Counter Application</i>	Romadhona dan Chasanah (2017)	Melakukan Validasi Aplikasi PRIFACOUNTER	Jalan Malioboro, Jalan C. Simanjuntak, Jalan Affandi, Jalan L. Adisucipto, Dr. Jalan Wahidin S., Jalan Seturan Jalan Raya, Sudirman, Jalan Magelang, Yogyakarta	Metode Uji T dan Uji Tanda	Hasil Paired T-Test dan Sign Test menunjukkan tidak ada perbedaan yang signifikan antara perhitungan volume kendaraan dengan cara manual dengan perhitungan menggunakan aplikasi PRIFACOUNTER.
Rancang Bangun Aplikasi Traffic Counter RFID	Yulianto dan Yuliansyah (2015)	Pembuatan Aplikasi Traffic Counter RFID	Posko <i>monitoring</i> daerah Kulon Progo dan Kota Cirebon	Metode <i>Blackbox Test</i> dan <i>Whitebox Test</i>	<i>Blackbox test</i> membuktikan bahwa setiap fungsi yang terdapat pada perangkat lunak berfungsi sesuai rancangan. Sedangkan <i>whitebox test</i> membuktikan bahwa logika struktur kontrol aplikasi sudah berjalan dengan benar.

Sumber: Romadhona dan Chasanah (2017) Yulianto dan Yuliansyah (2015)

Lanjutan Tabel 2.1 Perbandingan Penelitian Terdahulu dengan Penelitian yang Dilakukan

Judul	Peneliti	Tujuan	Lokasi Penelitian	Metode Penelitian	Hasil Penelitian
<i>Data Accuracy on Automatic Traffic Counting: The SMART Project Results</i>	Bellucci dan Cipriani (2010)	Memilih sistem yang tepat untuk digunakan dalam beberapa kasus tertentu dan meningkatkan ketelitian dalam survei lalu lintas seperti pengambilan data statistik, desain jalan, serta manajemen lalu lintas	Ruas jalan raya S. S. 1 Aurelia, Roma, Italia.	Sensor <i>piezoelectric</i> , sensor <i>quartz</i> , rekaman video, metode <i>double</i> metode <i>triple</i> , serta teknologi laser.	Tingkat ketelitian sistem yang menggunakan teknologi gabungan sangat tinggi, sedangkan sistem yang hanya menggunakan satu jenis teknologi menghasilkan performa yang rendah. Namun tingkat ketelitian dan reliabilitas tidak selalu berbanding lurus, sehingga kekuatan suatu sistem juga menjadi pertimbangan agar alat tersebut dapat bekerja dengan presisi.

Sumber: Bellucci dan Cipriani (2010)

BAB III

LANDASAN TEORI

3.1 Pengertian Umum

Berdasarkan Undang-undang No. 22 Tahun 2009 tentang Lalu Lintas dan Angkutan Jalan, lalu lintas didefinisikan sebagai gerak kendaraan dan orang di ruang lalu lintas jalan.

Menurut Direktorat Jenderal Bina Marga, kecepatan merupakan jarak yang ditempuh suatu kendaraan dalam satuan waktu tertentu, sehingga memiliki satuan km/jam.

Pressman (2002) menyatakan bahwa perangkat lunak atau *software* merupakan suatu perintah program dalam sebuah komputer, yang apabila dieksekusi oleh penggunanya dapat memberikan fungsi dan juga unjuk kerja yang diinginkan oleh pengguna. Pernyataan ini menggambarkan bahwa *software* atau perangkat lunak ini berfungsi untuk memerintah komputer, agar komputer tersebut dapat berfungsi secara optimal, sesuai dengan keinginan pengguna atau *brainware* yang memberikan perintah kepadanya.

Daulay (2007) menyatakan bahwa perangkat lunak atau *software* itu sendiri merupakan sebuah perangkat yang berfungsi sebagai pengatur aktivitas kerja komputer dan semua instruksi yang mengarah kepada sebuah sistem komputer. Lebih lanjut disebutkan pula bahwa sebuah perangkat lunak merupakan sebuah perangkat yang menjembatani interaksi pengguna dengan komputer yang menggunakan bahasa mesin.

3.2 Survey Lalu Lintas

Putranto (2016: 39) berpendapat bahwa survei lalu lintas merupakan salah satu bagian dari studi transportasi yang bertujuan untuk mengumpulkan data yang kemudian dianalisis baik untuk keperluan pengambilan keputusan pada tingkat perencanaan, perancangan, maupun evaluasi. Adapun unsur penting dalam melakukan survei lalu lintas antara lain tujuan survei, metode survei, surveyor, serta peralatan.

Tujuan survei harus sesuai dengan tujuan studi transportasi serta dinyatakan secara jelas karena berkaitan langsung dengan metode survei. Begitu pula sebaliknya, metode survei harus sesuai dengan tujuan survei dan memungkinkan untuk dilaksanakan baik ditinjau dari aspek legalitas, ketersediaan teknologi, kondisi lokasi, dan sebagainya. Metode survei juga harus mempertimbangkan keterbatasan waktu, biaya, serta personil.

Hal-hal yang perlu dipersiapkan sebelum melaksanakan survei lalu lintas adalah sebagai berikut:

- a. Merancang survei dengan mempertimbangkan hasil dan metode survei yang pernah dilakukan.
- b. Melaksanakan survei pendahuluan untuk membuat sketsa lokasi, merencanakan posisi surveyor, serta pemilihan peralatan yang digunakan.
- c. Melaksanakan rekrutmen serta pelatihan surveyor.
- d. Mempersiapkan peralatan dan formulir survei.

3.3 Kecepatan

Kecepatan didefinisikan sebagai jarak yang ditempuh oleh suatu benda per satuan waktu. Adapun satuan internasional untuk kecepatan adalah meter per detik, namun satuan yang sering digunakan dalam kehidupan sehari-hari adalah kilometer per jam atau mil per jam.

Menurut Hewitt (2006), Galileo Galilei adalah ilmuwan yang sering dianggap sebagai orang pertama yang mengukur kecepatan berdasarkan jarak yang ditempuh serta waktu yang dibutuhkan. Sehingga dari pernyataan tersebut, dihasilkan rumus kecepatan sebagai berikut.

$$v = \frac{d}{t}$$

Dari rumus tersebut, v merupakan kecepatan, d merupakan jarak, serta t merupakan waktu. Sebagai contoh, apabila ada pesepeda yang menempuh jarak 30 meter dalam 2 detik memiliki kecepatan sebesar 15 m/s. Adapun untuk mengubah satuan tersebut menjadi km/h adalah dengan mengalikan nilai kecepatan dengan 3,6. Sehingga kecepatan pesepeda tersebut adalah 54 km/h.

Hobbs (1979: 66) menyatakan bahwa secara umum kecepatan dibagi menjadi 3 (tiga) jenis yaitu kecepatan setempat (*spot speed*), kecepatan bergerak (*running speed*), dan kecepatan perjalanan (*journey speed*). Kecepatan setempat adalah kecepatan kendaraan pada suatu saat diukur dari suatu tempat yang ditentukan. Kecepatan bergerak adalah kecepatan kendaraan rata-rata pada suatu jalur pada saat kendaraan bergerak dan didapat dengan membagi panjang jalur dibagi dengan lama waktu kendaraan bergerak menempuh jalur tersebut. Kecepatan perjalanan adalah kecepatan efektif kendaraan yang sedang dalam perjalanan antara dua tempat, dan merupakan jarak antara dua tempat yang dibagi dengan lama waktu bagi kendaraan untuk menyelesaikan perjalanan antara dua tempat tersebut, dengan lama waktu ini mencakup setiap waktu berhenti yang ditimbulkan oleh hambatan (tundaan) lalu lintas.

3.4 Arus dan Komposisi Lalu Lintas

Menurut Direktorat Jenderal Bina Marga, Arus lalu lintas merupakan jumlah kendaraan yang melalui suatu titik pada jalan dalam satuan waktu. Arus lalu lintas dinyatakan dalam satuan kendaraan penumpang (smp) per jam, sehingga semua nilai arus lalu lintas (per arah dan total) harus dikonversikan ke dalam satuan mobil penumpang (smp) dengan menggunakan ekivalensi mobil penumpang (emp) yang diturunkan secara empiris untuk tipe kendaraan berikut.

- a. Kendaraan ringan (LV), mencakup mobil penumpang, minibus, *pick-up*, truk kecil, dan *jeep*;
- b. Kendaraan berat (HV), mencakup truk dan bus;
- c. Sepeda motor (MC).

Nilai emp untuk masing-masing tipe kendaraan tergantung pada tipe jalan dan arus lalu lintas total yang dinyatakan dalam satuan kendaraan per jam.

3.5 Satuan Mobil Penumpang

Menurut Direktorat Jenderal Bina Marga, satuan mobil penumpang (smp) merupakan satuan arus lalu lintas di mana arus dari berbagai tipe kendaraan diubah menjadi kendaraan ringan dengan menggunakan nilai ekivalensi mobil penumpang (emp). Untuk memperoleh nilai arus lalu lintas dalam satuan mobil penumpang,

dibutuhkan nilai ekivalensi mobil penumpang atau emp. Nilai emp untuk setiap jenis kendaraan antara lain tergantung pada konfigurasi jalan. Dalam penelitian ini, Jalan Siliwangi merupakan jalan antar kota yang memiliki empat lajur, dua arah, serta terdapat median (4/2 D). Berikut merupakan nilai emp untuk ruas jalan empat lajur dua arah menurut Direktorat Jenderal Bina Marga.

Tabel 3.1 Ekivalensi Mobil Penumpang untuk Ruas Jalan Luar Kota Empat Lajur Dua Arah

Tipe alinyemen	Arus total (kend/jam)		emp			
	Jalan terbagi per arah kend/jam	Jalan tak terbagi total kend/jam	MHV	LB	LT	MC
Datar	0	0	1,2	1,2	1,6	0,5
	1000	1700	1,4	1,4	2,0	0,6
	1800	3250	1,6	1,7	2,5	0,8
	> 2150	> 3950	1,3	1,5	2,0	0,5
Bukit	0	0	1,8	1,6	4,8	0,4
	750	1350	2,0	2,0	4,6	0,5
	1400	2500	2,2	2,3	4,3	0,7
	> 1750	> 3150	1,8	1,9	3,5	0,4
Gunung	0	0	3,2	2,2	5,5	0,3
	550	1000	2,9	2,6	5,1	0,4
	1100	2000	2,6	2,9	4,8	0,6
	> 1500	> 2700	2,0	2,4	3,8	0,3

Sumber: Direktorat Jenderal Bina Marga

3.6 Kapasitas

Menurut Direktorat Jenderal Bina Marga, kapasitas merupakan arus maksimum pada suatu bagian jalan yang dapat dipertahankan pada kondisi tertentu. Jika memungkinkan, nilai kapasitas diamati melalui pengumpulan data lapangan. Namun kapasitas juga dapat diperkirakan dari analisis kondisi iringan lalu lintas dan secara teoritis dengan mengasumsikan hubungan matematik antara arus, kecepatan, dan kepadatan. Kapasitas dinyatakan dalam satuan mobil penumpang (smp).

Adapun persamaan untuk menentukan kapasitas jalan raya menurut Direktorat Jenderal Bina Marga adalah sebagai berikut.

$$C = C_o \times FC_W \times FC_{SP} \times FC_{SF} \times FC_{CS}$$

Dimana:

C_o = Kapasitas (smp/jam)

FC_W = Kapasitas dasar (smp/jam)

FC_{SP} = Faktor penyesuaian lebar jalan

FC_{SF} = Faktor penyesuaian hambatan samping

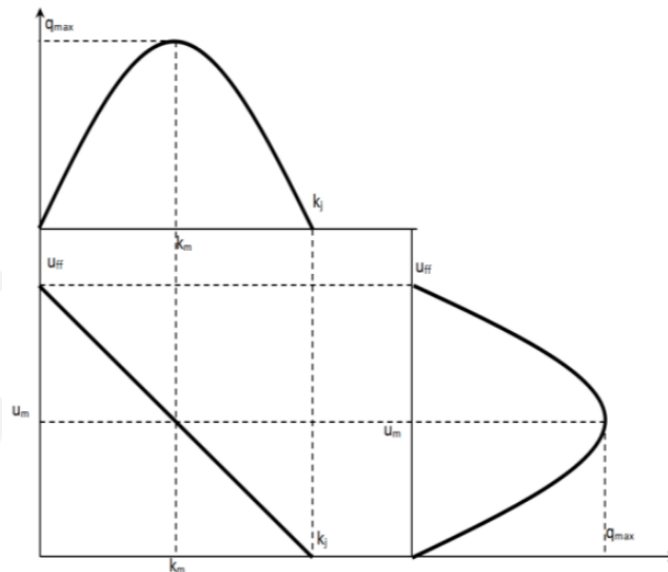
FC_{CS} = Faktor penyesuaian ukuran kota

Jika pada kondisi sesungguhnya sama dengan kondisi ideal yang ditentukan sebelumnya, maka semua faktor penyesuaian menjadi 1,0 sehingga kapasitas menjadi sama dengan kapasitas dasar.

3.7 Hubungan Kecepatan-Arus-Kepadatan Menurut *Greenshields*

Dalam analisis kapasitas segmen jalan terdapat prinsip dasar yaitu kecepatan akan berkurang ketika arus bertambah. Ketika arus pada ruas jalan tersebut rendah, maka pengurangan kecepatan akibat penambahan arus memiliki nilai yang kecil. Semakin tinggi nilai arus maka pengurangan kecepatan akan semakin besar.

Dalam prinsip *Greenshields*, hubungan antara kepadatan dengan arus dan hubungan antara arus dengan kecepatan digambarkan dalam kurva polinomial. Sedangkan hubungan antara kecepatan dengan arus digambarkan dalam kurva linear. Adapun grafik hubungan kecepatan-arus-kepadatan menurut *Greenshields* dapat dilihat pada Gambar 3.1 berikut.



Gambar 3.1 Pemodelan Dasar Greenshields

Sumber: Direktorat Jenderal Bina Marga

Hubungan ini telah ditentukan secara kuantitatif untuk kondisi standar untuk setiap tipe jalan. Setiap kondisi standar mempunyai geometrik dan karakteristik lingkungan tertentu. Jika karakteristik jalan lebih baik dari kondisi standar, seperti memiliki lebar yang lebih dari jalur lalu lintas normal, maka nilai kapasitas akan semakin tinggi, diikuti dengan kecepatan yang lebih tinggi pada arus tertentu. Sedangkan jika karakteristik jalan lebih buruk dari kondisi standar, seperti adanya hambatan samping yang tinggi, maka nilai kapasitas akan semakin berkurang dan kecepatan pada arus tertentu menjadi lebih rendah.

3.8 Sistem Operasi *Android*

Android adalah sistem operasi yang saat ini banyak digunakan pada perangkat bergerak, salah satunya adalah telepon seluler cerdas atau *smartphone*. Sistem operasi ini pada mulanya dikembangkan oleh *Android Inc.* yang kemudian diambil alih oleh *Google* pada tahun 2005. Sistem operasi *Android* dirilis pada tahun 2008 dan bersifat *open source*, yang artinya siapa saja dapat menggunakannya secara gratis, termasuk penggunaan kode sumber yang sangat berguna dalam pengembangan sistem operasi tersebut.

BAB IV

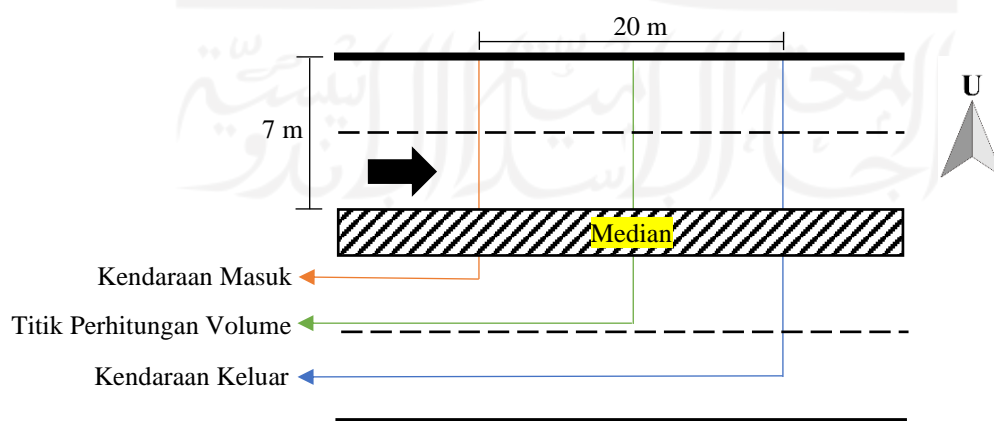
METODOLOGI PENELITIAN

4.1 Lokasi dan Waktu Penelitian

Penelitian ini mengambil studi kasus pada ruas Jalan Siliwangi, Kabupaten Sleman, Daerah Istimewa Yogyakarta. Waktu penelitian dilaksanakan pada bulan Mei hingga bulan Juni 2020. Adapun detail lokasi penelitian dapat dilihat pada Gambar 4.1 berikut.



Gambar 4.1 Peta Lokasi Penelitian



Gambar 4.2 Detail Lokasi Penelitian

4.2 Objek Penelitian

Objek pada penelitian ini menggunakan data primer berupa kecepatan dan arus ruas jalan perkotaan yang diperoleh dari survei yang dilakukan di ruas Jalan Siliwangi, Kabupaten Sleman, Daerah Istimewa Yogyakarta. Pengambilan data survei dilakukan menggunakan alat survei berupa perangkat lunak *Android* serta alat survei konvensional.

4.3 Analisis Data

Analisis dan pengolahan data dilakukan dengan cara membandingkan tingkat akurasi hasil survei yang diperoleh dari perangkat lunak *Android* dengan hasil survei secara manual. Selain itu, hasil kecepatan, arus, dan kepadatan ruas jalan yang diperoleh dari masing-masing metode survei akan digunakan untuk menghitung kapasitas dengan model *Greenshields* untuk mengetahui apakah perangkat lunak *Android* yang dikembangkan untuk survei ruas jalan raya ini dapat digunakan dalam bidang ketekniksipilan serta bagaimana hasil perhitungannya bila dibandingkan dengan kapasitas ruas jalan yang diperoleh dari survei secara manual.

4.4 Tahapan Penelitian

Adapun tahapan dalam penelitian ini adalah sebagai berikut.

- a. Tahap perumusan masalah, meliputi perumusan topik penelitian, termasuk perumusan tujuan serta manfaat penelitian.
- b. Tahap studi pustaka untuk menemukan informasi dari berbagai sumber buku, catatan, literatur, hasil laporan penelitian sejenis yang relevan sesuai dengan objek yang sedang diteliti.
- c. Tahap perumusan teori, merupakan pengkajian teori yang melandasi penelitian serta ketentuan-ketentuan yang dijadikan acuan dalam pelaksanaan penelitian.
- d. Tahap pengembangan perangkat lunak, sesuai tata cara pelaksanaan survei kecepatan, arus, dan kepadatan ruas jalan.
- e. Tahap pengumpulan data yaitu data primer yang didapatkan dari survei yang dilakukan pada ruas Jalan Siliwangi, Kabupaten Sleman, Daerah Istimewa Yogyakarta, baik menggunakan alat survei berupa perangkat lunak *Android* maupun dengan alat survei konvensional.

- f. Tahap analisis dan validasi data yang dilakukan untuk menguji kegunaan perangkat lunak *Android* dalam bidang ketekniksipilan, serta mendapatkan nilai perbandingan tingkat akurasi hasil survei kecepatan, arus, dan kepadatan ruas jalan antara perangkat lunak *Android* dengan survei secara manual. Dalam penelitian ini, validasi dilakukan dengan cara menerapkan dua metode uji, yaitu uji t berpasangan dan uji wilcoxon.
- g. Tahap penulisan, pembahasan dan penarikan kesimpulan. Tahap ini meliputi penulisan laporan penelitian berdasarkan aturan yang berlaku, pembahasan yang didapatkan dari hasil validasi perangkat lunak, serta kesimpulan yang diambil berdasarkan teori dan hasil validasi yang digunakan untuk menjawab masalah yang timbul.

4.5 Metode Pengembangan Perangkat Lunak

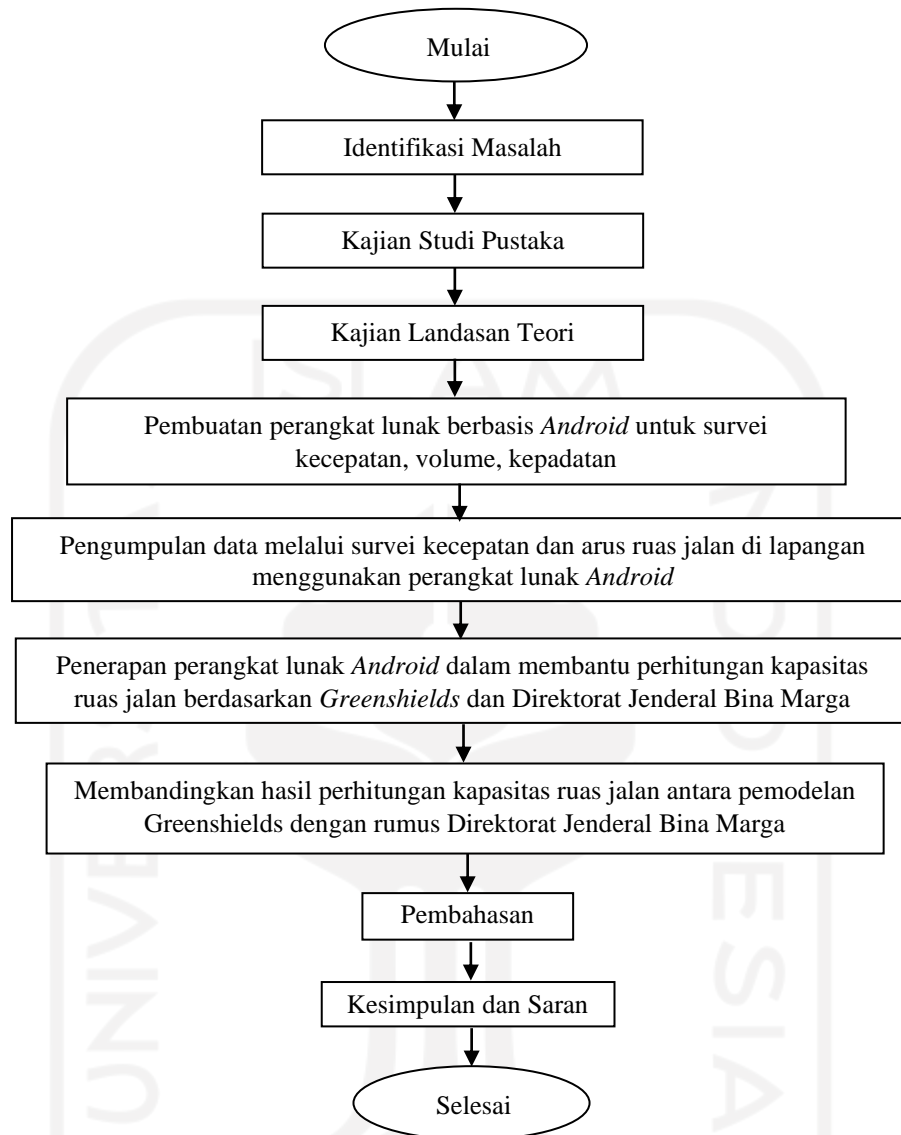
Dalam penelitian ini, pengembangan perangkat lunak dilakukan dengan metode *Waterfall*. Metode ini memiliki tahapan berupa *requirement, design, implementation, integration and testing*, serta *operation and maintenance*. Adapun penjelasan dari masing-masing tahapan tersebut adalah sebagai berikut.

- a. *Requirement*, yaitu menentukan kebutuhan dari sebuah perangkat lunak yang akan dikerjakan. Dalam penelitian ini, perangkat lunak *Android* membutuhkan data berupa waktu tempuh dan jumlah kendaraan di lapangan untuk kemudian dapat diproses menjadi data kecepatan, volume, dan kepadatan ruas jalan.
- b. *Design*, yaitu menentukan desain perangkat lunak sebelum dilakukan proses koding. Hal ini mencakup bahasa pemrograman yang akan digunakan, bagaimana perangkat lunak tersebut bekerja, serta rancangan tampilan perangkat lunak. Dalam penelitian ini, bahasa pemrograman yang digunakan adalah *Flutter*, sedangkan untuk cara kerja dan tampilan perangkat lunak akan dibahas dalam Bab V.
- c. *Implementation*, yaitu tahap di mana rancangan yang sudah dibuat diterjemahkan ke dalam bahasa pemrograman agar menjadi sebuah perangkat lunak. Tahapan ini biasa disebut dengan *coding*.

- d. *Integration and testing*, di mana modul yang sudah dibuat akan digabungkan sebelum akhirnya dilakukan pengujian. Tahapan ini bertujuan untuk mengetahui apakah perangkat lunak sudah sesuai rancangan, baik dari segi tampilan maupun fungsionalitas. Dalam penelitian ini, perangkat lunak diuji dengan cara melakukan survei yang berlokasi di ruas Jalan Siliwangi, Yogyakarta. Survei ini dilakukan dengan dua metode, yaitu metode otomatis dengan menggunakan alat survei berupa perangkat lunak *Android*, serta metode manual dengan menggunakan alat survei konvensional seperti formulir survei, *stopwatch*, dan *hand tally*. Survei dilakukan selama satu jam. Hasil survei perangkat lunak berupa data kecepatan, volume, dan kepadatan akan dibandingkan dengan hasil survei dengan alat konvensional untuk mengetahui tingkat akurasi dari masing-masing metode survei. Untuk mengetahui apakah perangkat lunak ini benar-benar dapat menggantikan metode survei dengan alat konvensional di lapangan, maka dilakukan penelitian lebih lanjut. Hasil survei berupa data kecepatan, volume, dan kepadatan dari masing-masing metode akan digunakan untuk menghitung kapasitas ruas jalan raya dengan model Greenshields. Dengan adanya tahapan ini, maka dapat mengetahui tingkat akurasi perangkat lunak, serta mengurangi kemungkinan adanya *bug* atau kesalahan sebelum diproduksi atau digunakan oleh surveyor di lapangan.
- e. *Operation and maintenance*, yaitu tahapan pemeliharaan perangkat lunak. Meskipun sudah dilakukan pengujian, tidak menutup kemungkinan adanya beberapa permasalahan baru yang muncul pada perangkat lunak ketika digunakan oleh pengguna. Oleh karena itu tahapan ini bertujuan untuk mengurangi atau memperbaiki kesalahan yang ditemukan dalam perangkat lunak setelah diproduksi atau digunakan oleh *user*.

4.6 Bagan Alir Penelitian

Adapun bagan alir penelitian secara rinci dapat dilihat pada Gambar 4.3 berikut ini.



Gambar 4.3 Bagan Alir Penelitian

BAB V

HASIL DAN PEMBAHASAN

5.1 Pengembangan Perangkat Lunak “Survey App”

Dalam penelitian ini, pengembangan perangkat lunak *Survey App* difokuskan untuk gawai dengan sistem operasi *Android*. Hal ini dilakukan karena gawai dengan sistem operasi *Android* lebih banyak dijangkau oleh masyarakat Indonesia bila dibandingkan dengan pengguna sistem operasi lain seperti *iOS*. Bahasa pemrograman yang digunakan adalah *Flutter*. *Flutter* merupakan bahasa pemrograman yang relatif baru, sehingga pengembangan di masa mendatang akan lebih mudah. Perangkat lunak yang dikembangkan dengan bahasa pemrograman ini juga membutuhkan memori yang tidak besar, sehingga menjadikan perangkat lunak tersebut lebih ringan.

Berikut merupakan *sample* bahasa pemrograman *Flutter* yang digunakan dalam pengembangan menu utama pada perangkat lunak “*Survey App*”.

```
checkPermission() async {
  await Permission.storage.status.then((status) async {
    await [Permission.storage].request();
    if (!await Permission.storage.status.isGranted) {
      final snackBar = SnackBar(
        content: Text('Anda harus mengizinkan akses penyimpanan
!'),
        action: SnackBarAction(
          label: 'OK',
          onPressed: () {
            _key.currentState.hideCurrentSnackBar();
          },
        ),
      );
      _key.currentState.showSnackBar(snackBar);
    }
  }).catchError((onError) {
    print('onError : $onError');
  });
}
```

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    key: _key,
    appBar: AppBar(
      title: Text('Survey App'),
    ),
    body: body(),
    bottomNavigationBar: bottomNavigation(),
  );
}

Widget body() {
  switch(_currentFragment) {
    case 0:
      return FragmentSurveySpeed();
    case 1:
      return FragmentSurveyVolume();
    case 2:
      return FragmentHitungKepadatan();
    default:
      return null;
  }
}

Widget bottomNavigation() {
  return BottomNavigationBar(
    currentIndex: _currentFragment,
    type: BottomNavigationBarType.fixed,
    onTap: bottomNavPressed,
    items: bottomList.map((BottomList bottomList) {
      return BottomNavigationBarItem(
        icon: Icon(bottomList.icon),
        title: Text(bottomList.title),
      );
    }).toList()
  );
}

bottomNavPressed(int index) {
  setState(() => _currentFragment = index);
}
```

```

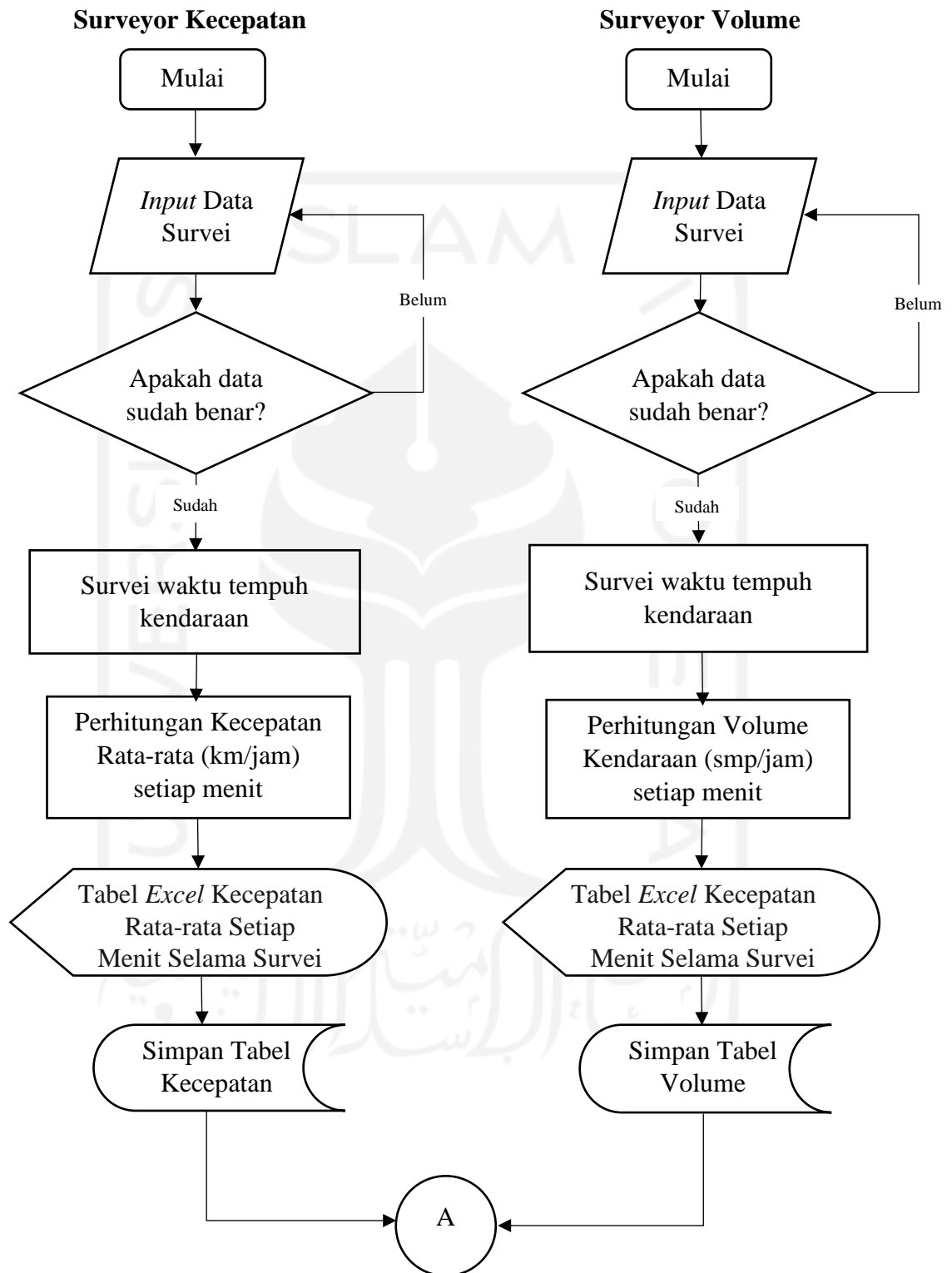
}

class BottomList {
    final int index;
    final String title;
    final IconData icon;
    const BottomList({this.index, this.title, this.icon});
}

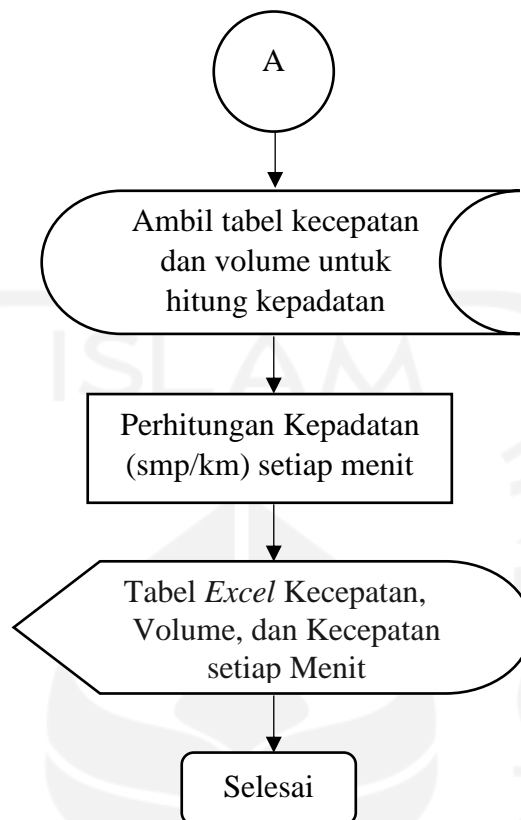
const List<BottomList> bottomList = <BottomList>[
    const BottomList(index: 0, title: 'Survey Speed', icon: Icons.s
hutter_speed),
    const BottomList(index: 1, title: 'Survey Volume', icon: Icons.
departure_board),
    const BottomList(index: 2, title: 'Hitung Kepadatan', icon: Ico
ns.event_note),
];

```

Perangkat lunak ini dirancang untuk digunakan oleh 2 surveyor sekaligus dalam satu waktu. Surveyor pertama bertugas untuk menghitung kecepatan sedangkan surveyor yang lain bertugas untuk menghitung volume. Adapun bagan alir atau *flowchart* mengenai cara kerja perangkat lunak “*Survey App*” adalah sebagai berikut.



Gambar 5.1 Bagan Alir Kinerja "Survey App"



Gambar 5.2 Lanjutan Bagan Alir Kinerja “Survey App”

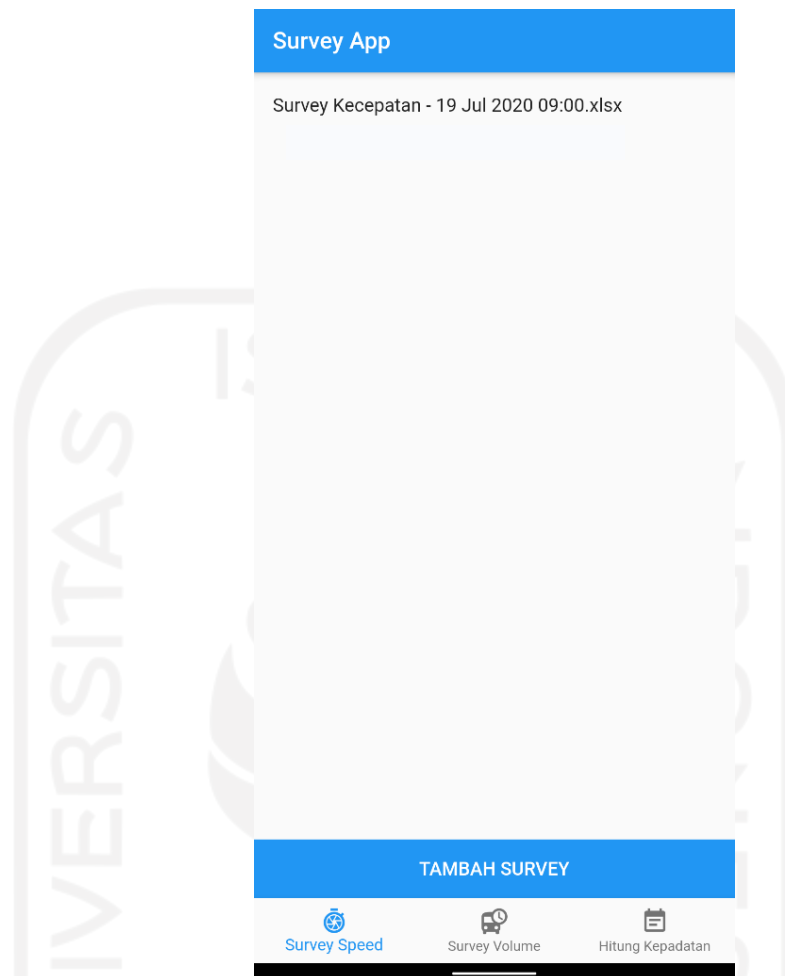
Langkah-langkah dalam pengambilan data kecepatan, volume, dan kepadatan menggunakan perangkat lunak *Survey App* adalah sebagai berikut.

a. Pengambilan Data Kecepatan (*Spot Speed*)

Dalam tahap ini, surveyor yang bertugas untuk mengambil data kecepatan adalah Surveyor 1. Jenis kecepatan yang disurvei adalah kecepatan setempat. Adapun langkah-langkah dalam pengambilan data kecepatan menggunakan perangkat lunak *Survey App* adalah sebagai berikut.

1) Memilih jenis survei yang akan dilakukan

Dalam perangkat lunak *Survey App*, terdapat halaman utama dengan tiga jenis survei yang dapat dipilih pada bagian bawah yaitu survei kecepatan, survei volume, dan perhitungan kepadatan. Tampilan halaman utama dapat dilihat pada Gambar 5.3 berikut ini.



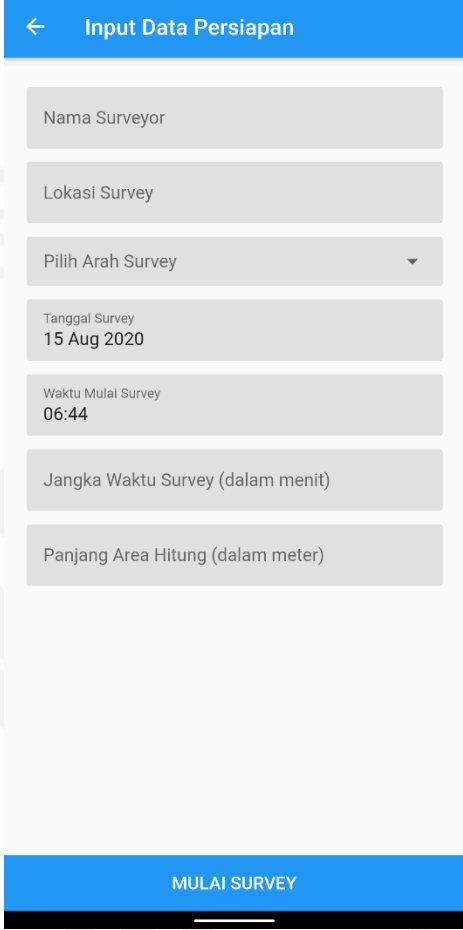
Gambar 5.3 Halaman Utama Survey App

Surveyor 1 bertugas untuk melakukan survei kecepatan, sehingga Surveyor 1 harus memilih tautan “Survei Speed”, kemudian klik “Tambah Survey”.

2) Memasukkan Data yang Diperlukan Untuk Survei

Adapun data yang dibutuhkan dalam survei kecepatan antara lain lokasi survei, arah ruas jalan yang disurvei, panjang area hitung (dalam meter), jangka waktu survei, serta waktu pelaksanaan survei. Setelah seluruh data survei dimasukkan kemudian klik tombol “Mulai Survey”. Kemudian aplikasi akan mengkonfirmasi apakah data yang dimasukkan sudah benar

sebelum memulai survei. Tampilan input data survei kecepatan dapat dilihat pada Gambar 5.4 Berikut.

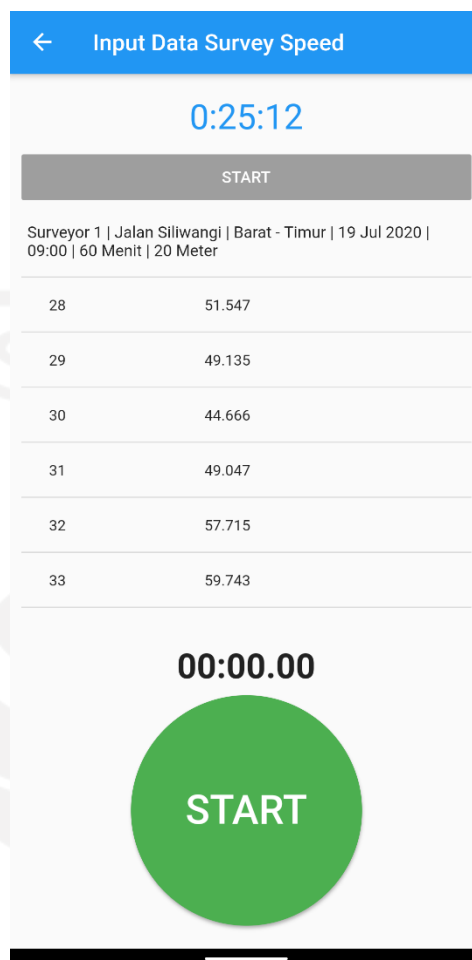


UNIVERSITAS

Gambar 5.4 Input Data Survei Kecepatan

3) Perhitungan Waktu Tempuh Kendaraan Setiap Menit

Surveyor menekan tombol “start” pada bagian atas layar untuk memulai survei sesuai jangka waktu yang ditentukan. Kemudian pada bagian bawah layar juga terdapat tombol “start” berwarna hijau yang digunakan untuk melakukan pengukuran waktu tempuh kendaraan. Tampilan perhitungan waktu tempuh dapat dilihat pada Gambar 5.5 berikut.



Gambar 5.5 Perhitungan Waktu Tempuh Kendaraan

Surveyor menekan tombol “start” pada saat kendaraan memasuki area hitung. Kemudian akan muncul sebuah *stopwatch* yang akan mengukur waktu tempuh kendaraan. Ketika kendaraan tersebut keluar dari area hitung, surveyor menekan tombol “stop”. Perhitungan dilakukan terhadap setiap kendaraan yang lewat pada menit tersebut. Setelah selesai satu menit, aplikasi akan secara otomatis menghitung kecepatan rata-rata pada menit tersebut dalam satuan km/jam kemudian perhitungan akan dilanjutkan pada menit berikutnya. Langkah ini dilakukan sepanjang jangka waktu survei yang telah dimasukkan di awal.

4) Hasil Keluaran Survei Kecepatan

Survei kecepatan dengan menggunakan perangkat lunak “*Survey App*” menghasilkan keluaran berupa tabel kecepatan rata-rata pada setiap menit dalam bentuk *spreadsheet*. Adapun tampilan hasil survei kecepatan menggunakan aplikasi adalah sebagai berikut.

	A	B	C	D
1	Nama Surveyor	Surveyor 1		
2	Lokasi Survey	Jalan Siliwangi		
3	Arah Survey	Barat - Timur		
4	Tanggal & Waktu Survey	19 Jul 2020 09:00		
5	Jangka Waktu	60 Menit		
6	Panjang Area	20 Meter		
7				
8	Menit	Kecepatan Rata-rata(km/jam)		
9	1	53.39		
10	2	53.522		
11	3	53.76		
12	4	54.267		
13	5	53.148		
14	6	54.544		
15	7	54.962		
16	8	49.692		
17	9	49.599		
18	10	47.492		
19	11	45.05		
20	12	50.306		
21	13	50.614		
22	14	49.078		
23	15	52.934		
24	16	41.573		
25	17	47.888		
26	18	58.049		
27	19	48.719		
28	20	53.748		
29	21	44.499		

Gambar 5.6 Output Survei Kecepatan

b. Pengambilan Data Volume

Dalam penelitian ini, surveyor yang bertugas untuk mengambil data volume adalah Surveyor 2. Adapun langkah-langkah dalam pengambilan data volume menggunakan perangkat lunak *Survey App* adalah sebagai berikut.

1) Memilih Jenis Survei yang Akan Dilakukan

Seperti halnya pada survei kecepatan, surveyor perlu memilih jenis survei yang akan dilakukan. Surveyor 2 bertugas untuk melakukan survei volume, sehingga Surveyor 2 harus memilih tautan “Survei Volume”, kemudian klik “Tambah Survey”.

2) Memasukkan Data yang Diperlukan Untuk Survei

Data survei yang diperlukan untuk survei volume sedikit berbeda dengan survei kecepatan. Adapun data yang dibutuhkan dalam survei kecepatan antara lain lokasi survei, arah ruas jalan yang disurvei, jangka waktu survei, waktu pelaksanaan survei serta jenis kendaraan yang akan disurvei. Pada bagian pemilihan jenis kendaraan, terdapat kolom untuk memasukkan nilai ekivalensi mobil penumpang atau emp. Nilai emp dimasukkan oleh surveyor secara manual berdasarkan ketentuan dalam MKJI sesuai dengan jenis ruas jalan yang disurvei. Tampilan input data survei volume dapat dilihat pada Gambar 5.7 Berikut.

The screenshot shows a mobile application interface titled "Input Data Persiapan". It contains several input fields and a list of vehicle types with checkboxes. The fields are: "Nama Surveyor", "Lokasi Survey", "Pilih Arah Survey" (a dropdown menu), "Tanggal Survey" (displaying "15 Aug 2020"), "Waktu Mulai Survey" (displaying "07:42"), and "Jangka Waktu Survey (dalam menit)". Below these fields is a list of vehicle types with checkboxes and "Nilai EMP" input fields:

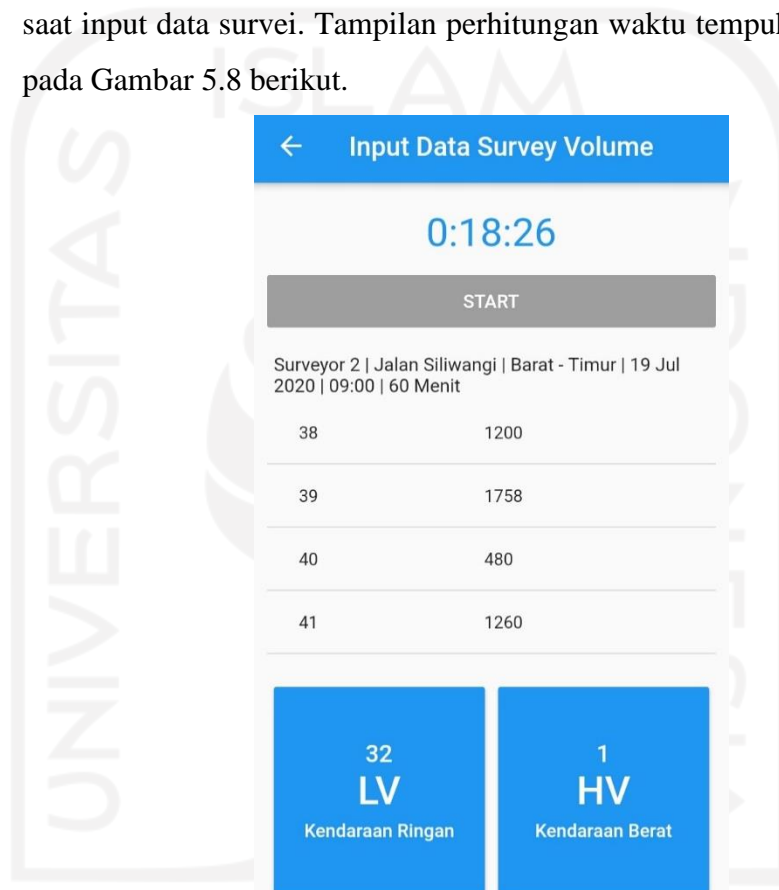
Jenis Kendaraan	Nilai EMP
<input type="checkbox"/> LV - Kendaraan Ringan	Nilai EMP
<input type="checkbox"/> HV - Kendaraan Berat	Nilai EMP
<input type="checkbox"/> MC - Sepeda Motor	Nilai EMP

At the bottom of the screen is a blue button labeled "MULAI SURVEY".

Gambar 5.7 Input Data Survei Volume

3) Perhitungan Jumlah Kendaraan Setiap Menit

Perhitungan jumlah kendaraan akan dimulai setelah surveyor menekan tombol “mulai” pada layar untuk memulai survei sesuai jangka waktu yang ditentukan. Kemudian pada bagian bawah layar terdapat tombol yang sesuai dengan jenis kendaraan yang ditentukan oleh surveyor pada saat input data survei. Tampilan perhitungan waktu tempuh dapat dilihat pada Gambar 5.8 berikut.



Gambar 5.8 Perhitungan Jumlah Kendaraan

Ketika posisi kendaraan persis di depan surveyor, surveyor menekan tombol sesuai jenis kendaraan yang lewat, sehingga pada tombol akan tertera jumlah kendaraan yang melalui ruas jalan tersebut. Perhitungan dilakukan terhadap setiap kendaraan yang lewat pada menit tersebut. Setelah selesai satu menit, aplikasi akan secara otomatis mengkonversi jumlah kendaraan ke dalam satuan mobil penumpang atau smp serta menghitung volume kendaraan pada menit tersebut dalam satuan

smp/jam, kemudian perhitungan akan dilanjutkan pada menit berikutnya. Langkah ini dilakukan sepanjang jangka waktu survei yang telah dimasukkan di awal.

4) Hasil Keluaran Survei Volume

Survei volume dengan menggunakan perangkat lunak “*Survey App*” menghasilkan keluaran berupa tabel volume kendaraan pada setiap menit dalam format Microsoft Excel. Adapun tampilan hasil keluaran survei kecepatan adalah sebagai berikut.

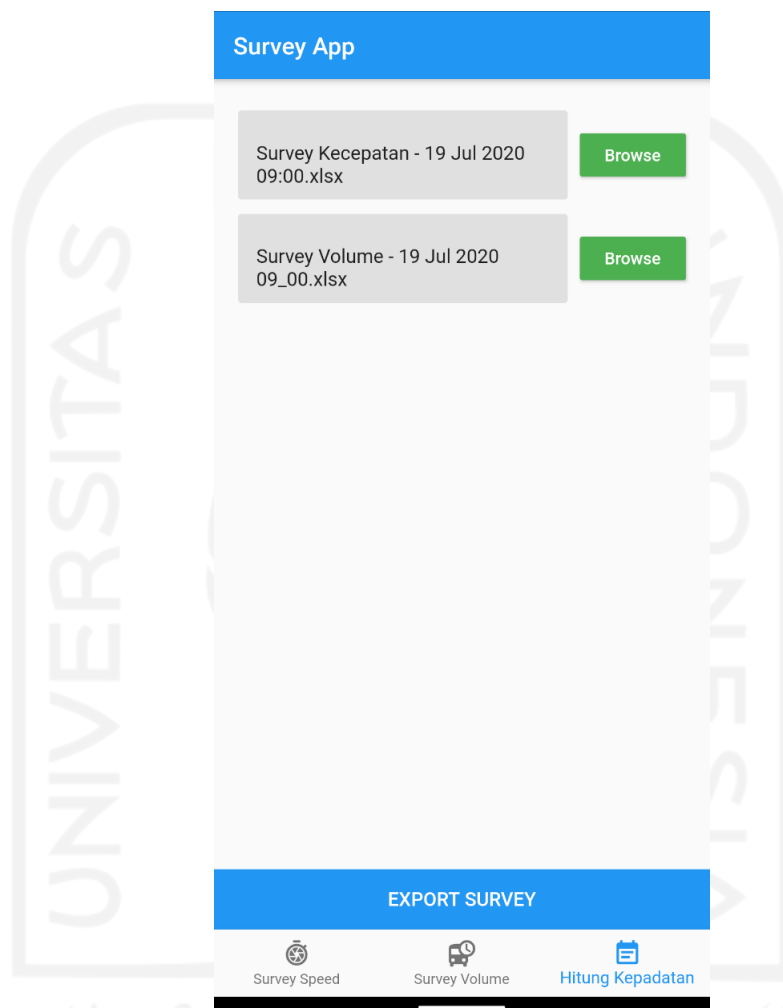
	A	B	C	D
1	Nama Surveyor	Surveyor 2		
2	Lokasi Survey	Jalan Siliwangi		
3	Arah Survey	Barat - Timur		
4	Tanggal & Waktu Survey	19 Jul 2020 09:00		
5	Jangka Waktu	60 Menit		
6				
7	Menit	Volume(smp/jam)		
8	1	660		
9	2	1200		
10	3	2358		
11	4	1020		
12	5	2256		
13	6	780		
14	7	1500		
15	8	2418		
16	9	1620		
17	10	1398		
18	11	1260		
19	12	1620		
20	13	1320		
21	14	660		
22	15	2196		
23	16	1320		
24	17	1440		
25	18	780		
26	19	2100		
27	20	558		
28	21	1278		
29	22	1140		
30	23	720		

Gambar 5.9 Output Survei Volume

c. Perhitungan Kepadatan

Perhitungan kepadatan ruas jalan raya dilakukan dengan cara menggabungkan *output* dari kedua jenis survei ke dalam salah satu telepon seluler *Android*, kemudian dimasukkan ke dalam aplikasi “*Survey App*”. Cara memasukkan kedua *output* survei ke dalam aplikasi adalah dengan menggunakan fitur

“*browse*” yang terdapat pada tautan “perhitungan kepadatan” di halaman utama. Adapun tampilan tautan perhitungan kepadatan dapat dilihat pada Gambar 5.10 berikut ini.



Gambar 5.10 Perhitungan Kepadatan

Setelah kedua *output* survei dimasukkan ke dalam aplikasi, surveyor menekan tombol “export survey”, sehingga aplikasi akan secara otomatis menghitung kepadatan serta menyimpan hasil keluarannya dalam format Microsoft Excel. Hasil keluaran dari perhitungan kepadatan adalah sekaligus data keluaran dari keseluruhan aplikasi “*Survey App*”. Adapun hasil keluaran aplikasi “*Survey App*” terdiri atas nilai kecepatan, volume, dan kepadatan setiap menit, yang dapat dilihat pada Gambar 5.11 berikut ini.

	A	B	C	D
1	Nama Surveyor	Surveyor 1 - Surveyor 2		
2	Lokasi Survey	Jalan Siliwangi		
3	Arah Survey	Barat - Timur		
4	Tanggal & Waktu Survey	19 Jul 2020 09:00		
5	Jangka Waktu	60 Menit		
6	Panjang Area	20 Meter		
7				
8	Menit	Kecepatan (km/jam)	Volume (smp/jam)	Kepadatan (smp/km)
9	1	53.39	660	12.361
10	2	53.522	1200	22.42
11	3	53.76	2358	43.861
12	4	54.267	1020	18.795
13	5	53.148	2256	42.447
14	6	54.544	780	14.3
15	7	54.962	1500	27.291
16	8	49.692	2418	48.659
17	9	49.599	1620	32.661
18	10	47.492	1398	29.436
19	11	45.05	1260	27.968
20	12	50.306	1620	32.202
21	13	50.614	1320	26.079
22	14	49.078	660	13.447
23	15	52.934	2196	41.485
24	16	41.573	1320	31.751
25	17	47.888	1440	30.07
26	18	58.049	780	13.436
27	19	48.719	2100	43.104
28	20	53.748	558	10.381
29	21	44.499	1278	28.719
30	22	47.511	1140	23.994
31	23	52.679	720	13.667
32	24	43.379	1818	41.909
33	25	57.594	936	16.251
34	26	50.417	2040	40.462

Gambar 5.11 Output Perangkat Lunak “Survey App”

5.2 Pengambilan Data Survei

Data diperoleh dari survei yang dilakukan di ruas jalan raya. Survei dilakukan menggunakan perangkat lunak *Android “Survey App”*. Adapun data yang diperoleh dari survei menggunakan perangkat lunak ini adalah berupa kecepatan, volume, dan kepadatan kendaraan setiap menit dalam format *spreadsheet*.

Survei dilakukan selama 1 jam (60 menit) pada ruas Jalan Siliwangi, Daerah Istimewa Yogyakarta, yang merupakan jalan antar kota. Dalam penelitian ini, area perhitungan waktu tempuh kendaraan ditetapkan sepanjang 20 meter. Kendaraan yang disurvei adalah kendaraan ringan (LV) dan kendaraan berat (HV). Survei dimulai pada pukul 09.00 WIB. Adapun hasil survei menggunakan perangkat lunak *Survey App* dapat dilihat pada Tabel 5.1 berikut ini.

Tabel 5.1 Hasil Perhitungan Kecepatan, Volume, dan Kepadatan Menggunakan Perangkat Lunak “*Survey App*”

Menit	Volume (smp/menit)	Kecepatan (km/menit)	Kepadatan (smp/km)
1	18,6	0,90	20,64
2	15,6	0,88	17,66
3	15	0,88	17,05
4	22,3	0,86	25,80
5	13,3	0,79	16,89
6	21,3	0,74	28,75
7	8,3	0,77	10,84
8	13,3	0,75	17,84
9	20,3	0,87	23,30
10	19	0,73	25,99
11	10	0,95	10,52
12	32,9	0,75	43,83
13	14	0,92	15,26
14	16,6	0,79	21,04
15	17,6	0,85	20,72
16	12	0,87	13,72
17	34,6	0,74	46,80
18	10	0,89	11,30
19	13	0,83	15,60
20	25,2	0,83	30,44
21	8	0,83	9,60

**Lanjutan Tabel 5.1 Hasil Perhitungan Kecepatan, Volume, dan Kepadatan
Menggunakan Perangkat Lunak “Survey App”**

Menit	Volume (smp/menit)	Kecepatan (km/menit)	Kepadatan (smp/km)
22	14	0,85	16,57
23	29	0,80	36,20
24	12	0,74	16,29
25	16,3	0,88	18,62
26	30,3	0,77	39,60
27	16,3	0,74	21,95
28	12,6	0,94	13,40
29	22,3	0,80	27,74
30	15	0,86	17,44
31	30,3	0,87	34,64
32	18	0,87	20,60
33	40	0,75	53,26
34	10	0,78	12,80
35	6	0,78	7,72
36	20,3	0,90	22,57
37	33,9	0,79	43,12
38	12,3	0,97	12,73
39	12	0,79	15,29
40	17,9	0,81	22,07
41	17,3	0,84	20,54
42	11	0,81	13,52
43	33	0,83	39,70
44	16	0,75	21,30
45	15,3	0,87	17,57
46	26,1	0,72	36,15
47	15	0,78	19,28
48	10,6	0,80	13,24
49	22,9	0,85	26,94
50	6	0,76	7,94
51	10,9	0,64	17,11
52	9	0,80	11,20
53	23,6	0,70	33,76
54	7	0,80	8,79
55	8	0,76	10,47

Lanjutan Tabel 5.1 Hasil Perhitungan Kecepatan, Volume, dan Kepadatan Menggunakan Perangkat Lunak “Survey App”

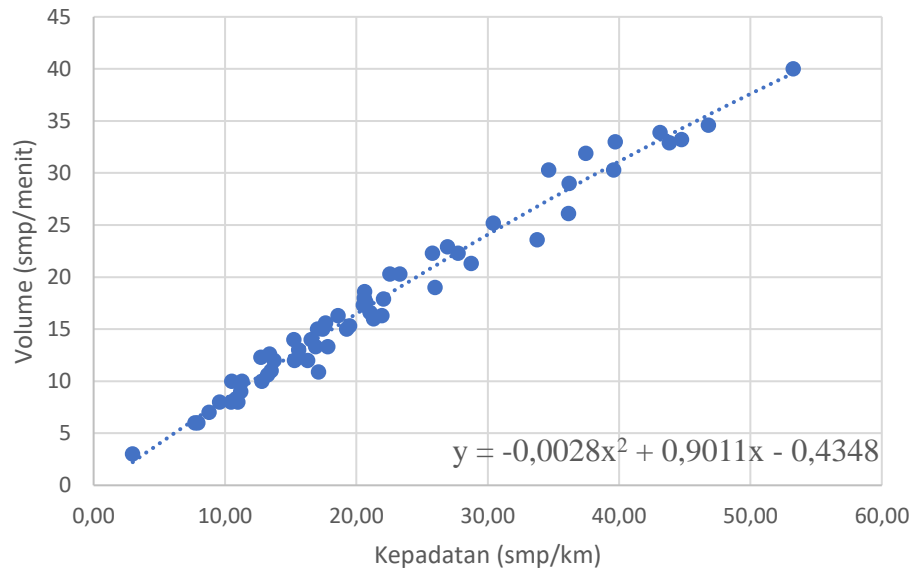
Menit	Volume (smp/menit)	Kecepatan (km/menit)	Kepadatan (smp/km)
56	31,9	0,85	37,46
57	15,3	0,79	19,48
58	8	0,73	10,98
59	33,2	0,74	44,76
60	3	1,01	2,97

5.3 Penerapan Perangkat Lunak “Survey App” dalam Membantu Perhitungan Kapasitas Ruas Jalan

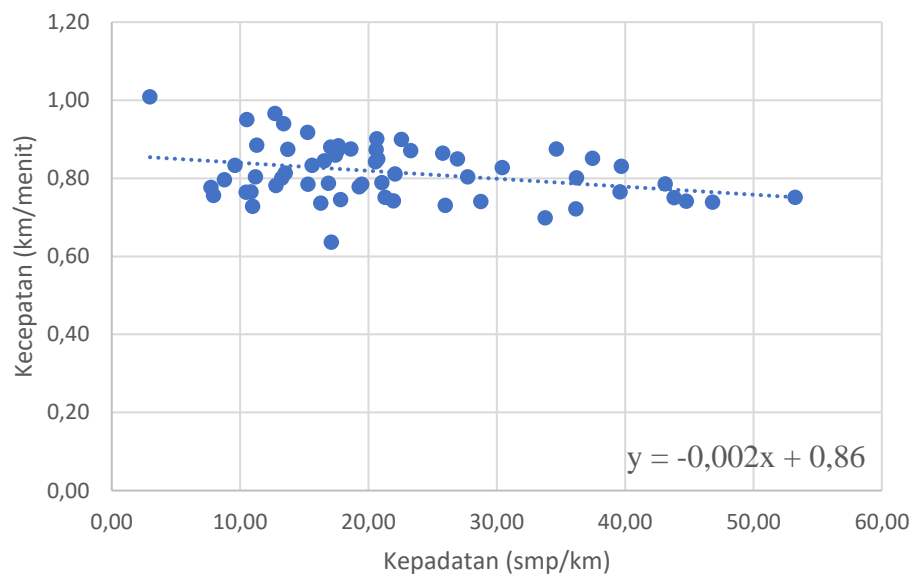
Salah satu manfaat dalam penggunaan perangkat lunak *Android “Survey App”* adalah hasil keluaran yang dapat digunakan untuk melakukan perhitungan kinerja ruas jalan, salah satunya adalah perhitungan kapasitas ruas jalan. Subbab ini membahas mengenai perhitungan kapasitas ruas jalan yang dilakukan dengan menggunakan data hasil keluaran perangkat lunak *Android “Survey App”*. Perhitungan kapasitas dilakukan dengan pemodelan *Greenshields* yang kemudian dibandingkan dengan perhitungan kapasitas ruas jalan dengan metode dari Direktorat Jenderal Bina Marga.

1. Perhitungan Kapasitas Ruas Jalan Raya Berdasarkan Pemodelan *Greenshields*

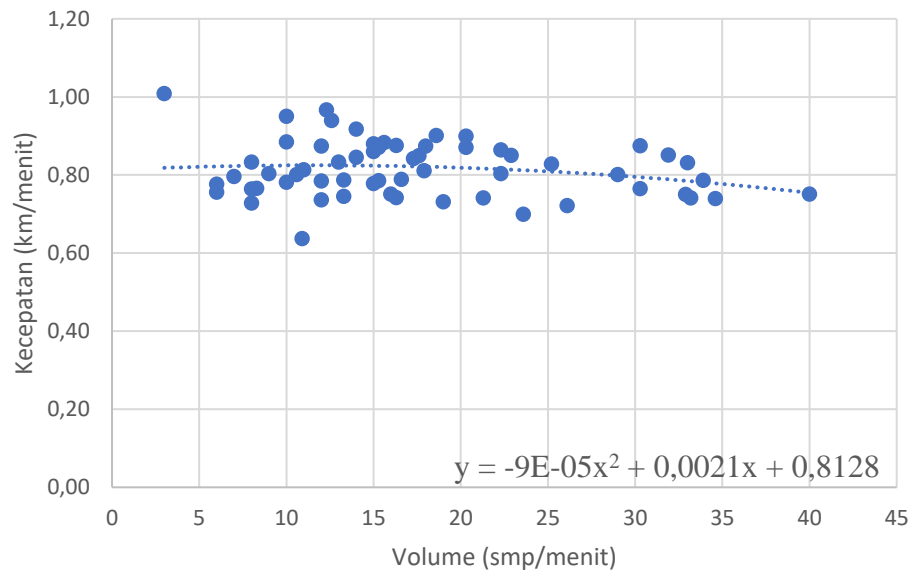
Hasil survei yang diperoleh dari perangkat lunak *Android “Survey App”* dapat dilihat pada Tabel 5.1 di atas. Dari Tabel 5.1 dapat diperoleh diagram pencar dan persamaan *trendline* dari hubungan antara kepadatan dengan volume, kepadatan dengan kecepatan, serta volume dengan kecepatan. Adapun diagram pencar serta persamaannya adalah sebagai berikut ini.



Gambar 5.12 Diagram Pencar dan Persamaan Orde 2 dari Hubungan antara Kepadatan dengan Volume dari Survei dengan “Survey App”



Gambar 5.13 Diagram Pencar dan Persamaan Linear dari Hubungan antara Kepadatan dengan Kecepatan dari Survei dengan “Survey App”



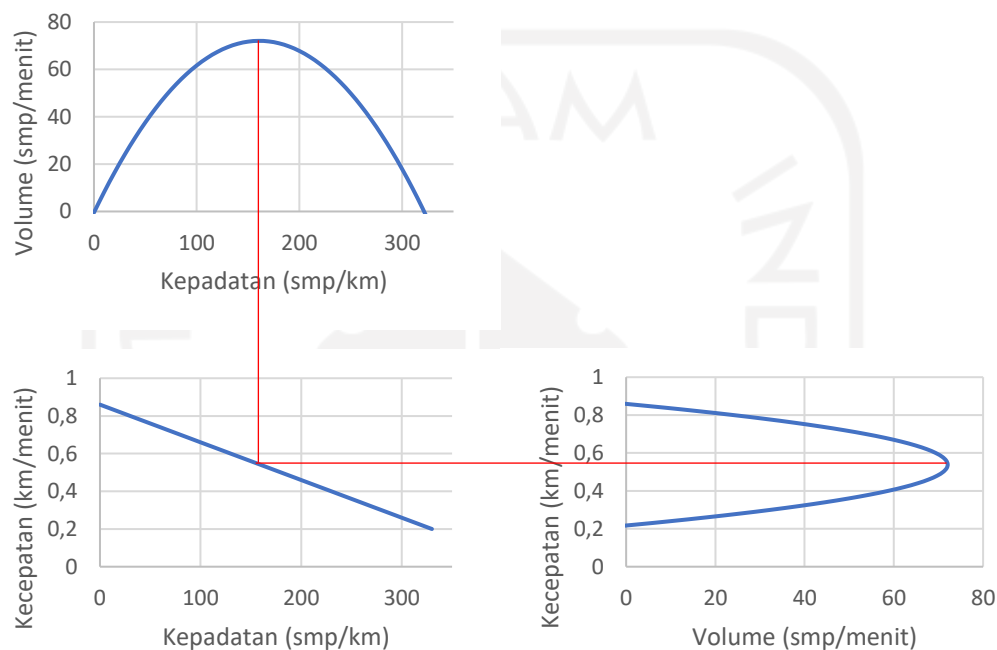
Gambar 5.14 Diagram Pencar dan Persamaan Orde 2 dari Hubungan antara Volume dengan Kecepatan dari Survei dengan “*Survey App*”

Dari ketiga grafik di atas dapat dilakukan kalibrasi data dengan cara mencari nilai volume berdasarkan persamaan yang diperoleh dari masing-masing grafik. Data kecepatan, volume, dan kepadatan yang sudah dikalibrasi tersebut yang akan membentuk grafik kinerja ruas jalan raya berdasarkan metode Greenshields. Untuk perhitungan volume digunakan persamaan yang diperoleh pada Gambar 5.13, sedangkan untuk perhitungan kecepatan digunakan persamaan yang diperoleh pada Gambar 5.14. Masing-masing perhitungan dimulai dari $y = 0$. Adapun hasil kalibrasi data dapat dilihat pada Tabel 5.5 berikut ini.

Tabel 5.2 Kalibrasi Data Greenshields untuk survei dengan “Survey App”

Kepadatan (smp/km)	Volume (smp/menit)	Kecepatan (km/menit)
10	8,2962	0,84
20	16,4672	0,82
30	24,0782	0,8
40	31,1292	0,78
50	37,6202	0,76
60	43,5512	0,74
70	48,9222	0,72
80	53,7332	0,7
90	57,9842	0,68
100	61,6752	0,66
110	64,8062	0,64
120	67,3772	0,62
130	69,3882	0,6
140	70,8392	0,58
150	71,7302	0,56
160	72,0612	0,54
170	71,8322	0,52
180	71,0432	0,5
190	69,6942	0,48
200	67,7852	0,46
210	65,3162	0,44
220	62,2872	0,42
230	58,6982	0,4
240	54,5492	0,38
250	49,8402	0,36
260	44,5712	0,34
270	38,7422	0,32
280	32,3532	0,3
290	25,4042	0,28
300	17,8952	0,26
310	9,8262	0,24
320	1,1972	0,22

Dari Tabel 5.2 dapat diperoleh grafik *Greenshields* hubungan antara kecepatan, volume, dan kepadatan pada ruas Jalan Siliwangi, Daerah Istimewa Yogyakarta dengan data survei yang diperoleh dari perangkat lunak *Android "Survey App"*. Adapun grafik tersebut dapat dilihat pada Gambar 5.15 berikut ini.



Gambar 5.15 Hubungan Antara Kecepatan, Volume, dan Kepadatan dengan Model *Greenshields* dari Survei dengan “*Survey App*”

Kapasitas ruas jalan raya merupakan volume maksimum dari kendaraan yang mampu melalui ruas jalan tersebut. Dari Gambar 5.15 dan Tabel 5.2 dapat dilihat bahwa kapasitas ruas Jalan Siliwangi, Daerah Istimewa Yogyakarta dengan metode *Greenshields* menggunakan data yang diambil dari survei menggunakan perangkat lunak *Android "Survey App"* adalah sebesar 72,0612 smp/menit, atau sebesar 4323,672 smp/jam.

2. Perhitungan Kapasitas Ruas Jalan Berdasarkan Metode dari Direktorat Jenderal Bina Marga

Jalan Siliwangi merupakan jalan antar kota yang memiliki 4 (empat) lajur, distribusi arah 50% - 50%, hambatan samping sangat rendah, serta lebar per lajur sebesar 3,5 meter. Adapun perhitungan kapasitas ruas jalan dengan metode yang

ditetapkan oleh Direktorat Jenderal Bina Marga dilakukan menggunakan persamaan berikut ini.

$$C = C_0 \times FC_W \times FC_{SP} \times FC_{SF}$$

Dengan:

C = Kapasitas Ruas Jalan (smp/jam)

C_0 = Kapasitas Dasar (smp/jam)

FC_W = Faktor Pengaruh Lebar Lajur

FC_{SP} = Faktor Pengaruh Distribusi Arah

FC_{SF} = Faktor Pengaruh Hambatan Samping

Sehingga dari persamaan dan kriteria ruas jalan tersebut dapat diperoleh nilai kapasitas sebagai berikut ini.

$$\begin{aligned} C &= (2 \times 1900) \times 1 \times 1 \times 0,99 \\ &= 3762 \text{ smp/jam} \end{aligned}$$

Rekapitulasi hasil perhitungan kapasitas dapat dilihat pada Tabel 5.6 berikut ini.

Tabel 5.6 Rekapitulasi Hasil Perhitungan Kapasitas

	Pemodelan <i>Greenshields</i>	Persamaan MKJI 1997	Selisih (%)
Kapasitas (smp/jam)	4323,672	3762	14,93

Dari perhitungan di atas diperoleh bahwa terdapat perbedaan sebesar 14,93% antara perhitungan kapasitas menggunakan pemodelan *Greenshields* dengan perhitungan kapasitas menggunakan rumus yang ditetapkan oleh Direktorat Jenderal Bina Marga. Hal ini disebabkan karena survei di lapangan hanya dilakukan selama 60 menit untuk keperluan pengujian perangkat lunak, sehingga belum merepresentasikan kinerja ruas secara keseluruhan.

5.4 Kelebihan dan Kekurangan Perangkat Lunak *Android* “*Survey App*”

Pada saat ini, perangkat lunak *Android* “*Survey App*” masih dalam tahap purwarupa atau *prototype*, sehingga memiliki kelebihan dan kekurangan. Adapun kelebihan dari perangkat lunak “*Survey App*” adalah sebagai berikut.

- a. Lebih mudah. Surveyor tidak perlu membawa banyak peranti survei seperti form survei, *stopwatch* serta *hand tallies* ke lapangan. Surveyor hanya perlu telepon seluler berbasis *Android*, pengukur jarak, serta penanda.
- b. Tampilan yang sederhana. Meskipun perangkat ini menggabungkan fungsi *stopwatch*, *hand tallies*, dan form survei menjadi satu, surveyor hanya perlu memasukkan data yang diperlukan dan tidak perlu melakukan perhitungan yang mendalam karena sebagian besar perhitungan dilakukan secara otomatis oleh perangkat lunak.
- c. Waktu survei yang singkat. Surveyor hanya perlu waktu selama 65-70 menit dari mulai survei hingga mendapatkan hasil akhir berupa tabel volume, kecepatan, dan kepadatan. Sedangkan survei dengan alat konvensional dapat memakan waktu hingga 90-120 menit karena surveyor harus menyalin data di lapangan yang terdapat dalam form survei ke dalam sebuah *spreadsheet* seperti *Microsoft Excel* kemudian menghitung kecepatan, volume, dan kepadatan secara manual hingga mencapai hasil akhir berupa tabel kecepatan, volume, dan kepadatan.

Adapun kekurangan dari perangkat lunak “*Survey App*” adalah sebagai berikut.

- a. Hasil keluaran perangkat lunak berupa *spreadsheet* hanya mencakup kecepatan rata-rata dan volume kendaraan setiap menit, belum mencakup secara keseluruhan waktu tempuh setiap kendaraan serta jumlah kendaraan pada menit tersebut. Hal ini disebabkan karena *bug* yang dialami pada saat proses pengembangan, sehingga perlu dilakukan proses perbaikan dan pembaruan lebih lanjut.
- b. Adanya risiko “salah tekan”, terutama pada survei volume. Surveyor pada umumnya cenderung menghitung jumlah kendaraan tanpa melihat peranti

survei. Perangkat lunak “*Survey App*” sudah dilengkapi dengan fitur getar sebagai *feedback* pada saat tombol-tombol pada perangkat lunak “*Survey App*” ditekan untuk meminimalisasi kesalahan dalam memasukkan data. Namun perlu adanya ketelitian dari surveyor serta kondisi telepon seluler yang prima agar tidak ada kesalahan saat memasukkan data.

- c. Hanya tersedia untuk telepon seluler berbasis *Android*. Hingga saat ini, aplikasi ini belum tersedia untuk telepon seluler dengan sistem operasi selain *Android* seperti *iOS* dan sejenisnya.



BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Setelah melakukan analisis terhadap pengembangan perangkat lunak *Android* untuk perhitungan kecepatan, volume, dan kepadatan, maka diperoleh kesimpulan sebagai berikut.

1. Sebuah perangkat lunak dikembangkan untuk gawai dengan sistem operasi *Android* untuk membantu surveyor dalam menghitung volume, kecepatan, dan kepadatan di ruas jalan. Perangkat lunak dikembangkan dengan bahasa pemrograman *Flutter* karena memori yang digunakan lebih rendah serta bahasa yang tergolong baru sehingga lebih mudah untuk pengembangan di masa mendatang.
2. Hasil keluaran perangkat lunak *Survey App* dapat digunakan untuk membantu perhitungan kapasitas ruas berdasarkan pemodelan *Greenshields*. Namun terdapat perbedaan antara hasil perhitungan kapasitas ruas jalan berdasarkan model *Greenshields* dengan hasil kapasitas yang dihitung menggunakan rumus dari Direktorat Jenderal Bina Marga. Hasil kapasitas berdasarkan pemodelan *Greenshields* adalah sebesar 4323,672 smp/jam, sedangkan kapasitas yang diperoleh dari rumus yang ditetapkan oleh Direktorat Jenderal Bina Marga adalah sebesar 3762 smp/jam. Hal ini disebabkan karena survei hanya dilakukan selama 60 menit untuk keperluan pengujian perangkat lunak.

6.2 Saran

Setelah melakukan analisis terhadap pembuatan perangkat lunak *Android* untuk perhitungan kecepatan, volume, dan kepadatan, terdapat beberapa saran yang dapat diajukan untuk penelitian selanjutnya.

1. Perangkat lunak "*Survey App*" direkomendasikan untuk digunakan dalam membantu perhitungan kapasitas ruas jalan karena *output* perangkat lunak ini berupa data kecepatan, volume, dan kepadatan kendaraan di lapangan. Hasil

validasi juga membuktikan bahwa perangkat lunak ini layak digunakan karena prinsip kerja perangkat lunak yang sama seperti *stopwatch* dan *hand tally*.

2. Penelitian ini dimaksudkan untuk memberikan ide mengenai sebuah perangkat lunak yang berpotensi untuk dikembangkan dan digunakan oleh banyak pengguna, sehingga perangkat lunak dalam penelitian ini masih dalam tahap *prototype*. Untuk penelitian di masa mendatang, perlu adanya pengembangan perangkat lunak "*Survey App*" dari segi penggunaan seperti penambahan fitur untuk melakukan perhitungan kecepatan persentil dan kapasitas ruas jalan, maupun fitur untuk melakukan survei pada simpang. Perbaikan terhadap beberapa *bug* agar menjadi sebuah perangkat lunak yang lebih stabil juga perlu dilakukan. Namun yang terpenting adalah adanya pengembangan perangkat lunak untuk gawai dengan sistem operasi iOS, sehingga mencakup lebih banyak pengguna dan dapat diunduh melalui *Google Play Store* maupun *Apple App Store*.

DAFTAR PUSTAKA

- Bellucci, P. dan Cipriani, E. 2010. Data Accuracy on Automatic Traffic Counting: The SMART Project Results. *European Transport Research Review*, Vol.2:175-187. Roma.
- Date, C. J. 2005. *Pengenalan Sistem Basisdata Jilid 2*. Indeks. Jakarta.
- Daulay, M. S. 2007. *Mengenal Hardware-Software dan Pengelolaan Instalasi Komputer*. ANDI. Yogyakarta.
- Direktorat Jenderal Bina Marga. 1997. *Manual Kapasitas Jalan Indonesia (MKJI)*. Departemen Pekerjaan Umum. Jakarta.
- Everest, G. C. 2005. *Database Management*. MCGrawHill. Minnesota.
- Hobbs, F. D. 1979. *Traffic Planning and Engineering*. Pergamon Press. Oxford.
- Hakim, L. dan Musalini, U. 2004. *Cara Mudah Memadukan Web Design dan Web Programming Jilid I*. Elex Media Komputindo. Jakarta.
- Herlinah dan K. H., Musliadi. 2019. *Pemrograman Aplikasi Android dengan Android Studio, Photoshop, dan Audition*. Elex Media Komputindo. Jakarta.
- Hewitt, P. G. 2006. *Conceptual Physics 10th Edition*. Addison-Wesley. Boston
- Pal'o, J. et al. 2019. The Comparison of Automatic Traffic Counting and Manual Traffic Counting. *IOP Conference Series: Materials Science and Engineering*. Vol. 710: 012041. Zilina.
- Pressman, R. S. 2002. *Rekayasa Perangkat Lunak*. Penerbit Andi. Yogyakarta.
- Putranto, L. S. 2019. *Rekayasa Lalu Lintas Edisi 3*. Indeks. Jakarta.
- Romadhona, P. J. dan Chasanah, F. 2018. Prifacounter: The Innovation of Traffic Counter Application for Android OS. *International Journal of Engineering & Technology*, Vol.7:97-101. Yogyakarta.
- Romadhona, P. J. dan Chasanah, F. 2018. Prifacounter Validation: Technology Development Innovation Of Traffic Counter Application. *International Journal of Civil Engineering & Technology*, Vol.9 No.11. Yogyakarta.
- Sunyoto, A. 2007. *AJAX Membangun Web dengan Teknologi Asynchrone Javascript & XML*. Penerbit Andi. Yogyakarta.
- Yulianto, D. dan Yuliansyah, H. 2015. Rancang Bangun Aplikasi Traffic Counter RFID. *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI) Jurusan Teknik Elektro dan Teknik Informasi Universitas Gadjah Mada*, Vol 4 No. 1. Yogyakarta.



LAMPIRAN

Lampiran I Hasil Survei Kecepatan Menggunakan Peralatan Konvensional

Menit	Waktu Tempuh (s)	Kecepatan (m/s)	Kecepatan (km/jam)	Kecepatan Rata-rata (km/jam)
1	1.18	16.94915	61.02	54.62
	0.94	21.2766	76.60	
	1.87	10.69519	38.50	
	1.13	17.69912	63.72	
	1.36	14.70588	52.94	
	1.33	15.03759	54.14	
	1.45	13.7931	49.66	
	1.57	12.73885	45.86	
	1.44	13.88889	50.00	
	1.34	14.92537	53.73	
2	1.38	14.49275	52.17	53.02
	1.36	14.70588	52.94	
	1.33	15.03759	54.14	
	1.76	11.36364	40.91	
	1.57	12.73885	45.86	
	1.07	18.69159	67.29	
	1.18	16.94915	61.02	
	1.68	11.90476	42.86	
	1.24	16.12903	58.06	
	1.31	15.26718	54.96	
3	1.56	12.82051	46.15	52.50
	1.41	14.1844	51.06	
	1.38	14.49275	52.17	
	1.41	14.1844	51.06	
	1.16	17.24138	62.07	

Menit	Waktu Tempuh (s)	Kecepatan (m/s)	Kecepatan (km/jam)	Kecepatan Rata-rata (km/jam)
4	1.16	17.24138	62.07	52.62
	1.33	15.03759	54.14	
	1.2	16.66667	60.00	
	1.38	14.49275	52.17	
	1.67	11.97605	43.11	
	1.89	10.58201	38.10	
	1.2	16.66667	60.00	
	0.94	21.2766	76.60	
	1.44	13.88889	50.00	
	2.4	8.333333	30.00	
5	1.85	10.81081	38.92	49.55
	1.53	13.0719	47.06	
	1.11	18.01802	64.86	
	1.33	15.03759	54.14	
	1.71	11.69591	42.11	
	1.39	14.38849	51.80	
	1.5	13.33333	48.00	
6	2.11	9.478673	34.12	46.77
	1.41	14.1844	51.06	
	1.5	13.33333	48.00	
	1.42	14.08451	50.70	
	1.24	16.12903	58.06	
	1.16	17.24138	62.07	
	2.01	9.950249	35.82	
	2.1	9.52381	34.29	

Menit	Waktu Tempuh (s)	Kecepatan (m/s)	Kecepatan (km/jam)	Kecepatan Rata-rata (km/jam)
7	1.29	15.50388	55.81	47.03
	1.8	11.11111	40.00	
	1.78	11.23596	40.45	
	1.44	13.88889	50.00	
	1.64	12.19512	43.90	
	1.26	15.87302	57.14	
	1.76	11.36364	40.91	
	1.5	13.33333	48.00	
8	1.2	16.66667	60.00	49.09
	1.03	19.41748	69.90	
	1.18	16.94915	61.02	
	1.87	10.69519	38.50	
	2.48	8.064516	29.03	
	1.29	15.50388	55.81	
	2.45	8.163265	29.39	
9	1.06	18.86792	67.92	59.93
	0.89	22.47191	80.90	
	1.76	11.36364	40.91	
	1.19	16.80672	60.50	
	1.03	19.41748	69.90	
	1.36	14.70588	52.94	
	1.13	17.69912	63.72	
	1.09	18.34862	66.06	
	1.6	12.5	45.00	
	1.4	14.28571	51.43	

Menit	Waktu Tempuh (s)	Kecepatan (m/s)	Kecepatan (km/jam)	Kecepatan Rata-rata (km/jam)
10	1.95	10.2564	36.92	45.39
	1.8	11.1111	40.00	
	1.13	17.6991	63.72	
	1.67	11.976	43.11	
	1.6	12.5	45.00	
	1.53	13.0719	47.06	
	1.36	14.7059	52.94	
	1.5	13.3333	48.00	
	1.94	10.3093	37.11	
	1.8	11.1111	40.00	
11	1.07	18.6916	67.29	59.76
	1.18	16.9492	61.02	
	0.9	22.2222	80.00	
	1.31	15.2672	54.96	
	1.22	16.3934	59.02	
	1.16	17.2414	62.07	
	1.26	15.873	57.14	
	1.42	14.0845	50.70	
	1.57	12.7389	45.86	
	1.21	16.5289	59.50	
12	1.53	13.0719	47.06	46.52
	1.47	13.6054	48.98	
	2.12	9.43396	33.96	
	1.53	13.0719	47.06	
	1.44	13.8889	50.00	
	1.6	12.5	45.00	
	1.41	14.1844	51.06	
	1.47	13.6054	48.98	
	1.44	13.8889	50.00	
1.67	11.976	43.11		

Menit	Waktu Tempuh (s)	Kecepatan (m/s)	Kecepatan (km/jam)	Kecepatan Rata-rata (km/jam)
13	1.85	10.8108	38.92	59.93
	1.16	17.2414	62.07	
	1.1	18.1818	65.45	
	1.44	13.8889	50.00	
	1.12	17.8571	64.29	
	0.92	21.7391	78.26	
	1.22	16.3934	59.02	
	1.67	11.976	43.11	
	0.91	21.978	79.12	
	1.22	16.3934	59.02	
14	1.41	14.1844	51.06	49.99
	1.53	13.0719	47.06	
	1.26	15.873	57.14	
	1.24	16.129	58.06	
	1.53	13.0719	47.06	
	1.31	15.2672	54.96	
	1.41	14.1844	51.06	
	1.38	14.4928	52.17	
	1.35	14.8148	53.33	
	2.57	7.7821	28.02	
15	1.31	15.2672	54.96	51.08
	1.28	15.625	56.25	
	1.39	14.3885	51.80	
	1.33	15.0376	54.14	
	1.67	11.976	43.11	
	1.47	13.6054	48.98	
	1.41	14.1844	51.06	
	1.46	13.6986	49.32	
	1.38	14.4928	52.17	
	1.47	13.6054	48.98	

Menit	Waktu Tempuh (s)	Kecepatan (m/s)	Kecepatan (km/jam)	Kecepatan Rata-rata (km/jam)
16	1.2	16.6667	60.00	55.51
	1.31	15.2672	54.96	
	1.33	15.0376	54.14	
	1.06	18.8679	67.92	
	1.26	15.873	57.14	
	1.89	10.582	38.10	
	1.33	15.0376	54.14	
	1.18	16.9492	61.02	
	1.38	14.4928	52.17	
17	2.11	9.47867	34.12	45.48
	1.97	10.1523	36.55	
	1.26	15.873	57.14	
	1.23	16.2602	58.54	
	1.38	14.4928	52.17	
	1.6	12.5	45.00	
	1.41	14.1844	51.06	
	1.85	10.8108	38.92	
	2.3	8.69565	31.30	
	1.44	13.8889	50.00	
18	1.07	18.6916	67.29	57.33
	0.85	23.5294	84.71	
	1.8	11.1111	40.00	
	1.44	13.8889	50.00	
	1.33	15.0376	54.14	
	1.31	15.2672	54.96	
	0.99	20.202	72.73	
	1.57	12.7389	45.86	
	1.37	14.5985	52.55	
1.41	14.1844	51.06		

Menit	Waktu Tempuh (s)	Kecepatan (m/s)	Kecepatan (km/jam)	Kecepatan Rata-rata (km/jam)
19	1.41	14.1844	51.06	51.89
	0.97	20.6186	74.23	
	1.01	19.802	71.29	
	2.06	9.70874	34.95	
	1.63	12.2699	44.17	
	1.35	14.8148	53.33	
	1.8	11.1111	40.00	
	1.82	10.989	39.56	
	1.3	15.3846	55.38	
	1.31	15.2672	54.96	
20	1.33	15.0376	54.14	50.81
	1.28	15.625	56.25	
	1.26	15.873	57.14	
	1.36	14.7059	52.94	
	1.8	11.1111	40.00	
	1.46	13.6986	49.32	
	1.36	14.7059	52.94	
	1.53	13.0719	47.06	
	1.5	13.3333	48.00	
	1.43	13.986	50.35	
21	1.42	14.0845	50.70	56.17
	1.35	14.8148	53.33	
	1.44	13.8889	50.00	
	0.96	20.8333	75.00	
	1.29	15.5039	55.81	
	1.38	14.4928	52.17	

Menit	Waktu Tempuh (s)	Kecepatan (m/s)	Kecepatan (km/jam)	Kecepatan Rata-rata (km/jam)
22	1.01	19.802	71.29	55.80
	1.03	19.4175	69.90	
	1.64	12.1951	43.90	
	1.04	19.2308	69.23	
	1.8	11.1111	40.00	
	1.38	14.4928	52.17	
	1.71	11.6959	42.11	
	1.38	14.4928	52.17	
	1.28	15.625	56.25	
	1.18	16.9492	61.02	
23	1.6	12.5	45.00	49.37
	1.64	12.1951	43.90	
	1.63	12.2699	44.17	
	1.26	15.873	57.14	
	1.44	13.8889	50.00	
	1.4	14.2857	51.43	
	1.44	13.8889	50.00	
	1.47	13.6054	48.98	
	1.24	16.129	58.06	
	1.6	12.5	45.00	
24	2.12	9.43396	33.96	51.78
	2.06	9.70874	34.95	
	1.2	16.6667	60.00	
	1.07	18.6916	67.29	
	1.31	15.2672	54.96	
	1.89	10.582	38.10	
	1.95	10.2564	36.92	
	1.71	11.6959	42.11	
	0.92	21.7391	78.26	
	1.01	19.802	71.29	

Menit	Waktu Tempuh (s)	Kecepatan (m/s)	Kecepatan (km/jam)	Kecepatan Rata-rata (km/jam)
25	1.24	16.129	58.06	53.77
	1.38	14.4928	52.17	
	1.64	12.1951	43.90	
	1.24	16.129	58.06	
	1.53	13.0719	47.06	
	1.21	16.5289	59.50	
	1.29	15.5039	55.81	
	1.26	15.873	57.14	
	1.38	14.4928	52.17	
26	1.76	11.3636	40.91	46.38
	2.06	9.70874	34.95	
	1.44	13.8889	50.00	
	1.42	14.0845	50.70	
	1.47	13.6054	48.98	
	1.44	13.8889	50.00	
	1.38	14.4928	52.17	
	1.41	14.1844	51.06	
	1.75	11.4286	41.14	
	1.64	12.1951	43.90	
27	1.41	14.1844	51.06	46.73
	1.12	17.8571	64.29	
	1.67	11.976	43.11	
	1.53	13.0719	47.06	
	1.76	11.3636	40.91	
	1.54	12.987	46.75	
	1.6	12.5	45.00	
	1.57	12.7389	45.86	
	1.71	11.6959	42.11	
	1.75	11.4286	41.14	

Menit	Waktu Tempuh (s)	Kecepatan (m/s)	Kecepatan (km/jam)	Kecepatan Rata-rata (km/jam)
28	1.95	10.2564	36.92	57.54
	1.26	15.873	57.14	
	1.31	15.2672	54.96	
	0.9	22.2222	80.00	
	1.16	17.2414	62.07	
	1.38	14.4928	52.17	
	1.22	16.3934	59.02	
	1.14	17.5439	63.16	
	1.18	16.9492	61.02	
	1.47	13.6054	48.98	
29	1.57	12.7389	45.86	49.99
	1.71	11.6959	42.11	
	1.33	15.0376	54.14	
	1.5	13.3333	48.00	
	1.54	12.987	46.75	
	1.44	13.8889	50.00	
	1.71	11.6959	42.11	
	1.26	15.873	57.14	
	1.47	13.6054	48.98	
	1.11	18.018	64.86	
30	1.24	16.129	58.06	52.33
	1.41	14.1844	51.06	
	1.95	10.2564	36.92	
	1.18	16.9492	61.02	
	1.38	14.4928	52.17	
	1.26	15.873	57.14	
	1.57	12.7389	45.86	
	1.36	14.7059	52.94	
	1.29	15.5039	55.81	

Menit	Waktu Tempuh (s)	Kecepatan (m/s)	Kecepatan (km/jam)	Kecepatan Rata-rata (km/jam)
31	1.1	18.1818	65.45	54.26
	2.53	7.90514	28.46	
	0.99	20.202	72.73	
	1.26	15.873	57.14	
	1.2	16.6667	60.00	
	1.27	15.748	56.69	
	1.64	12.1951	43.90	
	1.41	14.1844	51.06	
	1.26	15.873	57.14	
	1.44	13.8889	50.00	
32	1.44	13.8889	50.00	53.26
	1.89	10.582	38.10	
	1.53	13.0719	47.06	
	2.12	9.43396	33.96	
	1.67	11.976	43.11	
	0.98	20.4082	73.47	
	1.11	18.018	64.86	
	1.14	17.5439	63.16	
	1.32	15.1515	54.55	
	1.12	17.8571	64.29	
33	1.71	11.6959	42.11	46.51
	1.67	11.976	43.11	
	2	10	36.00	
	1.31	15.2672	54.96	
	1.36	14.7059	52.94	
	1.64	12.1951	43.90	
	1.53	13.0719	47.06	
	1.5	13.3333	48.00	
	1.53	13.0719	47.06	
	1.44	13.8889	50.00	

Menit	Waktu Tempuh (s)	Kecepatan (m/s)	Kecepatan (km/jam)	Kecepatan Rata-rata (km/jam)
34	1.2	16.6667	60.00	48.97
	1.26	15.873	57.14	
	2.18	9.17431	33.03	
	1.5	13.3333	48.00	
	2.19	9.13242	32.88	
	1.33	15.0376	54.14	
	1.3	15.3846	55.38	
	1.24	16.129	58.06	
	1.8	11.1111	40.00	
	1.41	14.1844	51.06	
35	1.44	13.8889	50.00	48.22
	1.95	10.2564	36.92	
	1.64	12.1951	43.90	
	1.16	17.2414	62.07	
36	1.44	13.8889	50.00	56.72
	1.89	10.582	38.10	
	0.94	21.2766	76.60	
	1.16	17.2414	62.07	
	1.26	15.873	57.14	
	1.11	18.018	64.86	
	1.16	17.2414	62.07	
	1.06	18.8679	67.92	
	1.33	15.0376	54.14	
	2.1	9.52381	34.29	

Menit	Waktu Tempuh (s)	Kecepatan (m/s)	Kecepatan (km/jam)	Kecepatan Rata-rata (km/jam)	
37	1.44	13.8889	50.00	46.47	
	1.47	13.6054	48.98		
	1.44	13.8889	50.00		
	1.39	14.3885	51.80		
	1.95	10.2564	36.92		
	2	10	36.00		
	1.67	11.976	43.11		
	1.31	15.2672	54.96		
38	1.1	18.1818	65.45	61.70	
	0.99	20.202	72.73		
	1.26	15.873	57.14		
	1.44	13.8889	50.00		
	1.14	17.5439	63.16		
39	1.57	12.7389	45.86	51.52	
	1.44	13.8889	50.00		
	1.26	15.873	57.14		
	2.4	8.33333	30.00		
	1.38	14.4928	52.17		
	1.24	16.129	58.06		
	1.67	11.976	43.11		
	1.24	16.129	58.06		
	1.04	19.2308	69.23		

Menit	Waktu Tempuh (s)	Kecepatan (m/s)	Kecepatan (km/jam)	Kecepatan Rata-rata (km/jam)
40	1.18	16.9492	61.02	50.18
	1.71	11.6959	42.11	
	1.38	14.4928	52.17	
	1.44	13.8889	50.00	
	1.5	13.3333	48.00	
	1.1	18.1818	65.45	
	1.6	12.5	45.00	
	1.53	13.0719	47.06	
	1.53	13.0719	47.06	
	1.64	12.1951	43.90	
41	1.95	10.2564	36.92	55.89
	1.22	16.3934	59.02	
	1.31	15.2672	54.96	
	2.06	9.70874	34.95	
	0.97	20.6186	74.23	
	1.01	19.802	71.29	
	1.36	14.7059	52.94	
	1.31	15.2672	54.96	
	1.13	17.6991	63.72	
42	1.29	15.5039	55.81	51.09
	1.67	11.976	43.11	
	2	10	36.00	
	1.24	16.129	58.06	
	1.22	16.3934	59.02	
	1.03	19.4175	69.90	
	1.29	15.5039	55.81	
	2.32	8.62069	31.03	

Menit	Waktu Tempuh (s)	Kecepatan (m/s)	Kecepatan (km/jam)	Kecepatan Rata-rata (km/jam)
43	1.3	15.3846	55.38	50.80
	1.85	10.8108	38.92	
	1.33	15.0376	54.14	
	1.38	14.4928	52.17	
	1.31	15.2672	54.96	
	1.3	15.3846	55.38	
	1.36	14.7059	52.94	
	1.44	13.8889	50.00	
	1.53	13.0719	47.06	
	1.53	13.0719	47.06	
44	1.57	12.7389	45.86	45.17
	1.64	12.1951	43.90	
	1.47	13.6054	48.98	
	1.24	16.129	58.06	
	1.5	13.3333	48.00	
	1.29	15.5039	55.81	
	1.76	11.3636	40.91	
	1.8	11.1111	40.00	
	1.75	11.4286	41.14	
	2.48	8.06452	29.03	
45	1.39	14.3885	51.80	53.58
	1.94	10.3093	37.11	
	2.18	9.17431	33.03	
	1.2	16.6667	60.00	
	1.6	12.5	45.00	
	1.16	17.2414	62.07	

	0.85	23.5294	84.71	
	1.31	15.2672	54.96	

Menit	Waktu Tempuh (s)	Kecepatan (m/s)	Kecepatan (km/jam)	Kecepatan Rata-rata (km/jam)
46	1.2	16.6667	60.00	43.87
	2.18	9.17431	33.03	
	1.31	15.2672	54.96	
	1.5	13.3333	48.00	
	1.67	11.976	43.11	
	2.33	8.58369	30.90	
	1.94	10.3093	37.11	
47	2.06	9.70874	34.95	50.46
	2.05	9.7561	35.12	
	0.83	24.0964	86.75	
	1.6	12.5	45.00	
48	1.4	14.2857	51.43	53.40
	1.67	11.976	43.11	
	1.8	11.1111	40.00	
	1.31	15.2672	54.96	

	1.03	19.4175	69.90
	1.18	16.9492	61.02

Menit	Waktu Tempuh (s)	Kecepatan (m/s)	Kecepatan (km/jam)	Kecepatan Rata-rata (km/jam)
49	1.1	18.1818	65.45	53.02
	1.12	17.8571	64.29	
	1.24	16.129	58.06	
	1.41	14.1844	51.06	
	1.6	12.5	45.00	
	1.53	13.0719	47.06	
	1.5	13.3333	48.00	
	1.38	14.4928	52.17	
	1.41	14.1844	51.06	
	1.5	13.3333	48.00	
50	1.48	13.5135	48.65	50.31
	2.05	9.7561	35.12	
	1.57	12.7389	45.86	
	1.36	14.7059	52.94	
	1.5	13.3333	48.00	
	1.01	19.802	71.29	
51	1.13	17.6991	63.72	42.82
	1.01	19.802	71.29	

1.26	15.873	57.14
2.77	7.22022	25.99
2.06	9.70874	34.95
1.95	10.2564	36.92
2.4	8.33333	30.00
1.94	10.3093	37.11
2	10	36.00
2.05	9.7561	35.12

Menit	Waktu Tempuh (s)	Kecepatan (m/s)	Kecepatan (km/jam)	Kecepatan Rata-rata (km/jam)
52	1.14	17.5439	63.16	48.66
	2.18	9.17431	33.03	
	1.29	15.5039	55.81	
	1.53	13.0719	47.06	
	1.76	11.3636	40.91	
	1.47	13.6054	48.98	
	1.6	12.5	45.00	
	1.38	14.4928	52.17	
	1.39	14.3885	51.80	
53	1.76	11.3636	40.91	43.78
	1.67	11.976	43.11	
	2	10	36.00	
	1.76	11.3636	40.91	
	1.68	11.9048	42.86	
	1.6	12.5	45.00	
	1.53	13.0719	47.06	
	1.6	12.5	45.00	
	1.57	12.7389	45.86	
	1.41	14.1844	51.06	
54	2.12	9.43396	33.96	54.13
	1.47	13.6054	48.98	

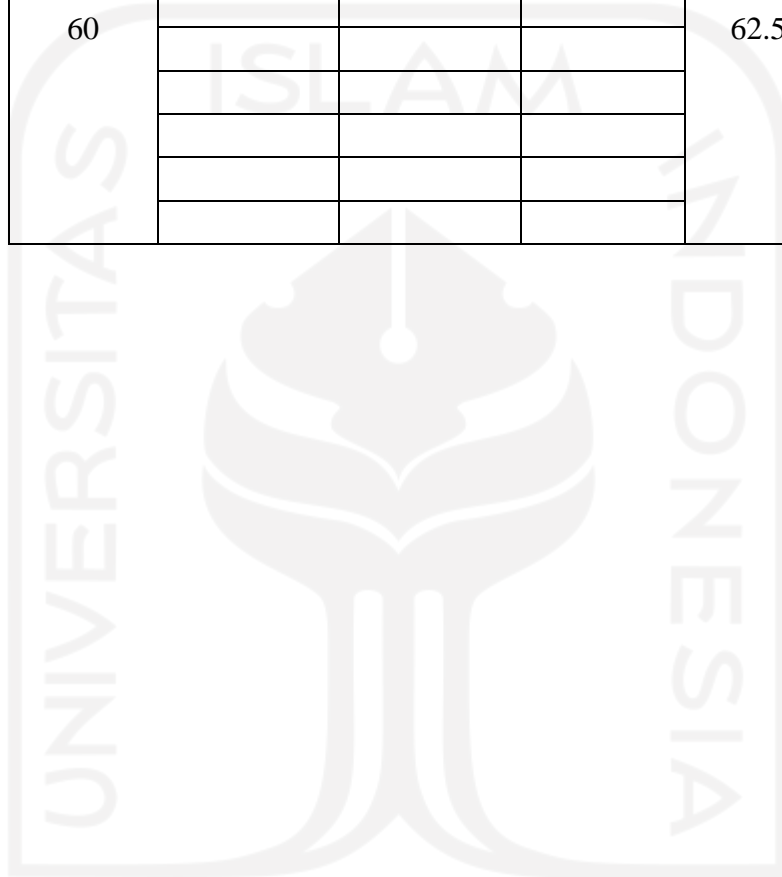
1.14	17.5439	63.16
1.03	19.4175	69.90
1.6	12.5	45.00
1.03	19.4175	69.90
1.5	13.3333	48.00

Menit	Waktu Tempuh (s)	Kecepatan (m/s)	Kecepatan (km/jam)	Kecepatan Rata-rata (km/jam)
55	1.24	16.129	58.06	45.76
	2.12	9.43396	33.96	
	1.33	15.0376	54.14	
	2.4	8.33333	30.00	
	1.24	16.129	58.06	
	2.57	7.7821	28.02	
	1.24	16.129	58.06	
56	1.18	16.9492	61.02	51.33
	1.26	15.873	57.14	
	1.36	14.7059	52.94	
	1.31	15.2672	54.96	
	1.89	10.582	38.10	
	1.67	11.976	43.11	
	1.44	13.8889	50.00	
	1.47	13.6054	48.98	
	1.33	15.0376	54.14	
	1.36	14.7059	52.94	

57	1.35	14.8148	53.33	48.95
	1.8	11.1111	40.00	
	1.5	13.3333	48.00	
	1.47	13.6054	48.98	
	1.57	12.7389	45.86	
	2.18	9.17431	33.03	
	1.53	13.0719	47.06	
	1.2	16.6667	60.00	
	1.38	14.4928	52.17	
	1.18	16.9492	61.02	

Menit	Waktu Tempuh (s)	Kecepatan (m/s)	Kecepatan (km/jam)	Kecepatan Rata-rata (km/jam)
58	1.29	15.5039	55.81	47.72
	1.63	12.2699	44.17	
	2.4	8.33333	30.00	
	1.06	18.8679	67.92	
	1.53	13.0719	47.06	
	1.64	12.1951	43.90	
	1.64	12.1951	43.90	
	1.47	13.6054	48.98	
59	1.64	12.1951	43.90	45.77
	1.22	16.3934	59.02	
	1.44	13.8889	50.00	
	1.38	14.4928	52.17	
	1.95	10.2564	36.92	
	1.8	11.1111	40.00	
	1.57	12.7389	45.86	
	1.67	11.976	43.11	

	1.64	12.1951	43.90	
	1.68	11.9048	42.86	
60	1.64	12.1951	43.90	62.56
	0.84	23.8095	85.71	
	1.24	16.129	58.06	



الجامعة الإسلامية
الاستد بالاندو

Lampiran II Hasil Survei Volume Menggunakan Peralatan Konvensional

Menit	LV	HV	LV (smp)	HV (smp)	Volume Kendaraan (smp/menit)
1	16	2	16	2.6	18.6
2	13	2	13	2.6	15.6
3	15	0	15	0.0	15.0
4	21	1	21	1.3	22.3
5	12	1	12	1.3	13.3
6	20	1	20	1.3	21.3
7	7	1	7	1.3	8.3
8	12	1	12	1.3	13.3
9	19	1	19	1.3	20.3
10	19	0	19	0.0	19.0
11	10	0	10	0.0	10.0
12	29	3	29	3.9	32.9
13	14	0	14	0.0	14.0
14	14	2	14	2.6	16.6
15	15	2	15	2.6	17.6
16	12	0	12	0.0	12.0
17	32	2	32	2.6	34.6
18	10	0	10	0.0	10.0
19	13	0	13	0.0	13.0
20	20	4	20	5.2	25.2
21	8	0	8	0.0	8.0
22	14	0	14	0.0	14.0
23	29	0	29	0.0	29.0
24	12	0	12	0.0	12.0
25	15	1	15	1.3	16.3
26	29	1	29	1.3	30.3
27	15	1	15	1.3	16.3
28	10	2	10	2.6	12.6
29	21	1	21	1.3	22.3
30	15	0	15	0.0	15.0

Menit	LV	HV	LV (smp)	HV (smp)	Volume Kendaraan (smp/menit)
31	29	1	29	1.3	30.3
32	18	0	18	0.0	18.0
33	40	0	40	0.0	40.0
34	10	0	10	0.0	10.0
35	6	0	6	0.0	6.0
36	19	1	19	1.3	20.3
37	30	3	30	3.9	33.9
38	11	1	11	1.3	12.3
39	12	0	12	0.0	12.0
40	14	3	14	3.9	17.9
41	16	1	16	1.3	17.3
42	11	0	11	0.0	11.0
43	33	0	33	0.0	33.0
44	16	0	16	0.0	16.0
45	14	1	14	1.3	15.3
46	17	7	17	9.1	26.1
47	15	0	15	0.0	15.0
48	8	2	8	2.6	10.6
49	19	3	19	3.9	22.9
50	6	0	6	0.0	6.0
51	7	3	7	3.9	10.9
52	9	0	9	0.0	9.0
53	21	2	21	2.6	23.6
54	7	0	7	0.0	7.0
55	8	0	8	0.0	8.0
56	28	3	28	3.9	31.9
57	14	1	14	1.3	15.3
58	8	0	8	0.0	8.0
59	28	4	28	5.2	33.2
60	3	0	3	0.0	3.0

Lampiran III Hasil Perhitungan Kepadatan Menggunakan Perangkat Konvensional

Menit	Volume (smp/menit)	Kecepatan (km/menit)	Kepadatan (smp/km)
1	18,6	0,91	21
2	15,6	0,88	18
3	15	0,88	18
4	22,3	0,88	26
5	13,3	0,83	17
6	21,3	0,78	28
7	8,3	0,78	11
8	13,3	0,82	17
9	20,3	1,00	21
10	19	0,76	26
11	10	1,00	11
12	32,9	0,78	43
13	14	1,00	15
14	16,6	0,83	20
15	17,6	0,85	21
16	12	0,93	13
17	34,6	0,76	46
18	10	0,96	11
19	13	0,86	16
20	25,2	0,85	30
21	8	0,94	9
22	14	0,93	16
23	29	0,82	36
24	12	0,86	14
25	16,3	0,90	19
26	30,3	0,77	40
27	16,3	0,78	21
28	12,6	0,96	14
29	22,3	0,83	27
30	15	0,87	18

Menit	Volume (smp/menit)	Kecepatan (km/menit)	Kepadatan (smp/km)
31	30,3	0,90	34
32	18	0,89	21
33	40	0,78	52
34	10	0,82	13
35	6	0,80	8
36	20,3	0,95	22
37	33,9	0,77	44
38	12,3	1,03	12
39	12	0,86	14
40	17,9	0,84	22
41	17,3	0,93	19
42	11	0,85	13
43	33	0,85	39
44	16	0,75	22
45	15,3	0,89	18
46	26,1	0,73	36
47	15	0,84	18
48	10,6	0,89	12
49	22,9	0,88	26
50	6	0,84	8
51	10,9	0,71	16
52	9	0,81	12
53	23,6	0,73	33
54	7	0,90	8
55	8	0,76	11
56	31,9	0,86	38
57	15,3	0,82	19
58	8	0,80	11
59	33,2	0,76	44
60	3	1,04	3

Lampiran IV Selisih Hasil Survei Volume Menggunakan Perangkat Konvensional dengan Survei Volume Menggunakan *Survey App*

Menit	Volume Perangkat Konvensional	Volume <i>Survey App</i>	Selisih (%)
1	18,6	18,6	0
2	15,6	15,6	0
3	15	15	0
4	22,3	22,3	0
5	13,3	13,3	0
6	21,3	21,3	0
7	8,3	8,3	0
8	13,3	13,3	0
9	20,3	20,3	0
10	19	19	0
11	10	10	0
12	32,9	32,9	0
13	14	14	0
14	16,6	16,6	0
15	17,6	17,6	0
16	12	12	0
17	34,6	34,6	0
18	10	10	0
19	13	13	0
20	25,2	25,2	0
21	8	8	0
22	14	14	0
23	29	29	0
24	12	12	0
25	16,3	16,3	0
26	30,3	30,3	0
27	16,3	16,3	0
28	12,6	12,6	0
29	22,3	22,3	0
30	15	15	0
31	30,3	30,3	0
32	18	18	0
33	40	40	0
34	10	10	0

35	6	6	0
----	---	---	---

Menit	Volume Perangkat Konvensional (smp/menit)	Volume <i>Survey App</i> (smp/menit)	Selisih (%)
36	20,3	20,3	0
37	33,9	33,9	0
38	12,3	12,3	0
39	12	12	0
40	17,9	17,9	0
41	17,3	17,3	0
42	11	11	0
43	33	33	0
44	16	16	0
45	15,3	15,3	0
46	26,1	26,1	0
47	15	15	0
48	10,6	10,6	0
49	22,9	22,9	0
50	6	6	0
51	10,9	10,9	0
52	9	9	0
53	23,6	23,6	0
54	7	7	0
55	8	8	0
56	31,9	31,9	0
57	15,3	15,3	0
58	8	8	0
59	33,2	33,2	0
60	3	3	0

Lampiran V Selisih Hasil Survei Kecepatan Menggunakan Perangkat Konvensional dengan Survei Volume Menggunakan *Survey App*

Menit	Kecepatan Perangkat Konvensional (km/menit)	Kecepatan <i>Survey App</i> (km/menit)	Selisih (%)
1	0,91	0,90	-1,0
2	0,88	0,88	0,0
3	0,88	0,88	0,5
4	0,88	0,86	-1,4
5	0,83	0,79	-4,7
6	0,78	0,74	-4,9
7	0,78	0,77	-2,3
8	0,82	0,75	-8,9
9	1,00	0,87	-12,8
10	0,76	0,73	-3,4
11	1,00	0,95	-4,6
12	0,78	0,75	-3,2
13	1,00	0,92	-8,1
14	0,83	0,79	-5,3
15	0,85	0,85	-0,2
16	0,93	0,87	-5,5
17	0,76	0,74	-2,5
18	0,96	0,89	-7,4
19	0,86	0,83	-3,7
20	0,85	0,83	-2,2
21	0,94	0,83	-11,0
22	0,93	0,85	-9,1
23	0,82	0,80	-2,6
24	0,86	0,74	-14,6
25	0,90	0,88	-2,3
26	0,77	0,77	-1,0
27	0,78	0,74	-4,7
28	0,96	0,94	-2,0
29	0,83	0,80	-3,5
30	0,87	0,86	-1,4

Menit	Kecepatan Perangkat Konvensional (km/menit)	Kecepatan Survey App (km/menit)	Selisih (%)
31	0,90	0,87	-3,3
32	0,89	0,87	-1,6
33	0,78	0,75	-3,1
34	0,82	0,78	-4,3
35	0,80	0,78	-3,4
36	0,95	0,90	-4,8
37	0,77	0,79	1,5
38	1,03	0,97	-6,0
39	0,86	0,79	-8,6
40	0,84	0,81	-3,0
41	0,93	0,84	-9,6
42	0,85	0,81	-4,5
43	0,85	0,83	-1,8
44	0,75	0,75	-0,2
45	0,89	0,87	-2,5
46	0,73	0,72	-1,3
47	0,84	0,78	-7,5
48	0,89	0,80	-10,0
49	0,88	0,85	-3,8
50	0,84	0,76	-9,9
51	0,71	0,64	-10,8
52	0,81	0,80	-0,9
53	0,73	0,70	-4,2
54	0,90	0,80	-11,7
55	0,76	0,76	0,2
56	0,86	0,85	-0,5
57	0,82	0,79	-3,7
58	0,80	0,73	-8,4
59	0,76	0,74	-2,8
60	1,04	1,01	-3,3

Lampiran VI Selisih Hasil Kepadatan Menggunakan Perangkat Konvensional dengan Survei Volume Menggunakan *Survey App*

Menit	Kepadatan Perangkat Konvensional (km/menit)	Kepadatan <i>Survey App</i> (km/menit)	Selisih (%)
1	20,43	20,64	1,00
2	17,65	17,66	0,03
3	17,14	17,05	-0,54
4	25,43	25,80	1,44
5	16,10	16,89	4,89
6	27,33	28,75	5,20
7	10,59	10,84	2,38
8	16,25	17,84	9,75
9	20,32	23,30	14,64
10	25,12	25,99	3,47
11	10,04	10,52	4,81
12	42,43	43,83	3,29
13	14,02	15,26	8,84
14	19,92	21,04	5,61
15	20,67	20,72	0,21
16	12,97	13,72	5,81
17	45,65	46,80	2,54
18	10,47	11,30	7,94
19	15,03	15,60	3,81
20	29,76	30,44	2,29
21	8,55	9,60	12,37
22	15,05	16,57	10,05
23	35,24	36,20	2,72
24	13,90	16,29	17,14
25	18,19	18,62	2,37
26	39,20	39,60	1,03
27	20,93	21,95	4,90
28	13,14	13,40	2,02
29	26,76	27,74	3,65
30	17,20	17,44	1,42

Menit	Kecepatan Perangkat Konvensional (km/menit)	Kecepatan <i>Survey App</i> (km/menit)	Selisih (%)
31	33,51	34,64	3,38
32	20,28	20,60	1,58
33	51,60	53,26	3,22
34	12,25	12,80	4,46
35	7,47	7,72	3,47
36	21,47	22,57	5,09
37	43,77	43,12	-1,47
38	11,96	12,73	6,40
39	13,98	15,29	9,38
40	21,40	22,07	3,11
41	18,57	20,54	10,60
42	12,92	13,52	4,68
43	38,97	39,70	1,86
44	21,25	21,30	0,24
45	17,13	17,57	2,56
46	35,69	36,15	1,29
47	17,84	19,28	8,07
48	11,91	13,24	11,14
49	25,92	26,94	3,95
50	7,16	7,94	10,93
51	15,27	17,11	12,07
52	11,10	11,20	0,92
53	32,35	33,76	4,37
54	7,76	8,79	13,24
55	10,49	10,47	-0,23
56	37,29	37,46	0,47
57	18,76	19,48	3,88
58	10,06	10,98	9,17
59	43,52	44,76	2,86
60	2,88	2,97	3,39

Lampiran VII Uji Perbandingan Hasil Survei Menggunakan *Survey App* dengan Hasil Survei Menggunakan Perangkat Konvensional

Perbandingan hasil survei dilakukan dengan dua metode uji, yaitu uji t berpasangan dan uji wilcoxon. Dalam penelitian ini ditentukan tingkat kepercayaan 95%. Hasil survei menggunakan peralatan konvensional dapat dilihat pada Lampiran I, II, dan III. Adapun untuk survei volume dapat dilihat bahwa tidak ada perbedaan antara survei menggunakan perangkat lunak *Android* dengan survei menggunakan peralatan konvensional sehingga pengujian pada penelitian ini hanya dilakukan untuk perhitungan kecepatan dan kepadatan.

Dalam penelitian ini ditetapkan dua jenis hipotesis yaitu H_0 dan H_1 . Adapun hipotesis yang diambil adalah sebagai berikut.

H_0 : Tidak ada perbedaan signifikan antara survei menggunakan perangkat lunak dengan survei menggunakan perangkat konvensional

H_1 : Ada perbedaan signifikan antara survei menggunakan perangkat lunak dengan survei menggunakan perangkat konvensional

Uji t berpasangan dan uji wilcoxon dilakukan dengan perangkat lunak *Microsoft Excel*. Adapun rekapitulasi hasil uji t untuk perhitungan kecepatan dan kepadatan dapat dilihat pada Tabel 5.2 dan Tabel 5.3 berikut.

Tabel 5.2 Rekapitulasi Hasil Uji t Berpasangan Data Kecepatan dari Kedua

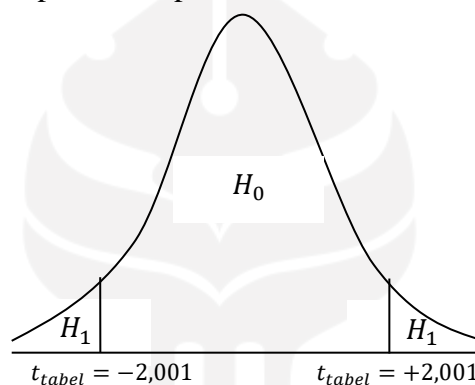
Metode Survei

	Survei Konvensional	Survei <i>Android</i>
Rata-rata	0,85	0,82
Standar Defiasi	0,077	0,071
Jumlah Sampel	60	60
Derajat Bebas	59	
Nilai t Statistik	-9,142	
Nilai t Kritis	±2,001	

Tabel 5.3 Rekapitulasi Hasil Uji t Berpasangan Data Kepadatan dari Kedua Metode Survei

	Survei Konvensional	Survei <i>Android</i>
Rata-rata	20,99992	21,82272
Standar Defiasi	11,186	11,255
Jumlah Sampel	60	60
Derajat Bebas	59	
Nilai t Statistik	9,949	
Nilai t Kritis	$\pm 2,001$	

Daerah keputusan uji t berpasangan untuk survei kecepatan dan kepadatan dengan kedua metode dapat dilihat pada Gambar 5.12 Berikut.



Gambar 5.12 Daerah Keputusan berdasarkan Uji t

Nilai t statistik untuk survei kecepatan adalah sebesar -9,142, sedangkan untuk survei kepadatan adalah sebesar 9,949. Dengan menggunakan daerah keputusan di atas, dapat disimpulkan bahwa keputusan yang diambil adalah H_1 , yang berarti terdapat perbedaan yang signifikan antara survei konvensional dengan survei menggunakan perangkat lunak *Android*.

Sedangkan rekapitulasi hasil uji wilcoxon untuk perhitungan kecepatan dan kepadatan dapat dilihat pada Tabel 5.4 berikut.

Tabel 5.4 Rekapitulasi Hasil Uji Wilcoxon Data Kecepatan dan Kepadatan dari Kedua Metode Survei

	Survei Kecepatan	Survei Kepadatan
<i>Ties</i> (Selisih 0)	0	0
Total Ranking Positif	19	1799
Total Ranking Negatif	1811	31
Rata-rata Ranking Positif	6,333	5,039
Rata-rata Ranking Negatif	31,772	10,333
Nilai w Statistik	19	31
Nilai w Kritis	915	915
Konversi Nilai z	-6,596	-6,508
Nilai z Pengujian 2 Arah	$4,22 \times 10^{-11}$	$7,633 \times 10^{-11}$

Dari uji wilcoxon yang dilakukan untuk kedua metode survei, baik survei kecepatan maupun kepadatan, diperoleh nilai w statistik yang lebih kecil daripada nilai w kritis. Ketika nilai w dikonversikan menjadi nilai z, kedua survei memiliki nilai z pengujian 2 arah yang lebih kecil dari 0,05. Sehingga H_0 ditolak dan H_1 diterima, yang berarti ada perbedaan yang signifikan antara survei menggunakan perangkat lunak dengan survei secara konvensional.

Selisih antara kedua metode survei tersebut lebih cenderung terjadi pada survei kecepatan. Perbedaan data kecepatan dari kedua metode survei berpengaruh terhadap perbedaan data kepadatan. Perbedaan cenderung terjadi karena faktor *human error*, yaitu adanya perbedaan reflek surveyor dari masing-masing metode survei dalam menekan tombol start atau stop pada stopwatch dan aplikasi pada saat pengukuran waktu tempuh kendaraan serta perbedaan persepsi penglihatan surveyor pada saat menentukan ketika kendaraan sudah masuk atau keluar dari area hitung. Sedangkan pada data volume tidak terjadi perbedaan hasil dikarenakan masing-masing surveyor menghitung kendaraan dengan jumlah yang sama sesuai dengan keadaan di lapangan.

Lampiran VIII Source Code “Survey App”

1. Menu Utama

```
class Main extends StatefulWidget {
  @override
  _MainState createState() => _MainState();
}

class _MainState extends State<Main> {

  final GlobalKey<ScaffoldState> _key = new GlobalKey<ScaffoldState>();

  int _currentFragment = 0;

  @override
  void initState() {
    Future.delayed(Duration(milliseconds: 200), () {
      checkPermission();
    });
    super.initState();
  }

  @override
  void dispose() {
    // TODO: implement dispose
    super.dispose();
  }

  checkPermission() async {
    await Permission.storage.status.then((status) async {
      await [Permission.storage].request();
    });
  }
}
```

```

if (!await Permission.storage.status.isGranted) {
  final snackBar = SnackBar(
    content: Text('Anda harus mengizinkan akses penyimpanan!'),
    action: SnackBarAction(
      label: 'OK',
      onPressed: () {
        _key.currentState.hideCurrentSnackBar();
      },
    ),
  );
  _key.currentState.showSnackBar(snackBar);
}
}).catchError((onError) {
  print('onError : $onError');
});
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    key: _key,
    appBar: AppBar(
      title: Text('Survey App'),
    ),
    body: body(),
    bottomNavigationBar: bottomNavigation(),
  );
}

Widget body() {
  switch(_currentFragment) {

```



```

case 0:
    return FragmentSurveySpeed();
case 1:
    return FragmentSurveyVolume();
case 2:
    return FragmentHitungKepadatan();
default:
    return null;
}
}

```

```

Widget bottomNavigation() {
return BottomNavigationBar(
    currentIndex: _currentFragment,
    type: BottomNavigationBarType.fixed,
    onTap: bottomNavPressed,
    items: bottomList.map((BottomList bottomList) {
        return BottomNavigationBarItem(
            icon: Icon(bottomList.icon),
            title: Text(bottomList.title),
        );
    }).toList()
);
}

```

```

bottomNavPressed(int index) {
    setState(() => _currentFragment = index);
}
}

```

```

class BottomList {

```

```

final int index;
final String title;
final IconData icon;
const BottomList({this.index, this.title, this.icon});
}

const List<BottomList> bottomList = <BottomList>[
  const BottomList(index: 0, title: 'Survey Speed', icon: Icons.shutter_speed),
  const BottomList(index: 1, title: 'Survey Volume', icon:
Icons.departure_board),
  const BottomList(index: 2, title: 'Hitung Kepadatan', icon: Icons.event_note),
];

```

2. Tampilan Survei Kecepatan

```

class FragmentSurveySpeed extends StatefulWidget {
  @override
  _FragmentSurveySpeedState createState() => _FragmentSurveySpeedState();
}

class _FragmentSurveySpeedState extends State<FragmentSurveySpeed> {

  final GlobalKey<RefreshIndicatorState> _refreshIndicatorKey = new
GlobalKey<RefreshIndicatorState>();

  List<FileSystemEntity> listFile = new List();

  bool _firstLoad = true;

  @override
  void initState() {
    checkFolder();

```

```

initGetListofFile();
super.initState();
}

@override
void dispose() {
  // TODO: implement dispose
  super.dispose();
}

checkFolder() async {
  final directory = new Directory('/storage/emulated/0/Download/Data
Survey/Kecepatan');
  if (!await directory.exists()) {
    directory.create();
  }
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: body(),
  );
}

Widget body() {
  return Column(
    children: <Widget>[
      Expanded(
        child: RefreshIndicator(
          key: _refreshIndicatorKey,

```

```

onRefresh: refresh,
child: _firstLoad ? Container() :
listFile.length > 0 ? ListView.separated(
  separatorBuilder: (context, index) => Divider(
    color: Colors.grey.withOpacity(0.3),
    height: 1.0,
  ),
  itemCount: listFile.length,
  itemBuilder: (context, index) {
    return ListTile(
      title: Text(path.basename(listFile[index].path)),
      onTap: () => false,
    );
  },
):
Center(
  child: MaterialButton(
    splashColor: Colors.transparent,
    hoverColor: Colors.transparent,
    highlightColor: Colors.transparent,
    onPressed: () => refresh(),
    child: Column(
      mainAxisAlignment: MainAxisAlignment.center,
      crossAxisAlignment: CrossAxisAlignment.center,
      mainAxisAlignment: MainAxisAlignment.min,
      children: <Widget>[
        Icon(Icons.insert_drive_file, color: Color(0xFFCFCFCF), size:
85.0),
        Padding(
          padding: EdgeInsets.symmetric(vertical: 10.0, horizontal: 10.0),

```

```

        child: Text('Tidak ada data', style: TextStyle(color:
Color(0xFF707070), fontSize: 17.0, fontWeight: FontWeight.bold)),
      ),
      Text('Ketuk untuk refresh', style: TextStyle(color:
Color(0xFF9E9E9E), fontSize: 16.0, fontWeight: FontWeight.w300)),
    ],
  ),
),
),
),
),
),
),
Container(
  height: 50.0,
  width: MediaQuery.of(context).size.width,
  child: RaisedButton(
    color: Colors.blue,
    elevation: 0.0,
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(0.0),
    ),
    onPressed: () => Navigator.push(context, MaterialPageRoute(builder:
(context) => ViewInputDataPreparation())),
    child: Text('TAMBAH SURVEY', style: TextStyle(color: Colors.white,
fontSize: 16.0)),
  ),
),
),
],
);
}

initGetListofFile() {

```

```

getListOfFile().then((List list) {
  _firstLoad = false;
  setState(() => listFile = list);
}).catchError((onError) {
  print('onError : $onError');
});
}

Future refresh() async {
  await getListOfFile().then((List list) {
    setState(() => listFile = list);
  }).catchError((onError) {
    print('onError : $onError');
  });
}

Future<List> getListOfFile() async {
  final directory = new Directory('/storage/emulated/0/Download/Data
Survey/Kecepatan');
  List list;
  list = directory.listFiles(recursive: true, followLinks: false);
  List data = list.reversed.toList();
  return data;
}
}

```

3. Proses Perhitungan Survei Kecepatan

```
Function setStatus;
```

```
int firstTimerStatus = 0;
```

```
class ViewInputDataSurveySpeed extends StatefulWidget {
  final List data;
  ViewInputDataSurveySpeed({this.data});
  @override
  _ViewInputDataSurveySpeedState createState() =>
  _ViewInputDataSurveySpeedState();
}
```

```
class _ViewInputDataSurveySpeedState extends
State<ViewInputDataSurveySpeed> with TickerProviderStateMixin {

  final GlobalKey<ScaffoldState> _key = new GlobalKey<ScaffoldState>();

  AnimationController _controller;
  ScrollController _scrollController = new ScrollController();

  final StopwatchTimer _stopWatchTimer = StopwatchTimer();

  var _stopwatch = new Stopwatch();
  bool _initStartStatus = true;
  bool _startStatus = true; // true == START & false == STOP
  List _waktuKendaraan = new List();
  List _kecepatanMS = new List();
  List _kecepatanKmJam = new List();
  List<SpeedSurveyList> _speedSurveyList = new List();
```

```
var timer;
```

```
@override
void initState() {
  print(widget.data);
  _controller = AnimationController(vsync: this, duration: Duration(minutes:
int.parse(widget.data[5])));
  setStatus = checkStatus;
  firstTimerStatus = 0;
  super.initState();
}
```

```
@override
void dispose() {
  _controller.stop(canceled: true);
  _scrollController.dispose();
  _stopwatch.stop();
  if (timer != null && timer.isActive) {
    timer.cancel();
  }
  _stopWatchTimer.dispose();
  super.dispose();
}
```

```
@override
Widget build(BuildContext context) {
  return Scaffold(
    key: _key,
    appBar: AppBar(
      title: Text('Input Data Survey Speed'),
    ),
```



```

    body: body(),
  );
}

```

```

Widget body() {
  return Container(
    padding: const EdgeInsets.symmetric(vertical: 20.0, horizontal: 10.0),
    child: Column(
      children: <Widget>[
        Container(
          child: Column(
            children: <Widget>[
              Countdown(
                animation: StepTween(
                  begin: int.parse(widget.data[5]) * 60,
                  end: 0,
                ).animate(_controller)
              ),
              Container(
                height: 40.0,
                width: MediaQuery.of(context).size.width,
                margin: const EdgeInsets.symmetric(vertical: 15.0),
                child: RaisedButton(
                  elevation: _initStartStatus ? 2.0 : 0.0,
                  color: _initStartStatus ? Colors.blue : Colors.grey,
                  onPressed: () {
                    if (_initStartStatus) {
                      setState(() => _initStartStatus = false );
                      _controller.forward();
                    }
                  },
                ),
              ),
            ],
          ),
        ),
      ],
    ),
  ),
}

```

```

        child: Text('START', style: TextStyle(color: Colors.white)),
      ),
    ),
  ],
)
),
Container(
  padding: const EdgeInsets.symmetric(vertical: 5.0, horizontal: 5.0),
  alignment: Alignment.centerLeft,
  child: Wrap(
    children: <Widget>[
      Text(widget.data[0]+' | '+widget.data[1]+' | '+widget.data[2]+' |
'+widget.data[3]+' | '+widget.data[4]+' | '+widget.data[5]+' Menit |
'+widget.data[6]+' Meter'),
    ],
  ),
),
Expanded(
  child: Container(
    width: MediaQuery.of(context).size.width,
    child: SingleChildScrollView(
      controller: _scrollController,
      physics: BouncingScrollPhysics(),
      child: DataTable(
        columns: <DataColumn>[
          DataColumn(label: Text("Menit")),
          DataColumn(label: Text("Kecepatan Rata-rata(km/jam)")),
        ],
        rows: _speedSurveyList.map((SpeedSurveyList list) {
          return DataRow(
            cells: <DataCell>[

```

```

        DataCell(Text(list.menit.toString())),
        DataCell(Text(list.average.toString())),
    ],
);
}).toList()
),
),
)
),
Container(
  padding: const EdgeInsets.only(top: 25.0),
  child: Column(
    mainAxisAlignment: MainAxisAlignment.end,
    children: <Widget>[
      StreamBuilder<int>(
        stream: _stopWatchTimer.rawTime,
        initialData: 0,
        builder: (context, snap) {
          final value = snap.data;
          final displayTime = StopwatchTimer.getDisplayTime(value);
          return Column(
            children: <Widget>[
              Padding(
                padding: const EdgeInsets.all(8),
                child: Text(
                  displayTime,
                  style: TextStyle(
                    fontSize: 30,
                    fontFamily: 'Helvetica',
                    fontWeight: FontWeight.bold
                  ),
                ),
            ],
          ),
        ),
      ),
    ],
  ),
),

```

```

        ),
    ),
],
);
},
),
RaisedButton(
    padding: const EdgeInsets.all(80.0),
    shape: CircleBorder(),
    color: !_initStartStatus || isDoneStatus ? Colors.grey : !_startStatus ?
Colors.green : Colors.red,
    onPressed: () => !_initStartStatus && !isDoneStatus ?
buttonPressed() : false,
    child: Text(_startStatus ? 'START' : 'STOP', style: TextStyle(color:
Colors.white, fontSize: 35.0)),
),
],
),
),
),
),
);
}

String currentTime = "";
bool isDoneStatus = false;

checkStatus(bool isDone, String time) {
    if (!isDoneStatus && isDone) {
        isDoneStatus = true;
        print('Done');
    }
}

```

```

print(_speedSurveyList.length);
Future.delayed(Duration(milliseconds: 200), () {
  setState(() => isDoneStatus);
});
createExcelFile();
} else if (!isDoneStatus && !isDone) {
  if (currentTime.isEmpty) {
    currentTime = time;
    calculateData();
  } else {
    if (currentTime != time) {
      currentTime = time;
      calculateData();
    }
  }
}
}

createExcelFile() {
  var excel = packageExcel.Excel.createExcel();
  var sheet = 'Sheet1';
  packageExcel.TextWrapping wrap =
packageExcel.TextWrapping.WrapText;
  // Nama Surveyor
  excel.updateCell(sheet, packageExcel.CellIndex.indexByString("A1"),
"Nama Surveyor", wrap: wrap);
  excel.updateCell(sheet, packageExcel.CellIndex.indexByString("B1"),
widget.data[0], wrap: wrap);
  // Lokasi Survey
  excel.updateCell(sheet, packageExcel.CellIndex.indexByString("A2"),
"Lokasi Survey", wrap: wrap);

```

```

        excel.updateCell(sheet, packageExcel.CellIndex.indexByString("B2"),
widget.data[1], wrap: wrap);
        // Arah Survey
        excel.updateCell(sheet, packageExcel.CellIndex.indexByString("A3"),
"Arah Survey", wrap: wrap);
        excel.updateCell(sheet, packageExcel.CellIndex.indexByString("B3"),
widget.data[2], wrap: wrap);
        // Tanggal & Waktu Survey
        excel.updateCell(sheet, packageExcel.CellIndex.indexByString("A4"),
"Tanggal & Waktu Survey", wrap: wrap);
        excel.updateCell(sheet, packageExcel.CellIndex.indexByString("B4"),
widget.data[3]+' '+widget.data[4], wrap: wrap);
        // Jangka Waktu Survey
        excel.updateCell(sheet, packageExcel.CellIndex.indexByString("A5"),
"Jangka Waktu", wrap: wrap);
        excel.updateCell(sheet, packageExcel.CellIndex.indexByString("B5"),
widget.data[5]+' Menit', wrap: wrap);
        // Panjang Area Survey
        excel.updateCell(sheet, packageExcel.CellIndex.indexByString("A6"),
"Panjang Area", wrap: wrap);
        excel.updateCell(sheet, packageExcel.CellIndex.indexByString("B6"),
widget.data[6]+' Meter', wrap: wrap);
        // Tabel Kecepatan
        excel.updateCell(sheet, packageExcel.CellIndex.indexByString("A8"),
"Menit", wrap: wrap);
        excel.updateCell(sheet, packageExcel.CellIndex.indexByString("B8"),
"Kecepatan Rata-rata(km/jam)", wrap: wrap);
        String fileName = 'Survey Kecepatan - '+widget.data[3]+' '+widget.data[4];
        String outputFile = "/storage/emulated/0/Download/Data
Survey/Kecepatan/$fileName.xlsx";
        int i = 9;

```

```

_speedSurveyList.forEach((f) {
    excel.updateCell(sheet, packageExcel.CellIndex.indexByString("A$i"),
f.menit, wrap: wrap);
    excel.updateCell(sheet, packageExcel.CellIndex.indexByString("B$i"),
f.average, wrap: wrap);
    i++;
});
excel.encode().then((onValue) {
    File(path.join(outputFile))
        ..createSync(recursive: true)
        ..writeAsBytesSync(onValue);
});
Future.delayed(Duration(milliseconds: 200), () {
    final snackBar = SnackBar(
        content: Text('Berhasil menyimpan data survey.'),
        action: SnackBarAction(
            label: 'OK',
            onPressed: () {
                _key.currentState.hideCurrentSnackBar();
            },
        ),
    );
    _key.currentState.showSnackBar(snackBar);
});
}

```

```

calculateData() {
    if (_waktuKendaraan.length > 0) {
//    print(_waktuKendaraan);
        _waktuKendaraan.forEach((f) {
            // m/s

```

```

List split1 = (int.parse(widget.data[6]) / f).toString().split('.');
String ms1 = split1[0];
// print(split1[1]);
String ms2 = split1[1].toString().length > 3 ? split1[1].substring(0,3) :
split1[1];
_kecepatanMS.add('$ms1.$ms2');
// km/jam
List split2 = (double.parse('$ms1.$ms2') * 3.6).toString().split('.');
int kmJam1 = int.parse(split2[0]);
int kmJam2 = int.parse(split2[1].toString().length > 3 ?
split2[1].substring(0,3) : split2[1]);
_kecepatanKmJam.add('$kmJam1.$kmJam2');
});
// print(_kecepatanMS);
// print(_kecepatanKmJam);
var average = _kecepatanKmJam.map((m) => double.parse(m)).reduce((a,
b) => a + b) / _kecepatanKmJam.length;
// print(average);
List split3 = average.toString().split('.');
String avg1 = split3[0];
String avg2 = split3[1].toString().length > 3 ? split3[1].substring(0,3) :
split3[1];
_speedSurveyList.add(SpeedSurveyList(menit: _speedSurveyList.length
== 0 ? 1 : _speedSurveyList[_speedSurveyList.length - 1].menit + 1, average:
double.parse('$avg1.$avg2')));
_waktuKendaraan.clear();
// print(_waktuKendaraan);
_kecepatanMS.clear();
_kecepatanKmJam.clear();
Future.delayed(Duration(milliseconds: 200), () {
setState() {

```



```

        _scrollController.animateTo(_scrollController.position.maxScrollExtent,
duration: Duration(milliseconds: 500), curve: Curves.ease);
    });
    });
} else {
    _speedSurveyList.add(SpeedSurveyList(menit: _speedSurveyList.length
== 0 ? 1 : _speedSurveyList[_speedSurveyList.length - 1].menit + 1, average:
0));
    _waktuKendaraan.clear();
    _kecepatanMS.clear();
    _kecepatanKmJam.clear();
    Future.delayed(Duration(milliseconds: 200), () {
        setState(() {
            _scrollController.animateTo(_scrollController.position.maxScrollExtent,
duration: Duration(milliseconds: 500), curve: Curves.ease);
        });
    });
}
}

buttonPressed() async {
    if (await Vibration.hasVibrator()) {
        Vibration.vibrate(duration: 300);
    }
    if (_startStatus) {
        _stopwatch.reset();
        _stopwatch.start();
        setState(() {
            _stopWatchTimer.onExecute.add(StopWatchExecute.start);
        });
    } else {

```

```

        _stopwatch.stop();
        setState(() {
            _stopWatchTimer.onExecute.add(StopWatchExecute.reset);
        });
        DateTime format =
DateTime.fromMillisecondsSinceEpoch(_stopwatch.elapsed.inMilliseconds);

        _waktuKendaraan.add(double.parse(format.second.toString()+ '.' +format.millis
econd.toString()));
        print(format.second.toString()+ '.' +format.millisecond.toString());
    }
    setState(() => _startStatus = !_startStatus);
}
}

class Countdown extends AnimatedWidget {
    Countdown({Key key, this.animation}) : super(key: key, listenable:
animation);
    final Animation<int> animation;

    @override
    build(BuildContext context) {
        Duration clockTimer = Duration(seconds: animation.value);

        String hour = clockTimer.inHours.toString();
        String minutes = clockTimer.inMinutes.remainder(60).toString().padLeft(2,
'0');
        String seconds = (clockTimer.inSeconds.remainder(60) %
60).toString().padLeft(2, '0');

```

```

if (firstTimerStatus == 1) {
    firstTimerStatus = 2;
} else if (firstTimerStatus == 2) {
    if (int.parse(seconds) == 0) {
        setStatus(false, '$hour:$minutes:$seconds');
    }
    if (int.parse(hour) == 0 && int.parse(minutes) == 0 && int.parse(seconds)
== 0) {
        setStatus(true, '$hour:$minutes:$seconds');
    }
} else {
    firstTimerStatus = 1;
}

return Text(
    '$hour:$minutes:$seconds',
    style: TextStyle(
        fontSize: 30.0,
        color: Theme.of(context).primaryColor,
    ),
);
}
}

class SpeedSurveyList {
    int menit;
    double average;
    SpeedSurveyList({this.menit, this.average});
}

```

4. Tampilan Survei Volume

```

class FragmentSurveyVolume extends StatefulWidget {
  @override
  _FragmentSurveyVolumeState createState() =>
  _FragmentSurveyVolumeState();
}

class _FragmentSurveyVolumeState extends State<FragmentSurveyVolume> {

  final GlobalKey<RefreshIndicatorState> _refreshIndicatorKey = new
  GlobalKey<RefreshIndicatorState>();

  List<FileSystemEntity> listFile = new List();

  bool _firstLoad = true;

  @override
  void initState() {
    checkFolder();
    initGetListofFile();
    super.initState();
  }

  @override
  void dispose() {
    // TODO: implement dispose
    super.dispose();
  }

  checkFolder() async {

```

```

final directory = new Directory('/storage/emulated/0/Download/Data
Survey/Volume');
if (!await directory.exists()) {
  directory.create();
}
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    body: body(),
  );
}

Widget body() {
  return Column(
    children: <Widget>[
      Expanded(
        child: RefreshIndicator(
          key: _refreshIndicatorKey,
          onRefresh: refresh,
          child: _firstLoad ? Container() :
listFile.length > 0 ? ListView.separated(
  separatorBuilder: (context, index) => Divider(
    color: Colors.grey.withOpacity(0.3),
    height: 1.0,
  ),
  itemCount: listFile.length,
  itemBuilder: (context, index) {
    return ListTile(
      title: Text(path.basename(listFile[index].path)),

```

```

        onTap: () => false,
      );
    },
  ):
  Center(
    child: MaterialButton(
      splashColor: Colors.transparent,
      hoverColor: Colors.transparent,
      highlightColor: Colors.transparent,
      onPressed: () => refresh(),
      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        crossAxisAlignment: CrossAxisAlignment.center,
        mainAxisAlignment: MainAxisAlignment.min,
        children: <Widget>[
          Icon(Icons.insert_drive_file, color: Color(0xFFCFCFCF), size:
85.0),
          Padding(
            padding: EdgeInsets.symmetric(vertical: 10.0, horizontal: 10.0),
            child: Text('Tidak ada data', style: TextStyle(color:
Color(0xFF707070), fontSize: 17.0, fontWeight: FontWeight.bold)),
          ),
          Text('Ketuk untuk refresh', style: TextStyle(color:
Color(0xFF9E9E9E), fontSize: 16.0, fontWeight: FontWeight.w300)),
        ],
      ),
    ),
  ),
),
),
),
),
),
),
),
Container(

```

```

height: 50.0,
width: MediaQuery.of(context).size.width,
child: RaisedButton(
  color: Colors.blue,
  elevation: 0.0,
  shape: RoundedRectangleBorder(
    borderRadius: BorderRadius.circular(0.0),
  ),
  onPressed: () => Navigator.push(context, MaterialPageRoute(builder:
(context) => ViewInputDataPreparation())),
  child: Text('TAMBAH SURVEY', style: TextStyle(color: Colors.white,
fontSize: 16.0)),
),
),
],
);
}

```

```

initGetListofFile() {
  getListofFile().then((List list) {
    _firstLoad = false;
    setState(() => listFile = list);
  }).catchError((onError) {
    print('onError : $onError');
  });
}

```

```

Future refresh() async {
  await getListofFile().then((List list) {
    setState(() => listFile = list);
  }).catchError((onError) {

```

```

    print('onError : $onError');
  });
}

```

```

Future<List> getListofFile() async {
  final directory = new Directory('/storage/emulated/0/Download/Data
Survey/Volume');
  List list;
  list = directory.listFiles(recursive: true, followLinks: false);
  List data = list.reversed.toList();
  return data;
}
}

```

5. Proses Perhitungan Survei Volume

```
Function setStatus;
```

```
int firstTimerStatus = 0;
```

```

class ViewInputDataSurveyVolume extends StatefulWidget {
  final List data;
  ViewInputDataSurveyVolume({ this.data });
  @override
  _ViewInputDataSurveyVolumeState createState() =>
  _ViewInputDataSurveyVolumeState();
}

```

```

class _ViewInputDataSurveyVolumeState extends
State<ViewInputDataSurveyVolume> with TickerProviderStateMixin {

```

```

  final GlobalKey<ScaffoldState> _key = new GlobalKey<ScaffoldState>();

```



```

AnimationController _controller;
ScrollController _scrollController = new ScrollController();

bool _initStartStatus = true;
int _jumlahLV = 0;
int _jumlahHV = 0;
int _jumlahMC = 0;
List<VolumeSurveyList> _volumeSurveyList = new List();

var timer;

@override
void initState() {
  print(widget.data);
  _controller = AnimationController(vsync: this, duration: Duration(minutes:
int.parse(widget.data[5])));
  setStatus = checkStatus;
  firstTimerStatus = 0;
  super.initState();
}

@override
void dispose() {
  _controller.stop(canceled: true);
  _scrollController.dispose();
  if (timer != null && timer.isActive) {
    timer.cancel();
  }
  super.dispose();
}

```

```

@override
Widget build(BuildContext context) {
  return Scaffold(
    key: _key,
    appBar: AppBar(
      title: Text('Input Data Survey Volume'),
    ),
    body: body(),
  );
}

Widget body() {
  return Container(
    padding: const EdgeInsets.symmetric(vertical: 20.0, horizontal: 10.0),
    child: Column(
      children: <Widget>[
        Container(
          child: Column(
            children: <Widget>[
              Countdown(
                animation: StepTween(
                  begin: int.parse(widget.data[5]) * 60,
                  end: 0,
                ).animate(_controller)
              ),
            ],
          ),
          Container(
            height: 40.0,
            width: MediaQuery.of(context).size.width,

```

```

margin: const EdgeInsets.symmetric(vertical: 15.0),
child: RaisedButton(
  elevation: _initStartStatus ? 2.0 : 0.0,
  color: _initStartStatus ? Colors.blue : Colors.grey,
  onPressed: () {
    if (_initStartStatus) {
      setState(() => _initStartStatus = false);
      _controller.forward();
    }
  },
  child: Text('START', style: TextStyle(color: Colors.white)),
),
),
],
)
),
Container(
  padding: const EdgeInsets.symmetric(vertical: 5.0, horizontal: 5.0),
  alignment: Alignment.centerLeft,
  child: Wrap(
    children: <Widget>[
      Text(widget.data[0]+' | '+widget.data[1]+' | '+widget.data[2]+' |
'+widget.data[3]+' | '+widget.data[4]+' | '+widget.data[5]+' Menit'),
    ],
  ),
),
Expanded(
  child: Container(
    width: MediaQuery.of(context).size.width,
    child: SingleChildScrollView(

```

```
    controller: _scrollController,  
    physics: BouncingScrollPhysics(),  
    child: DataTable(  
      columns: <DataColumn>[  
        DataColumn(label: Text("Menit")),  
        DataColumn(label: Text("Volume(smp/jam)")),  
      ],  
      rows: _volumeSurveyList.map((VolumeSurveyList list) {  
        return DataRow(  
          cells: <DataCell>[  
            DataColumn(Text(list.menit.toString())),  
            DataColumn(Text(list.volume.toString())),  
          ],  
        );  
      }).toList()  
    ),  
  ),  
),  
Container(  
  padding: const EdgeInsets.only(top: 25.0),  
  child: Column(  
    mainAxisAlignment: MainAxisAlignment.end,  
    children: <Widget>[  
      Row(  
        children: <Widget>[  
          lvButton(),  
          hvButton(),  
          mcButton(),
```

```

        ],
      ),
    ],
  )
),
],
),
);
}

lvButton() {
  if (widget.data[6][0]['LV']) {
    return Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 5.0),
        child: RaisedButton(
          padding: const EdgeInsets.symmetric(vertical: 40.0, horizontal: 3.0),
          color: !_initStartStatus || isDoneStatus ? Colors.grey : Colors.blue,
          onPressed: () => !_initStartStatus && !isDoneStatus ?
lvButtonPressed() : false,
          child: Column(
            children: <Widget>[
              Text(_jumlahLV.toString(), style: TextStyle(color: Colors.white,
fontSize: 18.0)),
              Text('LV', style: TextStyle(color: Colors.white, fontSize: 30.0)),
              SizedBox(height: 5.0),
              Text('Kendaraan Ringan', style: TextStyle(color: Colors.white,
fontSize: 14.0)),
            ],
          ),
        ),
      ),
    ],
  ),
),
);
}

```

```

    ),
  ),
);
} else {
  return Container();
}
}

hvButton() {
  if (widget.data[6][1]['HV']) {
    return Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 5.0),
        child: RaisedButton(
          padding: const EdgeInsets.symmetric(vertical: 40.0, horizontal: 3.0),
          color: !_initStartStatus || isDoneStatus ? Colors.grey : Colors.blue,
          onPressed: () => !_initStartStatus && !isDoneStatus ?
hvButtonPressed() : false,
          child: Column(
            children: <Widget>[
              Text(_jumlahHV.toString(), style: TextStyle(color: Colors.white,
fontSize: 18.0)),
              Text('HV', style: TextStyle(color: Colors.white, fontSize: 30.0)),
              SizedBox(height: 5.0),
              Text('Kendaraan Berat', style: TextStyle(color: Colors.white,
fontSize: 14.0)),
            ],
          ),
        ),
      ),
    ),
  ),
)

```

```

    );
  } else {
    return Container();
  }
}

mcButton() {
  if (widget.data[6][2]['MC']) {
    return Expanded(
      child: Padding(
        padding: const EdgeInsets.symmetric(horizontal: 5.0),
        child: RaisedButton(
          padding: const EdgeInsets.symmetric(vertical: 40.0, horizontal: 3.0),
          color: !_initStartStatus || isDoneStatus ? Colors.grey : Colors.blue,
          onPressed: () => !_initStartStatus && !isDoneStatus ?
mcButtonPressed() : false,
          child: Column(
            children: <Widget>[
              Text(_jumlahMC.toString(), style: TextStyle(color: Colors.white,
fontSize: 18.0)),
              Text('MC', style: TextStyle(color: Colors.white, fontSize: 30.0)),
              SizedBox(height: 5.0),
              Text('Sepeda Motor', style: TextStyle(color: Colors.white, fontSize:
14.0)),
            ],
          )
        ),
      ),
    );
  } else {

```

```
        return Container();
    }
}

lvButtonPressed() async {
    if (await Vibration.hasVibrator()) {
        Vibration.vibrate(duration: 300);
    }
    setState() {
        _jumlahLV += 1;
    });
}

hvButtonPressed() async {
    if (await Vibration.hasVibrator()) {
        Vibration.vibrate(duration: 300);
    }
    setState() {
        _jumlahHV += 1;
    });
}

mcButtonPressed() async {
    if (await Vibration.hasVibrator()) {
        Vibration.vibrate(duration: 300);
    }
    setState() {
        _jumlahMC += 1;
    });
}
```



```

}

String currentTime = "";
bool isDoneStatus = false;

checkStatus(bool isDone, String time) {
  if (!isDoneStatus && isDone) {
    isDoneStatus = true;
    print('Done');
    print(_volumeSurveyList.length);
    Future.delayed(Duration(milliseconds: 200), () {
      setState(() => isDoneStatus);
    });
    createExcelFile();
  } else if (!isDoneStatus && !isDone) {
    if (currentTime.isEmpty) {
      currentTime = time;
      calculateData();
    } else {
      if (currentTime != time) {
        currentTime = time;
        calculateData();
      }
    }
  }
}

createExcelFile() {
  var excel = packageExcel.Excel.createExcel();

```

```

var sheet = 'Sheet1';

packageExcel.TextWrapping wrap =
packageExcel.TextWrapping.WrapText;

// Nama Surveyor

excel.updateCell(sheet, packageExcel.CellIndex.indexByString("A1"),
"Nama Surveyor", wrap: wrap);

excel.updateCell(sheet, packageExcel.CellIndex.indexByString("B1"),
widget.data[0], wrap: wrap);

// Lokasi Survey

excel.updateCell(sheet, packageExcel.CellIndex.indexByString("A2"),
"Lokasi Survey", wrap: wrap);

excel.updateCell(sheet, packageExcel.CellIndex.indexByString("B2"),
widget.data[1], wrap: wrap);

// Arah Survey

excel.updateCell(sheet, packageExcel.CellIndex.indexByString("A3"),
"Arah Survey", wrap: wrap);

excel.updateCell(sheet, packageExcel.CellIndex.indexByString("B3"),
widget.data[2], wrap: wrap);

// Tanggal & Waktu Survey

excel.updateCell(sheet, packageExcel.CellIndex.indexByString("A4"),
"Tanggal & Waktu Survey", wrap: wrap);

excel.updateCell(sheet, packageExcel.CellIndex.indexByString("B4"),
widget.data[3]+' '+widget.data[4], wrap: wrap);

// Jangka Waktu Survey

excel.updateCell(sheet, packageExcel.CellIndex.indexByString("A5"),
"Jangka Waktu", wrap: wrap);

excel.updateCell(sheet, packageExcel.CellIndex.indexByString("B5"),
widget.data[5]+' Menit', wrap: wrap);

// Tabel Kecepatan

excel.updateCell(sheet, packageExcel.CellIndex.indexByString("A7"),
"Menit", wrap: wrap);

excel.updateCell(sheet, packageExcel.CellIndex.indexByString("B7"),
"Volume(smp/jam)", wrap: wrap);

String fileName = 'Survey Volume - '+widget.data[3]+' '+widget.data[4];

```

```

String outputFile = "/storage/emulated/0/Download/Data
Survey/Volume/$fileName.xlsx";

int i = 8;

_volumeSurveyList.forEach((f) {

    excel.updateCell(sheet, packageExcel.CellIndex.indexByString("A$i"),
f.menit, wrap: wrap);

    excel.updateCell(sheet, packageExcel.CellIndex.indexByString("B$i"),
f.volume, wrap: wrap);

    i++;
});

excel.encode().then((onValue) {
    File(path.join(outputFile))
        ..createSync(recursive: true)
        ..writeAsBytesSync(onValue);
});

Future.delayed(Duration(milliseconds: 200), () {
    final snackBar = SnackBar(
        content: Text('Berhasil menyimpan data survey.'),
        action: SnackBarAction(
            label: 'OK',
            onPressed: () {
                _key.currentState.hideCurrentSnackBar();
            },
        ),
    );

    _key.currentState.showSnackBar(snackBar);
});
}

calculateData() {

```

```

// print(_jumlahLV);
// print(_jumlahHV);
// print(_jumlahMC);
if (_jumlahLV != 0 || _jumlahHV != 0 || _jumlahMC != 0) {
    double _smpMenit = 0;
    if (widget.data[6][0]['EMP'] > 0.0) {
        _smpMenit = _smpMenit + (_jumlahLV * widget.data[6][0]['EMP']);
    }
    if (widget.data[6][1]['EMP'] > 0.0) {
        _smpMenit = _smpMenit + (_jumlahHV * widget.data[6][1]['EMP']);
    }
    if (widget.data[6][2]['EMP'] > 0.0) {
        _smpMenit = _smpMenit + (_jumlahMC * widget.data[6][2]['EMP']);
    }
    double _smpJam = _smpMenit * 60;
    _volumeSurveyList.add(VolumeSurveyList(menit:
    _volumeSurveyList.length == 0 ? 1 :
    _volumeSurveyList[_volumeSurveyList.length - 1].menit + 1, volume:
    _smpJam.toInt()));
    _jumlahLV = 0;
    _jumlahHV = 0;
    _jumlahMC = 0;
    Future.delayed(Duration(milliseconds: 200), () {
        setState() {
            _scrollController.animateTo(_scrollController.position.maxScrollExtent,
            duration: Duration(milliseconds: 500), curve: Curves.ease);
        });
    });
} else {
    _volumeSurveyList.add(VolumeSurveyList(menit:
    _volumeSurveyList.length == 0 ? 1 :
    _volumeSurveyList[_volumeSurveyList.length - 1].menit + 1, volume: 0));
}

```

```

    _jumlahLV = 0;
    _jumlahHV = 0;
    _jumlahMC = 0;
    Future.delayed(Duration(milliseconds: 200), () {
      setState() {
        _scrollController.animateTo(_scrollController.position.maxScrollExtent,
duration: Duration(milliseconds: 500), curve: Curves.ease);
      });
    });
  }
}
}
}

```

```

class Countdown extends AnimatedWidget {
  Countdown({Key key, this.animation}) : super(key: key, listenable:
animation);
  final Animation<int> animation;

  @override
  build(BuildContext context) {
    Duration clockTimer = Duration(seconds: animation.value);

    String hour = clockTimer.inHours.toString();

    String minutes = clockTimer.inMinutes.remainder(60).toString().padLeft(2,
'0');

    String seconds = (clockTimer.inSeconds.remainder(60) %
60).toString().padLeft(2, '0');

    if (firstTimerStatus == 1) {
      firstTimerStatus = 2;
    }
  }
}

```

```

} else if (firstTimerStatus == 2) {
    if (int.parse(seconds) == 0) {
        setStatus(false, '$hour:$minutes:$seconds');
    }
    if (int.parse(hour) == 0 && int.parse(minutes) == 0 && int.parse(seconds)
== 0) {
        setStatus(true, '$hour:$minutes:$seconds');
    }
} else {
    firstTimerStatus = 1;
}

return Text(
    '$hour:$minutes:$seconds',
    style: TextStyle(
        fontSize: 30.0,
        color: Theme.of(context).primaryColor,
    ),
);
}
}

```

```

class VolumeSurveyList {
    int menit;
    int volume;
    VolumeSurveyList({this.menit, this.volume});
}

```

6. Proses Perhitungan Kepadatan

```

class FragmentHitungKepadatan extends StatefulWidget {
    @override

```

```

    _FragmentHitungKepadatanState createState() =>
    _FragmentHitungKepadatanState();
}

```

```

class _FragmentHitungKepadatanState extends
State<FragmentHitungKepadatan> {

    final GlobalKey<RefreshIndicatorState> _refreshIndicatorKey = new
    GlobalKey<RefreshIndicatorState>();

    final GlobalKey<ScaffoldState> _key = new GlobalKey<ScaffoldState>();

    File _speedFile;
    File _volumeFile;

    final _speedController = new TextEditingController();
    final _volumeController = new TextEditingController();

    @override
    void initState() {
        // TODO: implement initState
        super.initState();
    }

    @override
    void dispose() {
        _speedController.dispose();
        _volumeController.dispose();
        super.dispose();
    }
}

```

```

getSpeedFile() async {
  _speedFile = await FilePicker.getFile(
    type: FileType.custom,
    allowedExtensions: ['xlsx']
  );
  _speedController.text = path.basename(_speedFile.path);
}

getVolumeFile() async {
  _volumeFile = await FilePicker.getFile(
    type: FileType.custom,
    allowedExtensions: ['xlsx']
  );
  _volumeController.text = path.basename(_volumeFile.path);
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    key: _key,
    body: body(),
  );
}

Widget body() {
  return Column(
    children: <Widget>[
      Expanded(

```



```

child: RefreshIndicator(
  key: _refreshIndicatorKey,
  onRefresh: refresh,
  child: SingleChildScrollView(
    padding: const EdgeInsets.symmetric(vertical: 20.0, horizontal: 20.0),
    child: Column(
      children: <Widget>[
        Row(
          children: <Widget>[
            Expanded(
              child: textFieldCustom(_speedController)
            ),
            Container(
              padding: const EdgeInsets.only(left: 10.0),
              child: RaisedButton(
                color: Colors.green,
                child: Text('Browse', style: TextStyle(color: Colors.white)),
                onPressed: () => getSpeedFile()
              ),
            ),
          ],
        ),
        Row(
          children: <Widget>[
            Expanded(
              child: textFieldCustom(_volumeController)
            ),
            Container(
              padding: const EdgeInsets.only(left: 10.0),

```

```

child: RaisedButton(
  color: Colors.green,
  child: Text('Browse', style: TextStyle(color: Colors.white)),
  onPressed: () => getVolumeFile()
),
),
),
),
),
),
),
),
),
),
),
),
),
),
),
),
Container(
  height: 50.0,
  width: MediaQuery.of(context).size.width,
  child: RaisedButton(
    color: Colors.blue,
    elevation: 0.0,
    shape: RoundedRectangleBorder(
      borderRadius: BorderRadius.circular(0.0),
    ),
    onPressed: () {
      if (_speedFile != null && _volumeFile != null) {
        List<KepadatanSurveyList> listKepadatan = [];
//      print(_speedFile);
//      print(_volumeFile);
        List listKecepatan = [];
        List listVolume = [];

```

```

listKecepatan.clear();
listVolume.clear();
listKepadatan.clear();
// Process Excel Speed
var bytes1 = File(_speedFile.path).readAsBytesSync();
var excel1 = Excel.decodeBytes(bytes1, update: true);
for (var table in excel1.tables.keys) {
    // Nama Surveyor
    listKecepatan.add(excel1.tables[table].rows[0][1]);
    listKecepatan.add(excel1.tables[table].rows[1][1]);
    listKecepatan.add(excel1.tables[table].rows[2][1]);
    listKecepatan.add(excel1.tables[table].rows[3][1]);
    listKecepatan.add(excel1.tables[table].rows[4][1]);
    listKecepatan.add(excel1.tables[table].rows[5][1]);
//    print(excel1.tables[table].rows.toList());
    print(listKecepatan);
//    for(int i = 8; i < excel1.tables[table].maxRows; i++) {
//        listKecepatan.add(excel1.tables[table].rows[i][1]);
//    }
}

// Process Excel Volume
var bytes2 = File(_volumeFile.path).readAsBytesSync();
var excel2 = Excel.decodeBytes(bytes2, update: true);
for (var table in excel2.tables.keys) {
    // Nama Surveyor
    listVolume.add(excel2.tables[table].rows[0][1]);
    listVolume.add(excel2.tables[table].rows[1][1]);
    listVolume.add(excel2.tables[table].rows[2][1]);
    listVolume.add(excel2.tables[table].rows[3][1]);

```

```

listVolume.add(excel2.tables[table].rows[4][1]);
//    print(excel2.tables[table].rows.toList());
print(listVolume);
//    for(int i = 8; i < excel2.tables[table].maxRows; i++) {
//        listVolume.add(excel2.tables[table].rows[i][1]);
//    }
// Hitung Kepadatan
int a = 1;
for(int i = 8; i <= excel2.tables[table].maxRows; i++) {
    double speedValue = double.parse(excel1.tables[table].rows[i][1]);
    int volumeValue = int.parse(excel2.tables[table].rows[i-1][1]);
    double kepadatanValue = speedValue == 0 ?
double.parse(volumeValue.toString()) : volumeValue / speedValue;
    List split1 = kepadatanValue.toString().split('.');
    String kpd1 = split1[0];
//    print(kepadatanValue.toString()+ ' = '+volumeValue.toString()+ ' /
'+speedValue.toString());
    String kpd2 = split1[1].toString().length > 3 ?
split1[1].substring(0,3) : split1[1];
    listKepadatan.add(KepadatanSurveyList(menit: a, speed:
speedValue, volume: volumeValue, kepadatan: double.parse('$kpd1.$kpd2')));
    a++;
}
}
// Create File Excel
var excel = Excel.createExcel();
var sheet = 'Sheet1';
TextWrapping wrap = TextWrapping.WrapText;
// Nama Surveyor
excel.updateCell(sheet, CellIndex.indexByString("A1"), "Nama
Surveyor", wrap: wrap);

```

```

        excel.updateCell(sheet, CellIndex.indexByString("B1"),
listKecepatan[0]+' - '+listVolume[0], wrap: wrap);

        // Lokasi Survey

        excel.updateCell(sheet, CellIndex.indexByString("A2"), "Lokasi
Survey", wrap: wrap);

        excel.updateCell(sheet, CellIndex.indexByString("B2"),
listKecepatan[1], wrap: wrap);

        // Arah Survey

        excel.updateCell(sheet, CellIndex.indexByString("A3"), "Arah
Survey", wrap: wrap);

        excel.updateCell(sheet, CellIndex.indexByString("B3"),
listKecepatan[2], wrap: wrap);

        // Tanggal & Waktu Survey

        excel.updateCell(sheet, CellIndex.indexByString("A4"), "Tanggal &
Waktu Survey", wrap: wrap);

        excel.updateCell(sheet, CellIndex.indexByString("B4"),
listKecepatan[3], wrap: wrap);

        // Jangka Waktu Survey

        excel.updateCell(sheet, CellIndex.indexByString("A5"), "Jangka
Waktu", wrap: wrap);

        excel.updateCell(sheet, CellIndex.indexByString("B5"),
listKecepatan[4], wrap: wrap);

        // Panjang Area Survey

        excel.updateCell(sheet, CellIndex.indexByString("A6"), "Panjang
Area", wrap: wrap);

        excel.updateCell(sheet, CellIndex.indexByString("B6"),
listKecepatan[5], wrap: wrap);

        // Tabel Kepadatan

        excel.updateCell(sheet, CellIndex.indexByString("A8"), "Menit",
wrap: wrap);

        excel.updateCell(sheet, CellIndex.indexByString("B8"), "Kecepatan
(km/jam)", wrap: wrap);

        excel.updateCell(sheet, CellIndex.indexByString("C8"), "Volume
(smp/jam)", wrap: wrap);

```



```

        ),
      );
      _key.currentState.showSnackBar(snackBar);
    });
  }
},
child: Text('EXPORT SURVEY', style: TextStyle(color: Colors.white,
fontSize: 16.0)),
),
),
],
);
}

Future refresh() async {
  setState(() {
    _speedFile = null;
    _volumeFile = null;
  });
}

Widget textFieldCustom(TextEditingController controller) {
  return Container(
    decoration: BoxDecoration(
      color: Colors.grey.shade300,
      borderRadius: BorderRadius.circular(3.0),
    ),
    margin: const EdgeInsets.symmetric(vertical: 6.0),
    child: GestureDetector(
      onTap: () => false,

```

```

child: AbsorbPointer(
  absorbing: true,
  child: TextFormField(
    maxLines: null,
    controller: controller,
    decoration: InputDecoration(
      enabled: false,
      contentPadding: const EdgeInsets.symmetric(vertical: 10.0,
horizontal: 15.0),
      isDense: true,
      border: InputBorder.none,
      hintStyle: TextStyle(color: Colors.grey),
      labelText: "
    ),
  ),
),
),
),
);
}
}

```

```

class KepadatanSurveyList {
  int menit;
  double speed;
  int volume;
  double kepadatan;
  KepadatanSurveyList({this.menit, this.speed, this.volume, this.kepadatan});
}

```