

BAB II

LANDASAN TEORI

2.1 Profil Perusahaan

Profil perusahaan merupakan penjelasan secara rinci tentang perusahaan tersebut, di dalamnya terdapat sejarah singkat perusahaan, tugas pokok, visi dan misi serta struktur organisasi perusahaan.

2.1.1 Sejarah Singkat Dinas Perhubungan Kabupaten Kampar

Dinas Perhubungan Kabupaten Kampar dibentuk berdasarkan Peraturan Daerah Kabupaten Kampar Nomor 6 Tahun 2008 tanggal 15 Februari 2008 dan diperbaharui dengan Peraturan Daerah Kabupaten Kampar Nomor 6 Tahun 2012 tanggal 12 Juni 2012. Pada awal pembentukannya, Dinas Perhubungan Kabupaten Kampar bergabung dengan Dinas Informasi dan Komunikasi, dengan nama Dinas Perhubungan, Informasi dan Komunikasi.

Peraturan Daerah yang berada di bawah pengelolaan Dinas Perhubungan Kabupaten Kampar yang masih berlaku sampai saat ini, yaitu:

1. Peraturan Daerah Kabupaten Kampar Nomor 3 Tahun 2009 tentang Nama Jalan dalam Kota Bangkinang.
2. Peraturan Daerah Kabupaten Kampar Nomor 13 Tahun 2011 tentang Pajak Parkir.
3. Peraturan Daerah Kabupaten Kampar Nomor 07 Tahun 2012 tentang Retribusi Jasa Umum, Retribusi Pengujian Kendaraan Bermotor dan Retribusi Parkir.
4. Peraturan Daerah Kabupaten Kampar Nomor 08 Tahun 2012 tentang Retribusi Jasa Usaha, Retribusi Terminal dan Retribusi Pemakaian Kekayaan Daerah.
5. Peraturan Pemerintah Daerah Kabupaten Kampar Nomor 09 Tahun 2012 tentang Retribusi Perizinan Tertentu dan Retribusi Izin Trayek.

2.1.2 Tugas Pokok dan Fungsi

Dinas Perhubungan Kabupaten Kampar memiliki tugas pokok dan fungsi berdasarkan Peraturan Daerah Kabupaten Kampar Nomor 6 Tahun 2008 dan Peraturan Bupati Kampar Nomor 35 Tahun 2008 tentang uraian tugas jabatan struktural Dinas Perhubungan Informasi dan Komunikasi Kabupaten Kampar. Tugas pokok Dinas Perhubungan Kabupaten Kampar adalah melaksanakan kewenangan Pemerintah Kabupaten Kampar di Bidang Perhubungan. Untuk melaksanakan tugas tersebut Dinas Perhubungan Kabupaten Kampar memiliki fungsi, yaitu:

1. Perumusan kebijakan teknis bidang Perhubungan.
2. Pelaksanaan kebijakan teknis bidang Perhubungan.
3. Pelaksanaan evaluasi dan pelaporan sesuai dengan lingkup tugasnya.
4. Pelaksanaan administrasi Dinas Perhubungan.
5. Melakukan pembinaan terhadap Unit Pelaksana Teknis Dinas (UPTD).
6. Pelaksanaan fungsi lain yang diberikan oleh Bupati terkait dengan tugas dan fungsinya.

2.1.3 Visi dan Misi

Untuk menjaga kualitas pelayanan, Dinas Perhubungan Kabupaten Kampar memiliki Visi dan Misi sebagai berikut.

Visi

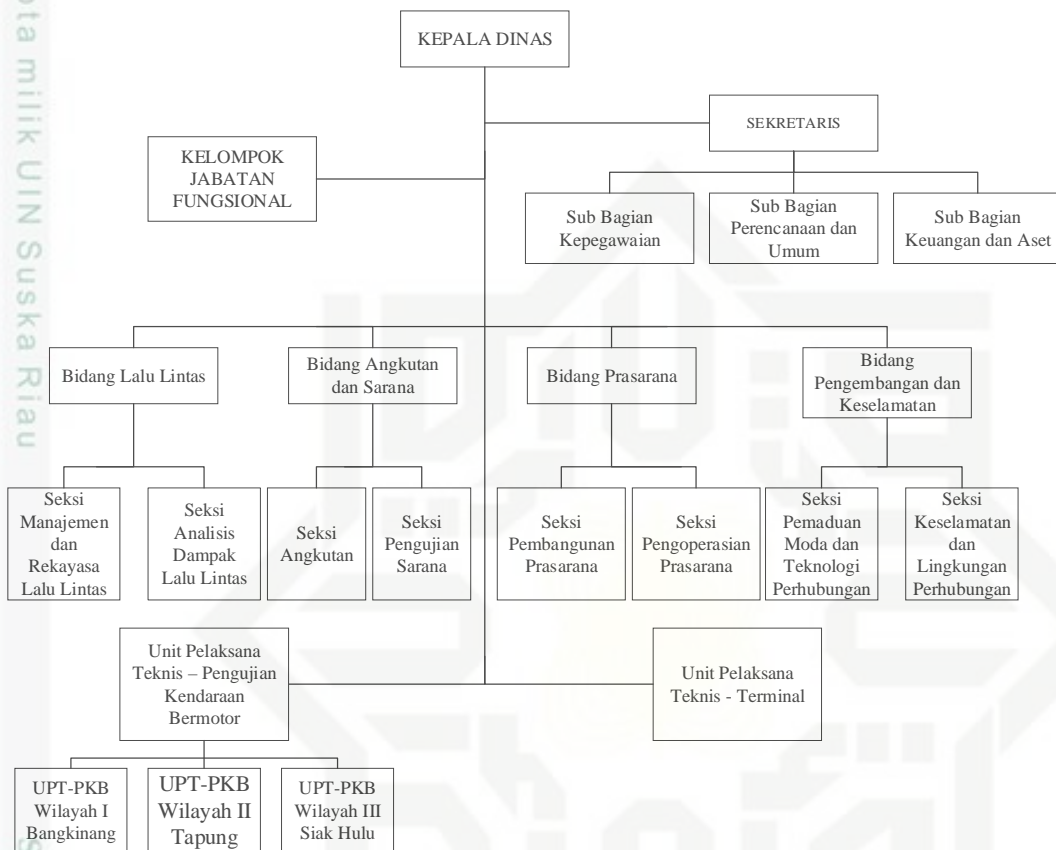
“Menjadi Institusi Penyelamat Transportasi dan Akses yang Prima”

Misi

1. Menata sistem transportasi yang selamat dan terjangkau.
2. Mengembangkan sarana dan prasarana perhubungan yang optimal.
3. Meningkatkan penegakan hukum yang berkeselamatan.
4. Menata sistem informasi dan komunikasi yang handal.

2.1.4 Struktur Organisasi Dinas Perhubungan Kabupaten Kampar

Berikut ini struktur organisasi Dinas Perhubungan Kabupaten Kampar dapat dilihat pada Gambar 2.1.



Gambar 2.1 Struktur Organisasi Dinas Perhubungan Kabupaten Kampar

2.2 Sistem Informasi Geografis (SIG)

Menurut ESRI dalam Prahasta (2009), SIG adalah kumpulan yang terorganisasi dari perangkat keras komputer, perangkat lunak, data geografi dan personil yang dirancang secara efisien untuk memperoleh, menyimpan, memperbarui, memanipulasi, menganalisis dan menampilkan semua bentuk informasi yang bereferensi geografis.

Pengertian SIG serupa juga dijelaskan menurut Irwansyah (2013), SIG atau *Geographic Information System* (GIS) adalah sebuah sistem yang didesain untuk menangkap, menyimpan, memanipulasi, menganalisa, mengatur dan menampilkan seluruh jenis data geografis. Sedangkan pengertian SIG menurut

Budiyanto (2016) memberikan tambahan data geografis yang dapat diolah menggunakan SIG yaitu, perencanaan dan pengelolaan penggunaan lahan, sumber daya alam, lingkungan transportasi, fasilitas kota dan pelayanan umum lainnya.

Menurut Sasongko (2016), SIG saat ini dapat dimanfaatkan untuk mengetahui posisi dan wilayah secara visualisasi sehingga dapat memudahkan menentukan keputusan dengan cepat tanpa harus langsung *survey* turun ke lapangan. Selain itu menurut Dewi (2015), SIG telah menunjukkan manfaat di berbagai bidang seperti mitigasi bencana terkait banjir, berguna bagi pertanian, memantau tata kota, membaca tren, pariwisata dan merencanakan atau membuat tata kota.

2.3 Perusahaan Umum (PERUM) DAMRI

PERUM DAMRI adalah Perusahaan Umum yang berbadan hukum dengan tugas utama menyelenggarakan angkutan penumpang dan barang di atas jalan dengan menggunakan kendaraan bermotor. DAMRI adalah singkatan dari Djawatan Angkoetan Motor Repoeblik Indonesia (ER, EYD: Jawatan Angkutan Motor Republik Indonesia) yang dibentuk berdasarkan Maklumat Kementerian Perhubungan RI Nomor. 01/DAMRI/46 tanggal 25 November 1946 dengan tugas utama menyelenggarakan kendaraan bermotor. Dalam perkembangan selanjutnya sebagai PERUM, nama DAMRI tetap diabadikan sebagai *brand mark* dari Badan Usaha Milik Negara (BUMN) yang hingga saat ini masih tetap konsisten menjalankan tugasnya sebagai salah satu penyelenggara jasa angkutan penumpang barang dengan menggunakan bus dan truk. Hingga saat ini DAMRI memiliki jaringan pelayanan tersebar hampir di seluruh wilayah Republik Indonesia. Dalam kegiatan usahanya DAMRI menyelenggarakan pelayanan angkutan kota, angkutan antar kota dalam provinsi, angkutan kota antar provinsi, angkutan khusus bandar udara, angkutan pariwisata, angkutan logistik, angkutan keperintisan dan angkutan lintas batas negara.

Menurut Peraturan Pemerintah Republik Indonesia Nomor 31 Tahun 2002 pasal 6 tentang sifat, maksud dan tujuan PERUM DAMRI, maksud dan tujuan Perusahaan adalah menyelenggarakan usaha yang bertujuan untuk kemanfaatan

umum penyelenggaraan jasa angkutan umum, penumpang dan barang di atas jalan dengan kendaraan bermotor yang bermutu tinggi dengan memperoleh keuntungan sesuai dengan prinsip pengelolaan perusahaan. Selanjutnya dalam pasal 7 dijelaskan mengenai kegiatan dan pengembangan usaha yang dibagi menjadi tiga, yaitu:

1. Jasa angkutan penumpang untuk umum dan atau barang.
2. Angkutan perintis berdasarkan penugasan pemerintah.
3. Usaha-usaha lain yang dapat menunjang tercapainya maksud dan tujuan perusahaan.

Berdasarkan SK Dirjen 687/2002, mengatur lama perjalanan ke dan dari tempat tujuan setiap hari rata-rata satu sampai satu setengah jam, maksimum dua sampai tiga jam. Kecepatan perjalanan (kilometer per jam), adalah waktu yang dibutuhkan untuk menempuh satu kilometer dari panjang trayek, satuan yang digunakan menit per kilometer.

Berdasarkan hasil wawancara dengan Bapak Yurisdian, S.Si., ST., S.IPem., M.Eng., M.Si., Ph.D selaku Kepala Sub Bagian Perencanaan dan Evaluasi Dinas Perhubungan Kabupaten Kampar (Lampiran A) menjelaskan saat ini Kabupaten Kampar telah mengoperasikan bus DAMRI sebanyak 3 unit dari total 5 unit alokasi bantuan bus sedang perintis tahun anggaran 2015 untuk Provinsi Riau yang tertera pada Keputusan Menteri Perhubungan Nomor KP 697 Tahun 2015.

Tiga unit bus sedang perintis masing-masing dengan kode kendaraan 4856, 4857 yang telah melayani perjalanan dari Terminal Kampar menuju Bandara Internasional Sultan Syarif Kasim II Pekanbaru untuk transportasi DAMRI Bandara dengan tarif satu kali perjalanan sebesar Rp. 35.000,- per penumpang. Selain rute tersebut, bus DAMRI yang diawasi oleh Dinas Perhubungan Kabupaten Kampar dengan kode kendaraan 4858 juga melayani tujuan pariwisata yaitu Terminal Kampar menuju Candi Muara Takus dengan tarif untuk umum Rp. 10.000,- dan pelajar Rp. 5.000,-.

2.4 Trayek

Menurut Keputusan Menteri Perhubungan Nomor KM 35 Tahun 2003 tentang penyelenggaraan angkutan orang di jalan dengan kendaraan umum, pengertian trayek adalah lintasan kendaraan umum untuk pelayanan jasa angkutan orang dengan bus, yang mempunyai asal dan tujuan perjalanan tetap, lintasan tetap dan jadwal tetap maupun tidak terjadwal. Sedangkan pengertian Jaringan Trayek menurut Keputusan Menteri Perhubungan Nomor KM 35 Tahun 2003 tentang penyelenggaraan angkutan orang di jalan dengan kendaraan umum adalah kumpulan dari trayek-trayek yang menjadi satu kesatuan jaringan pelayanan angkutan orang. Jaringan trayek ditetapkan mengikuti dan searah dengan Jaringan Transportasi Jalan yang ditetapkan pada Pasal 6 undang-undang Nomor 14 Tahun 1992 tentang lalu lintas dan angkutan jalan, sebagai berikut:

1. Untuk mewujudkan lalu lintas dan angkutan jalan yang terpadu dengan moda transportasi lain ditetapkan jaringan transportasi jalan yang menghubungkan seluruh wilayah tanah air.
2. Penetapan jaringan transportasi jalan sebagaimana dimaksud dalam ayat (1) didasarkan pada kebutuhan transportasi, fungsi, peranan, kapasitas lalu lintas dan kelas jalan.

Pasal 7 Keputusan Menteri Perhubungan Nomor KM 35 Tahun 2003 tentang penyelenggaraan angkutan orang di jalan dengan kendaraan umum bahwa jaringan trayek, kebutuhan kendaraan dan evaluasi kebutuhan penambahan kendaraan untuk jaringan trayek melalui antar kabupaten atau kota dalam suatu daerah provinsi dilakukan oleh gubernur.

Trayek adalah teknik yang ditetapkan pemerintah untuk mengatur pola pelayanan angkutan umum. Penetapan jaringan trayek diatur dalam Keputusan Menteri Perhubungan Nomor KM 35 Tahun 2003 Pasal 3 bahwa penetapan jaringan trayek dilakukan berdasarkan jaringan transportasi jalan dengan mempertimbangkan:

1. Bangkitan dan tarikan perjalanan pada daerah asal dan tujuan.
2. Jenis pelayanan angkutan.

3. Hirarki kelas jalan yang sama dan atau lebih tinggi sesuai ketentuan kelas jalan yang berlaku.
4. Tipe terminal yang sesuai dengan jenis pelayanannya dan simpul transportasi lainnya yang meliputi bandar udara, pelabuhan dan stasiun kereta api.
5. Tingkat pelayanan jalan yang berupa perbandingan antara kapasitas jalan dan volume lalu lintas.

Kinerja-kinerja trayek angkutan umum yang diperhitungkan antara lain adalah (Gianopoulos, 1989):

1. Okupansi atau *load factor*.
2. Jam operasi.
3. Biaya operasi kendaraan.
4. *Headway*.
5. Frekuensi.
6. Daerah pelayanan.
7. Struktur rute.
8. *Route directness*.
9. Panjang rute.
10. Titik transfer.
11. *Overlapping*.
12. Rasio volume kapasitas.
13. Kecepatan.

2.5 QGIS

QGIS yang sebelumnya dikenal sebagai *Quantum GIS* adalah aplikasi SIG berbasis *desktop* yang bersifat *open source* dan dapat dijalankan di semua *platform* yang menyediakan tampilan, penyuntingan dan analisis data. Dibuat menggunakan bahasa pemrograman C++, Python dan Qt. QGIS pertama kali dirilis pada Juli 2002 oleh *QGIS Development Team* dan Gary Sherman sebagai pencetus aplikasi ini dengan menggunakan lisensi GNU GPL.

Fungsi QGIS mirip dengan aplikasi SIG lainnya, yaitu memungkinkan pengguna untuk membuat peta dengan banyak lapisan (*Layers*) menggunakan berbagai proyeksi peta atau CRS. QGIS juga dapat digunakan untuk mengolah data bertipe *raster* dan *vektor* serta terintegrasi dengan *package* GIS lainnya seperti PostGIS, GRASS GIS dan MapServer. Di dalam QGIS terdapat *plugins* yang memperkaya kemampuan serta menambahkan fungsi-fungsi seperti *plugin* untuk membuat *geocode* menggunakan Google *Geocoding* API, melakukan *geoprocessing* menggunakan *fTools* yang mirip dengan alat-alat standar yang dapat ditemukan pada *software* ArcGIS dan antarmuka dengan basisdata PostgreSQL/PostGIS, SpatiaLite dan MySQL. *Plugin* ditulis dengan bahasa pemrograman Python atau C++, sehingga dapat dikembangkan oleh orang lain yang ingin menambahkan fungsi yang dibutuhkan kedalam QGIS.

2.6 Android Studio

Pada tahun 2013, Android Studio resmi dirilis walaupun masih dalam versi beta. *Software* ini akan menjadi *Integrated Development Environment* (IDE) resmi untuk pengembangan Android. Android Studio berbasis IntelliJ Jetbrain, IntelliJ merupakan IDE Java komersial juga memiliki versi *open source*, ini adalah versi yang akan berfungsi sebagai dasar Android Studio (Hagos, 2018).

Adapun fitur dari *software* Android Studio ini adalah mampu melakukan *coding* dan melakukan iterasi dengan cepat, konfigurasi pembangunan tanpa batas, *coding* dengan percaya diri, membuat aplikasi yang kaya dan terkoneksi serta menghilangkan tugas yang melelahkan (Satyaputra dkk, 2016).

2.7 Python

Python dikembangkan oleh Guido van Rossum pada awal tahun 1990 di *Stichting Mathematisch Centrum*. Python sendiri dapat dikatakan merupakan penerus dari bahasa ABC. Pada tahun 1995, sebagian besar pengembangan Python dilakukan di *Corporation for National Research Initiatives*. Pengembangan Python saat ini dinaungi oleh Python *Software Foundation*.

Python merupakan interpreter yang dapat diandalkan dalam penanganan hal-hal yang sangat rumit dan membutuhkan perhitungan dengan ketelitian tingkat

tinggi. Dalam dunia *free software*, jarang sekali terdapat implementasi suatu bahasa pemrograman yang datang satu paket dengan GUI *toolkit* sendiri termasuk dalam hal ini Python. Meskipun Python menggunakan pustaka Tk, namun bukan dikembangkan secara khusus untuk Python. Selain itu, dukungan berbagai GUI *toolkit* untuk Python dapat ditemukan dimana-mana seperti PyGTK dan PyQt. Contoh aplikasi GUI yang dibangun dari bahasa Python adalah *installer* RedHat Linux. Pustaka-pustaka lain yang tidak datang bersama suatu bahasa pemrograman umumnya diimplementasikan oleh pihak ketiga dan dapat digunakan secara bebas.

Python dapat dijalankan di hampir semua *platform* mulai dari GNU/Linux, Windows dan Mac OS. Python termasuk ke dalam *general purpose programming language* dimana hampir semua tugas pemrograman di lingkungan sistem (dalam Linux), jaringan, sampai pemrograman berbasis *web*. Python juga menyediakan *framework* untuk membuat aplikasi jaringan.

2.8 PyQGIS

PyQGIS merupakan singkatan dari Python dan QGIS. Adalah pengembangan pada aplikasi QGIS dengan membuat *code* Python sehingga dapat dijalankan *standalone* sesuai kebutuhan pengguna. Sistem QGIS sendiri ditulis dalam bahasa C++ dan memiliki kumpulan API sendiri yang juga ditulis dalam bahasa C++. Python API diimplementasikan sebagai pembungkus di sekitar C++ API ini (Westra, 2014).

Mulai dari rilis 0.9, QGIS memiliki dukungan *scripting* opsional menggunakan bahasa Python, kami telah memutuskan untuk Python karena ini adalah salah satu bahasa yang paling favorit untuk *scripting*, *binding* PyQGIS bergantung pada SIP dan PyQt4, alasannya untuk menggunakan SIP daripada SWIG yang lebih banyak digunakan adalah bahwa keseluruhan kode QGIS bergantung pada perpustakaan QT, Python *binding* untuk Qt (PyQt) dilakukan juga menggunakan SIP dan ini memungkinkan integrasi PyQGIS dengan Qt (Lawhead, 2015).

2.9 Qt Designer

Qt adalah *toolkit* yang dipakai untuk membangun aplikasi GUI. Qt merupakan dasar bagi desktop manager *K Desktop Environment* (KDE). (Berjalan sama baiknya di bawah Gnome, selama telah terinstal *library* Qt).

Qt memiliki dua macam distribusi, yaitu Qt *Commercial Edition* dan Qt *Free Edition*. Edisi komersial (*Professional Edition* dan *Enterprise Edition*) memiliki lisensi Qt *Commercial License Agreement*. Sedangkan edisi bebas memiliki lisensi di bawah *General Public License* (GPL). Qt *Free Edition* juga tersedia di bawah lisensi *Q Public License* (QPL).

Qt dirancang oleh Trolltech mulai tahun 1992 dan mulai dirilis pada tahun 1995. Bahasa pemrograman yang dipakai adalah C++ Trolltech berpusat di Oslo, Norwegia.

Qt bersifat *cross platform*. Artinya, program berbasis Qt yang kita buat dalam Linux, misalnya, dapat di-*compile* dan dijalankan juga dalam sistem operasi lainnya, seperti juga dalam berbagai varian Unix, Windows dan Mac. Qt *Designer* merupakan *User Interface Designer* (UID) yang hampir mirip dengan IDE. Disebut UID dan belum berupa IDE karena dalam Qt *Designer* kita hanya dapat merancang GUI dan tidak dapat melakukan *compiling*, *debugging*, ataupun menjalankan program (Suharto dkk, 2012).

2.10 Firebase

Firebase adalah *database NoSQL* yang menyimpan data sebagai dokumen *Javascript Object Notation* (JSON) sederhana (Waikar, 2015). Firebase adalah *realtime database* yang dikembangkan oleh Google. Fitur utama dari Firebase adalah mampu menyelaraskan data secara otomatis tanpa melakukan *update request* atau *refresh* aplikasi (Wiratno dkk, 2017). Skema penyelarasan data secara otomatis dapat dilihat pada Gambar 2.2.



Gambar 2.2 Skema Penyelarasan Data Menggunakan Firebase
(Sumber: cloud.google.com, 2017)

Firestore memiliki struktur *database* yang unik dari beberapa *database* yang lain. Setiap *database* Firestore disimpan sebagai objek JSON yang mirip pohon (Waikar, 2015). *Read* data dari *database* Firestore dilakukan secara *asynchronous* menggunakan *event listener*. Setelah menambahkan *event listener* tertentu ke objek *ref*, *event listener* akan memanggil data terkait (Cheng, 2017).

2.11 PostGIS/PostgreSQL

PostGIS adalah program perangkat lunak *open source* yang menambahkan dukungan untuk objek geografis ke *database relasional* PostgreSQL. PostGIS mengikuti fitur sederhana untuk spesifikasi *Structured Query Language* (SQL) dari *Open Geospatial Consortium* (OGS). Ada sembilan alasan yang mendukung penggunaan PostGIS menurut (Urbano dkk, 2014):

1. Kedua perangkat lunak tersebut *free* dan *open source*, jadi sumber keuangan apapun bisa digunakan untuk kostumisasi dan bukan lisensi perangkat lunak dan dapat digunakan oleh penelitian kelompok yang bekerja dengan dana terbatas.

2.12 Leaflet Javascript

Leaflet adalah perpustakaan *open source javascript* yang membantu pembuatan peta pada halaman *web* menjadi lebih mudah. Sebagai *software open source* berarti *script* yang terdapat pada Leaflet dapat dengan mudah untuk dilihat cara kerjanya, siapa pun dapat menggunakannya dan yang lebih penting siapa pun dapat berkontribusi pada proyek dengan perbaikan kinerja Leaflet (Tanjaya dkk, 2016). *Javascript file* yang dimuat bersamaan dengan halaman *web* dapat menyediakan akses ke berbagai fungsi yang memungkinkan untuk menyajikan peta. Ada dukungan untuk *browser* modern di *desktop* dan *platform mobile* sehingga *user* dapat menyebarkan peta di mana saja.

2. PostgreSQL adalah sistem *database* yang maju dan banyak digunakan dan menawarkan banyak fitur yang berguna untuk pengelolaan data gerakan hewan.
3. PostGIS saat ini adalah salah satu *database* spasial paling banyak, tapi bukan yang paling canggih.
4. Ekstensi yang tersedia dan perkembangannya oleh komunitas IT sangat cepat.
5. PostGIS mencakup dukungan untuk data *raster*, data spasial geografi jenis khusus, topologi dan jaringan dan memiliki perpustakaan fungsi spasial yang sangat besar.
6. Tersedia komunitas yang luas, aktif dan sangat kolaboratif untuk PostgreSQL dan PostGIS.
7. Tersedia dokumentasi yang sangat bagus untuk PostgreSQL dan PostGIS.
8. PostgreSQL dan PostGIS menerapkan standar secara luas, yang membuat mereka sangat *interoperable* dengan seperangkat alat lain untuk pengolahan data, analisis, visualisasi dan diseminasi.
9. Tersedia untuk semua sistem operasi dan arsitektur CPU yang paling umum, terutama x86 dan x86_64.

Leaflet dirancang dengan sederhana, kinerja dan kegunaan dalam pikiran. Yang bekerja secara efisien di semua *platform desktop* dan *mobile device*, dapat dikembangkan dengan banyak *plugin*, mudah untuk digunakan dan dokumentasi API yang baik dan sederhana.

Selain itu Leaflet menyediakan fungsionalitas untuk menambahkan *marker*, *popup*, garis *overlay* dan bentuk, menggunakan beberapa lapisan, *zoom*, *pan*, tapi ini hanya fitur inti Leaflet. Salah satu kekuatan yang signifikan dari Leaflet adalah kemampuan untuk memperluas fungsionalitas dari *script* dengan plugin dari pihak ketiga. Pada saat pengkodean ada lebih dari 80 *plugin* terpisah yang memungkinkan fitur seperti *overlay heatmap*, *animating markers*, *loading csv files of data*, *drawing complex shapes*, *measuring distance*, *manipulating layers* dan *displaying coordinates*.

2.13 GeoJSON

GeoJSON adalah format standar terbuka yang dirancang untuk mewakili fitur geografis sederhana, bersama dengan atribut non spasialnya dan format ini berdasarkan pada format JSON (Butler dkk, 2016). Fitur-fitur tersebut meliputi *points* (alamat dan lokasi), *line strings* (jalan dan batas), *polygons* (negara dan provinsi) dan fitur *multipart* dari jenis-jenis ini.

2.14 Bootstrap

Bootstrap adalah sebuah alat bantu untuk membuat sebuah tampilan halaman *website* yang dapat mempercepat pekerjaan seorang pengembang *website* ataupun desainer halaman *website*. Bootstrap merupakan modular dan terdiri dari serangkaian *Less stylesheet* yang mengimplementasikan berbagai komponen *toolkit*. *Stylesheet* ini umumnya dikompilasi menjadi *bundle* dan dimasukkan ke dalam halaman *website*, tetapi masing-masing komponen dapat dimasukkan atau dihapus. Bootstrap menyediakan sejumlah variabel konfigurasi yang mengontrol hal-hal seperti warna dan *padding* berbagai komponen.

2.15 Google Maps *Application Programming Interface* (API)

API merupakan suatu dokumentasi yang terdiri *interface*, fungsi, kelas, struktur dan sebagainya untuk membangun sebuah perangkat lunak. Bahasa pemrograman yang digunakan oleh Google Maps yang terdiri dari HTML, *Javascript* dan AJAX serta XML memungkinkan untuk menampilkan peta Google Maps di *website* lain. Google menyediakan layanan Google Maps API yang memungkinkan para pengembang untuk mengintegrasikan Google Maps ke dalam *website* masing-masing dengan menambahkan data *point* sendiri (Amri, 2011). Google Maps API memberi pengembang banyak kelas fondasi untuk membangun solusi yang kompleks kasus yang berbeda, terutama untuk SIG (Davis, 2006).

2.16 *Web Map Tile Service* (WMTS)

Pemetaan *web* telah menjadi cara yang paling banyak digunakan untuk mendistribusikan pemetaan *online* melalui internet (Garcia, 2012). Beberapa layanan, seperti Google Maps atau Microsoft Bing Maps yang populer, memungkinkan pengguna untuk menampilkan kartografi dengan menggunakan *browser web* sederhana dan koneksi internet. Namun, informasi geografis adalah sumber daya yang mahal dan untuk alasan inilah standarisasi diperlukan untuk mempromosikan ketersediaan dan penggunaan kembali. Untuk membuat standar layanan peta semacam ini, *Open Geospatial Consortium* (OGC) mengembangkan *Web Maps Service* (WMS) rekomendasi (Beaujardiere, 2006). Standar ini menyediakan antarmuka HTTP sederhana untuk meminta gambar peta georeferensi dari satu atau lebih *database* geospasial terdistribusi. Dulu dirancang untuk render peta khusus, memungkinkan klien untuk meminta persis peta yang diinginkan gambar. Dengan cara ini, klien dapat meminta gambar peta berukuran sembarangan ke *server*, menempatkan diatas beberapa lapisan, yang meliputi kotak pembatas geografis, di semua CRS atau bahkan menerapkan gaya dan warna latar belakang tertentu.

Namun, fleksibilitas ini mengurangi potensi untuk meng-*cache* gambar peta, karena probabilitasnya menerima dua permintaan peta yang persis sangat rendah.

Oleh karena itu, ia memaksa gambar menjadi dinamis dihasilkan dengan cepat setiap kali permintaan diterima. Ini membutuhkan waktu yang sangat lama dan proses komputasi mahal yang berdampak negatif terhadap skalabilitas layanan dan pengguna *Quality of Service* (QoS). Pendekatan umum untuk meningkatkan kemampuan permintaan adalah membagi peta menjadi diskrit mengatur gambar, yang disebut ubin dan membatasi permintaan pengguna ke set itu (John, 2010). Ada beberapa spesifikasi telah dikembangkan untuk mengatasi bagaimana ubin gambar yang dapat di-cache ditampilkan dari sisi-server dan bagaimana permintaan klien ubin gambar yang di-cache. *Open Source Geospatial Foundation* (OSGeo) mengembangkan *WMS Tile Caching* (biasanya dikenal sebagai WMS-C) proposal (OGF, 2008). Nanti, OGC merilis *Web Map Tile Service Standard* (WMTS) (Maso, 2010) yang terinspirasi oleh yang pertama dan inisiatif serupa lainnya.

2.17 Apache

Apache *Web Server* awalnya dibuat pada tahun 1995 yang didasarkan dari server NCSA sebelumnya. Server NCSA dibuat oleh *National Center for Supercomputing Application* (yang juga mengembangkan browser Mosaic, pendahulunya hingga sebagian besar browser hari ini dan berhubungan dengan Netscape dan Mozilla serta pengaruh yang cukup besar termasuk MSIE).

Sebagai *web server*, Apache tergolong sukses terbukti pada bulan April 1996 telah menyusul NCSA sebagai *web server* yang paling banyak digunakan di internet (Kew, 2007). Tugas utamanya adalah melakukan *parse file* apa saja yang diminta oleh browser dan menampilkan hasil yang benar sesuai *code* di dalam *file* tersebut. Apache cukup kuat dan bisa melakukan tugas apa pun sebagai *web master*.

2.18 Unified Modeling Language (UML)

Menurut Sholiq (2006), notasi UML dibuat sebagai kolaborasi dari Grady Booch, DR. James Rumbough, Ivar Jacobson, Rebecca Wirfs-Brock, Peter Yourdon dan lainnya. UML menyediakan beberapa diagram yang menunjukkan

berbagai aspek dalam sistem. Ada delapan diagram yang disediakan dalam UML antara lain:

1. Diagram *use case* (*use case diagram*).
2. Diagram aktivitas (*activity diagram*).
3. Diagram sekuensial (*sequence diagram*).
4. Diagram kolaborasi (*collaboration diagram*).
5. Diagram kelas (*class diagram*).
6. Diagram *statechart* (*statechart diagram*).
7. Diagram komponen (*component diagram*).
8. Diagram *deployment* (*deployment diagram*).

Dalam penulisan Tugas Akhir ini, penulis menggunakan *use case diagram* dan *class diagram*. Diagram *use case* adalah model persyaratan sistem pada tingkat tinggi. Diagram *use case* digunakan untuk menunjukkan interaksi antara *use case* dan *actor*. *Use case* mewakili sistem fungsionalitas, persyaratan sistem dari perspektif pengguna sedangkan *actor* mewakili orang atau sistem yang menyediakan atau menerima informasi dari sistem (Boggs dkk, 2007).

2.19 Metode Pengembangan Sistem V Model

Metode pengembangan sistem V Model merupakan perluasan dari metode *waterfall*. Disebut sebagai perluasan karena tahapan-tahapannya mirip dengan yang terdapat dalam metode *waterfall*. Jika dalam metode *waterfall* proses dijalankan secara linear maka dalam V Model proses dilakukan bercabang. Dalam V Model ini menggambarkan hubungan antara tahap pengembangan *software* dengan tahap pengujiannya.

Proses Pengembangan pada V Model seimbang dan bergantung pada verifikasi dari langkah sebelumnya sebelum dilanjutkan. Hasil dari setiap tahap perlu diperiksa dan disetujui sebelum dilanjutkan (Balaji, 2012).

Berikut penjelasan masing-masing tahap beserta tahap pengujiannya.

1. *User Requirement & Acceptance Testing*

Tahap *User Requirement* sama seperti yang terdapat dalam metode *waterfall*. Hasil dari tahap ini adalah dokumentasi kebutuhan pengguna.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

- a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
- b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Acceptance Testing merupakan tahap yang akan mengkaji apakah dokumentasi yang dihasilkan tersebut dapat diterima oleh para pengguna atau tidak.

2. *System Requirements & System Testing*

Dalam tahap ini analisis sistem mulai merancang sistem dengan mengacu pada dokumentasi kebutuhan pengguna yang sudah dibuat pada tahap sebelumnya. Keluaran dari tahap ini adalah spesifikasi *software* yang meliputi organisasi sistem secara umum, struktur data dan yang lain.

3. *Global Design & Integration Testing*

Sering juga disebut *Hight Level Design*. Dasar dari pemilihan arsitektur yang akan digunakan berdasarkan kepada beberapa hal seperti pemakaian kembali tiap modul, ketergantungan tabel dalam basis data, hubungan antar interface, detail teknologi yang dipakai. Penyajian pada tahap desainpun berbagai macam, seperti menggunakan UML dan diagram.

4. *Detail Design & Unit Testing*

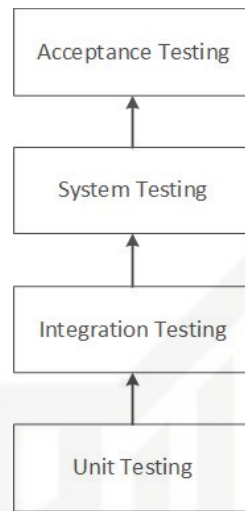
Perancangan dipecah menjadi modul-modul yang lebih kecil. Setiap modul tersebut diberi penjelasan yang cukup untuk memudahkan *programmer* melakukan *coding*. Tahap ini menghasilkan spesifikasi program seperti fungsi dan logika tiap modul, pesan kesalahan, proses input dan output untuk tiap modul dan lain-lain.

5. *Implementation*

Design yang telah dirancang kemudian ditranslasikan kedalam kode melalui *event-event* untuk mengimplementasikan logika program. Proses implementasi ini dilakukan pada perangkat lunak pengembangan.

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:
 - a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
 - b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.
2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.

Menurut Rifai (2015), terdapat empat fase pengujian yang dapat dilihat pada Gambar 2.3.



Gambar 2.3 Tahapan Fase Pengujian
(Sumber: Rifai dkk, 2015)

1. *Unit Testing*

Unit Testing adalah proses pengujian perangkat lunak dimana masing-masing unit atau komponen diuji. Tujuannya adalah untuk memvalidasi bahwa setiap unit perangkat lunak sudah melakukan seperti apa yang telah dirancang.

Menurut Pressman (2010), *unit testing* berfokus pada upaya verifikasi terhadap unit terkecil dari perancangan perangkat lunak. Pengujian unit berfokus pada logika pemrosesan internal dan struktur data didalam komponen.

Unit testing merupakan proses dimana pengujian dilakukan pada bagian *basic* dari kode program. Contohnya adalah memeriksa kode program pada *event*, *procedure* dan *function*. *Unit testing* meyakinkan bahwa masing-masing unit tersebut berjalan sebagaimana mestinya.

2. *Integration Testing*

Integration Testing adalah proses pengujian perangkat lunak dimana unit individu digabungkan dan diuji sebagai sebuah kelompok. Sehingga pengujian ini mampu menampilkan kesalahan dalam interaksi antar unit. Menurut Pressman (2010), pengujian integrasi adalah teknik untuk membangun arsitektur perangkat lunak, sementara pada saat yang sama melakukan pengujian untuk menemukan kesalahan terkait antarmuka. Tujuannya adalah untuk mengambil

komponen yang diuji dan membangun struktur program yang telah ditentukan oleh perancangan.

Setelah melakukan *component testing*, langkah berikutnya adalah memeriksa bagaimana unit-unit tersebut bekerja sebagai suatu kombinasi, bukan lagi sebagai suatu unit yang individual. Sebagai contoh, kita memiliki sebuah proses yang dikerjakan oleh dua *function*, di mana satu *function* menggunakan hasil *output* dari *function* yang lainnya.

Pada tahap *integration testing*, kita memeriksa hasil dari interaksi kedua *function* tersebut, apakah bekerja sesuai dengan hasil yang diharapkan. Kita juga harus memastikan bahwa seluruh kondisi yang mungkin terjadi dari hasil interaksi antar unit tersebut menghasilkan *output* yang diharapkan.

3. *System Testing*

System Testing adalah proses pengujian dimana perangkat lunak yang diuji sudah lengkap dan terintegrasi. Tujuan dari pengujian ini adalah untuk mengevaluasi kesesuaian sistem dengan persyaratan yang telah ditentukan.

4. *Acceptance Testing*

Acceptance Testing atau uji penerimaan adalah pengujian formal dilakukan untuk menentukan apakah sistem menerima kriteria penerimaan dan memastikan jika pengguna dapat menerima sistem. Tujuan dari pengujian ini adalah untuk mengetahui tingkat kelayakan dari perangkat lunak.

Seperti *integration testing*, *acceptance testing* juga meliputi pengujian keseluruhan aplikasi. Perbedaannya terletak pada siapa yang melakukan testing. Pada tahap ini, *end user* yang terpilih melakukan testing terhadap fungsi-fungsi aplikasi dan melaporkan permasalahan yang ditemukan. Proses ini merupakan salah satu tahap final sebelum pengguna menyetujui dan menerima penerapan sistem aplikasi yang baru.

2.20 Metode Pengujian *Blackbox Testing*

Teknik pengujian *blackbox* berfokus pada domain informasi dari perangkat lunak, dengan melakukan *test case* dengan mempartisi domain *input* dari suatu program dengan cara yang memberikan cakupan pengujian yang mendalam.

Blackbox merupakan metode pengujian perangkat lunak yang dilakukan hanya mengamati hasil eksekusi data uji dan memeriksa fungsional dari perangkat lunak. Cakupan pengujian yang dilakukan pada *blackbox testing* adalah perihail pengujian *interface* dan *form validation*. Pengujian *interface* adalah pengujian yang dilakukan secara langsung terhadap desain *interface* yang dibuat pada sistem. Tujuan yang diharapkan dalam melakukan *blackbox testing* dapat membuat desain yang sesuai dengan kebutuhan perusahaan. Pengujian *blackbox testing*, dilakukan untuk menemukan hal-hal sebagai berikut:

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*Interface errors*).
3. Kesalahan pada struktur data dan akses *database*.
4. Kesalahan performansi (*Performance errors*),
5. Kesalahan inisialisasi dan terminasi.

2.21 Penelitian yang Terkait

Penelitian terkait ini diambil dari jurnal-jurnal yang memiliki kesamaan topik dengan yang diteliti oleh peneliti. Berikut ini penelitian yang terkait dengan Tugas Akhir ini dapat dilihat pada Tabel 2.1.

Tabel 2.1 Penelitian yang Terkait dan Kontribusi Peneliti

No	Penelitian	Hasil	Kontribusi
1	<i>Implementation of firebase realtime database to track BRT</i> Trans Semarang (2017)	Penelitian ini menghasilkan aplikasi yang dapat menunjukkan lokasi bus trans semarang dengan menerapkan teknologi GPS <i>tracking</i> pada <i>smartphone</i> .	Kontribusi peneliti adalah mengintegrasikan aplikasi GPS <i>Tracking</i> dengan <i>web</i> sehingga dapat dilihat juga melalui <i>web</i> .
2	<i>Aplikasi Monitoring Armada Bus menggunakan GPS Tracking</i> pada <i>Smartphone</i>	Pada penelitian tersebut, menghasilkan aplikasi yang mampu menunjukkan posisi armada bus antar provinsi dengan menerapkan GPS <i>Tracking</i> pada <i>smartphone</i> .	Kontribusi peneliti adalah mengintegrasikan aplikasi GPS <i>Tracking</i> dengan <i>web</i> dan juga sistem PyQGIS.
3	Sistem Informasi Jalur dan Trayek Angkutan Umum di Kota Medan	Penelitian ini menghasilkan sistem informasi yang dapat menunjukkan jalur trayek angkutan umum berdasarkan nama trayek beserta informasi detail trayek.	Kontribusi peneliti adalah menghasilkan sistem informasi transportasi yang terintegrasi meliputi PyQGIS, WebGIS dan <i>mobile application Tracking</i> .

Tabel 2.1 Penelitian yang Terkait dan Kontribusi Peneliti (Lanjutan)

No	Penelitian	Hasil	Kontribusi
4	Analisis Kinerja dan Pemetaan Rute Bus DAMRI Bandara dengan GIS <i>Open Source</i> Kota Makassar (2012)	Dari penelitian tersebut mengidentifikasi karakteristik <i>demand</i> bus DAMRI bandara, tingkat pelayanan bus DAMRI dan mampu memperkirakan biaya operasional rute bus DAMRI bandara.	Kontribusi peneliti pada penelitian tersebut adalah mengimplementasikan hasil analisis karakteristik <i>demand</i> bus DAMRI bandara menjadi sebuah sistem informasi transportasi.
5	Penyusunan Sistem Informasi Geografis Infrastruktur Transportasi Kabupaten Kapuas Hulu Berbasis <i>Web</i> (2014)	Pada penelitian ini menghasilkan SIG yang dapat menangani data transportasi, serta menunjukkan penyebaran transportasi yang digambarkan dalam bentuk peta.	Kontribusi peneliti pada penelitian tersebut adalah menghasilkan SIG serupa namun terintegrasi dengan PyQGIS dan Android.
6	Sistem Informasi Transportasi dan Jalur Angkutan Kota untuk Penataan Ruang Wilayah Kota Semarang Guna Membantu Pengambilan Keputusan (2009)	Penelitian ini menghasilkan SIG yang mempermudah dalam menentukan lokasi kemacetan, pengembangan dan perluasan jaringan jalan serta penentuan areal rute kendaraan.	Kontribusi peneliti pada penelitian tersebut adalah menghasilkan SIG yang dapat menentukan rute terpendek bus DAMRI.
7	Sistem Informasi Geografis Jalan di Daerah Istimewa Yogyakarta Berbasis <i>Web</i> (2007)	Penelitian ini menghasilkan sistem yang dapat menampilkan peta jalan dan informasi atribut jalan sehingga memudahkan dalam mengakses informasi.	Kontribusi peneliti pada penelitian tersebut adalah menggunakan model informasi jalan pada sistem transportasi bus DAMRI.
8	Perancangan Sistem Informasi Trayek Angkutan Umum Kota Bogor Berbasis <i>Web</i> (2008)	Dari penelitian tersebut menghasilkan sistem yang dapat memberi kemudahan pengguna dalam merencanakan perjalanannya sesuai dengan trayek angkutan umum.	Kontribusi peneliti adalah memanfaatkan kemampuan pada penelitian tersebut untuk ditambahkan ke sistem transportasi bus DAMRI.
9	Pembuatan <i>plugin</i> analisis <i>least-cost path</i> pada QGIS untuk penentuan jalur trase jalan (2016)	Dari penelitian tersebut menghasilkan <i>plugin</i> yang mampu mencari jalur trase jalan dengan total akumulasi <i>cost</i> terendah antara dua lokasi dan mampu menghasilkan jalur dalam format <i>raster</i> dan <i>vektor</i> .	Kontribusi peneliti adalah membuat sistem transportasi yang memiliki kemampuan untuk menentukan jalan terdekat antara trayek dengan tujuan yang sama.

Tabel 2.1 Penelitian yang Terkait dan Kontribusi Peneliti (Lanjutan)

No	Penelitian	Hasil	Kontribusi
10	Pemetaan Rute dan <i>Demand</i> Angkutan Umum Kampus Universitas Hasanuddin Makassar berbasis Quantum GIS <i>Open Source</i> (2012)	Dalam penelitian yang dilakukan di Universitas Hasanuddin Makassar ini, menghasilkan pemetaan menggunakan metode <i>buffer</i> yang menunjukkan jangkauan pelayanan angkutan umum kampus dan hasil analisa aksesibilitas.	Kontribusi peneliti pada penelitian ini adalah menghasilkan SIG yang dapat memetakan jangkauan pelayanan bus DAMRI.

Hak Cipta Dilindungi Undang-Undang

1. Dilarang mengutip sebagian atau seluruh karya tulis ini tanpa mencantumkan dan menyebutkan sumber:

a. Pengutipan hanya untuk kepentingan pendidikan, penelitian, penulisan karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah.
b. Pengutipan tidak merugikan kepentingan yang wajar UIN Suska Riau.

2. Dilarang mengumumkan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin UIN Suska Riau.