BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Pada penelitian terkait ini, mengkaji penelitian mengenai sistem aplikasi yang dilakukan oleh penelitian sebelumnya yang dapat menjadi dasar dari penelitian diantaranya sebagai berikut:

Rahmi Elviana (2015) melakukan penelitian tentang "Perancangan Sistem Informasi Pemesanan Taksi Berbasis Mobile (Studi Kasus Taksi Kosti Padang)". Aplikasi ini dibuat untuk merancang model sistem informasi pemesanan tiket taksi *online* kemudian mengintegrasikan dengan sistem informasi geografis. Perancangan aplikasi dilakukan untuk menghubungkan basis data dengan perancangan pemesanan tiket online. Penelitian ini akan menggunakan bahasa pemrograman PHP, kemudian pembuatan basis data menggunakan *MySQL* dan menggunakan aplikasi *Basic4Android* untuk pembuatan aplikasi pada *smartphone*. Kemudian melakukan survei untuk menentukan aspek *usability* aplikasi yang dirancang. Berdasarkan dari perancangan sistem informasi pemesanan taksi berbasis *mobile* pada penelitian ini dapat diperoleh kesimpulan yaitu telah dihasilkan rancangan sistem informasi pemesanan taksi berbasis mobile pada Taksi Kosti Padang. User interface *android* digunakan untuk pelanggan dan supir, sedangkan aplikasi web untuk *operator controller*. Tampilan login dapat dilihat oleh pelanggan dan supir. Kemudian pelanggan dapat melakukan update data dan melihat data posisi, estimasi jarak, waktu, biaya dan rute. Supir dapat melihat tampilan untuk melihat data pelanggan serta data penjemputan. Sedangkan aplikasi *web* diperuntukkan untuk operator, di mana operator dapat memasukkan data pengendara, pelanggan, dan memeriksa data order. Aspek *usability* yang didapatkan dapat disimpulkan bahwa perangkat lunak aplikasi *android* dan *web* yang telah dibuat telah memiliki nilai *usability* yang baik

Tri Siddik Muhammad (2019) melakukan penelitian dengan judul "Aplikasi *Mobile* "Ke-Rent" (*Marketplace* Rental Kendaraan) Menggunakan *Location Based Service* (LBS)". Aplikasi yang dibuat untuk layanan proses penyewaan yang masih kurang efektif dan efisien. Pelanggan masih sulit dalam mencari informasi jasa

rental yang tersedia, harus mendatangi lokasi penyewaan secara langsung atau pemesanan via telpon, dan tidak ada tersedianya *marketplace* rental sebagai tempat proses pemesanan dan pemasaran rental kendaraan, di mana pelanggan tidak dapat memilih kendaraan yang sesuai dan mencari harga yang cocok. Sehingga dapat membantu dalam proses layanan rental kendaraan. Hasil dari penelitian ini berupa aplikasi "Ke-Rent" (rental kendaraan *online*) berbasis *Android*, para pemilik jasa rental dapat memasarkan kendaraannya dan masyarakat dengan mudah melakukan pencarian serta pemesanan rental kendaraan yang sesuai kebutuhan aplikasi yang dibangun (*marketplace* rental kendaraan) memiliki perbedaan dengan aplikasi sejenis yang sudah ada sebelumnya seperti aplikasi "Docar" dan "Nemob" yang terdapat pada *playstore Android*, aplikasi ini sudah cukup baik namun memiliki kekurangan fitur, seperti pemesanan lepas kunci, tidak adanya pilihan kategori kendaraan. Selain itu aplikasi lainnya yakni "Djaloer", pada aplikasi ini memiliki kelemahan terutama pada tampilan yang sangat sederhana dan kurang menarik, dan masih banyak fitur yang kurang sehingga akan membingungkan pengguna. Selain aplikasi yang disebutkan masih banyak aplikasi sejenis lainnya di mana perlunya pengembang. Beberapa perbedaan aplikasi (Ke-rent) yang akan dibangun adalah dengan pemanfaatan LBS pada fitur lepas kunci, di mana akan memberikan navigasi kepada pelanggan menuju lokasi jasa rental untuk memberikan syarat dan ketentuan pemesanan kendaraan lepas kunci. Selanjutnya proses pembayaran selain membayar dengan transfer tunai, pelanggan juga bisa menggunakan saldo pada dompet *virtual*. Pada aplikasi ini selain ditujukan kepada pemilik jasa rental pihak pribadi juga dapat mendaftarkan kendaraannya yang ingin direntalkan.

Gintoro, Iwan Wijaya Suharto, Febiyan Rachman, Daniel Halim (2010) melakukan penelitian dengan judul "Analisis Dan Perancangan Sistem Pencarian Taksi Terdekat Dengan Pelanggan Menggunakan Layanan Berbasis Lokasi". Penelitian ini membahas tentang sistem pemesanan taksi konvensional yang tidak bisa menemukan taksi-taksi terdekat dengan pemesan. Penelitian ini bertujuan untuk membangun sistem untuk membantu pencarian taksi-taksi terdekat dengan pemesan menggunakan layanan berbasis lokasi, yang diimplementasikan menggunakan GPS dan teknologi *BlackBerry Push*. Pada umumnya penyedia

layanan taksi masih menggunakan pemancar radio untuk menginformasikan adanya pemesan jasa taksi kepada para sopir taksi. Hal tersebut masih dilakukan secara manual oleh operator layanan taksi dan belum adanya konfirmasi taksi mana yang akan melayani pemesan taksi nantinya. Pencarian taksi terdekat untuk pemesan taksi juga tidak bisa dilakukan tanpa mengetahui lokasi pemesan taksi secara otomatis. Hampir semua tipe produk *BlackBerry* baru yang didistribusikan di Indonesia memiliki perangkat GPS (*Global Positioning System*) tertanam, yang memudahkan pengguna untuk mendapatkan lokasinya saat itu. Dengan memanfaatkan fasilitas GPS, maka dapat dikembangkan sistem untuk otomatisasi pencarian taksi terdekat untuk pemesanan taksi lewat *BlackBerry*. Pada penelitian ini, sistem yang dapat mengotomatisasi pencarian taksi dengan jarak terdekat terhadap posisi pemesan taksi serta mengotomatisasi penyebaran informasi ke taksi dan konfirmasi pemesanan layanan taksi ke pengguna layanan taksi akan dibuat.

Berikut adalah hal-hal yang membedakan setiap kajian terkait yang dijelaskan pada tabel 2.1 berikut ini:

Tabel 2. 1 Pebanding Kajian Terkait

No	Penulis	Judul	Keterangan
1.	Rahmi Elviana (2015) Universitas Andalas Padang Sumatra Barat	Perancangan Sistem Informasi Pemesanan Taksi Berbasis Mobile (Studi Kasus Taksi Kosti Padang)	1. Berbasis mobile. 2. Aplikasi ini dibuat untuk merancang model sistem informasi pemesanan tiket taksi *online* kemudian mengintegrasikan dengan sistem informasi geografis.
3.	Tri Siddik Muhammad (2019) UniversitasIslam Negeri Sultan Syarif	Aplikasi *Mobile* "Ke-Rent" (*Marketplace* Rental Kendaraan) Menggunakan	1. Berbasis mobile. 2. Aplikasi ini dirancang menggunakan metodologi RUP

	Kasim Riau Pekanbaru	*Location Based Service* (LBS)	(Rational Unified Process) 3. Mengelola informasi pemilik jasa rental untuk memasarkan kendaraannya dan masyarakat dengan mudah melakukan pencarian serta pemesanan rental kendaraan yang sesuai kebutuhan aplikasi yang dibangun (*marketplace* rental kendaraan)
4.	Gintoro, Iwan Wijaya Suharto, Febiyan Rachman, Daniel Halim (2010) Seminar Nasional Aplikasi Teknologi Informasi Yogyakarta	Analisis Dan Perancangan Sistem Pencarian Taksi Terdekat Dengan Pelanggan Menggunakan Layanan Berbasis Lokasi	1. Berbasis blackberry. 2. Menggunakan metode diagram usecase. 3. Penelitian ini membahas tentang sistem pemesanan taksi konvensional yang tidak bisa menemukan taksi-taksi terdekat dengan pemesan. 4. Membangun sistem untuk membantu pencarian taksi-taksi terdekat dengan pemesan menggunakan layanan berbasis lokasi, yang

			diimplementasikan menggunakan GPS dan teknologi *BlackBerry Push.*

Tabel 2. 2 Penelitian Yang Dilakukan.

No	Penulis	Judul	Keterangan
1.	Gregorius Eduward Dicky Candra Universitas Tanjungpura Pontianak	Aplikasi *Airport Taxi Sharing* Untuk Mengoptimalkan Calon Penumpang Ke Bandara Menggunakan Metode *Location Based Service* Berbasis *Progressive Web App*	1. Aplikasi Berbasis *Progressive web app.* 2. Menggunakan metode *location based service*. 3. Menggunakan GPS sebagai penentuan tiitk lokasi supir dan penumpang. 4. Aplikasi menampilkan titik-titik lokasi supir berdasarkan radius yang ditentukan. 5. Calon Penumpang dapat melihat status dan lokasi supir. 6. Supir Taksi dapat melihat lokasi penumpang untuk melakukan penjemputan.

2.2 Aplikasi

Aplikasi adalah suatu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti sistem perniagaan, game pelayanan masyarakat, periklanan, atau semua proses yang hampir dilakukan manusia Pramana (2012). Aplikasi merupakan program yang dikembangkan untuk memenuhi kebutuhan pengguna dalam menjalankan pekerjaan tertentu Yuhefizar (2012). Aplikasi adalah program yang memiliki aktivitas pemrosesan perintah yang diperlukan untuk melaksanakan permintaan pengguna dengan tujuan tertentu Supriyanto (2005),.

Berdasarkan dua pendapat ahli tersebut, maka dapat disimpulkan bahwa aplikasi merupakan sebuah program perangkat lunak yang dibuat dengan tujuan memenuhi kebutuhan dari aktivitas yang dilakukan oleh pengguna serta mempermudah pekerjaan dari pengguna tersebut. Pada penelitian ini Aplikasi akan menjadi alat bantu kepada *driver* taksi bandara yang dibangun dengan *Progressive Web App* (PWA*)*.

2.3 Transportasi

Transportasi adalah pemindahan manusia atau barang dari satu tempat ke tempat lainnya dalam waktu tertentu dengan menggunakan sebuah kendaraan yang digerakkan oleh manusia, hewan, maupun mesin, Christian (http://e-journal.uajy.ac.id, 2015).

Rustian Kamaludin (2003: 13) dalam Hamidah (2017), mengungkapkan bahwa transportasi berasal dari kata Latin, *transportare* di mana trans berarti seberang atau sebelah lain dan *portare* berarti mengangkut atau membawa. Jadi, transportasi berarti mengangkut atau membawa (sesuatu) ke sebelah lain atau dari suatu tempat ke tempat lainnya. Dengan demikian, transportasi adalah sebagai usaha dan kegiatan menyangkut atau membawa barang dan / atau penumpang dari suatu tempat ke tempat lainnya.

Menurut Christian (http://e-journal.uajy.ac.id, 2015), pengertian transportasi menurut beberapa ahli adalah sebagai berikut:

1. Menurut Morlok (1978), transportasi didefinisikan sebagai kegiatan memindahkan atau mengangkut sesuatu dari suatu tempat ke tempat lain.

2. Menurut Bowersox (1981), transportasi adalah perpindahan barang atau penumpang dari suatu tempat ke tempat lain, di mana produk dipindahkan ke tempat tujuan dibutuhkan. Secara umum transportasi adalah suatu kegiatan memindahkan sesuatu (barang dan/atau barang) dari suatu tempat ke tempat lain, baik dengan atau tanpa sarana.
3. Menurut Steenbrink (1974), transportasi adalah perpindahan orang atau barang dengan menggunakan alat atau kendaraan dari dan ke tempat-tempat yang terpisah secara geografis.
4. Menurut Papacostas (1987), transportasi didefinisikan sebagai suatu sistem yang terdiri dari fasilitas tertentu beserta arus dan sistem *control* yang memungkinkan orang atau barang dapat berpindah dari suatu tempat ke tempat lain secara efisien dalam setiap waktu untuk mendukung aktivitas manusia.

2.4 Angkutan Umum

Angkutan Umum dapat didefinisikan sebagai pemindahan manusia dan barang dari suatu tempat ke tempat lain dengan menggunakan kendaraan. Kendaraan umum adalah setiap kendaraan bermotor yang disediakan untuk umum dengan dipungut bayaran. Kendaraan umum dapat berupa mobil penumpang, bus kecil, bus sedang, dan bus besar.

Mobil penumpang yang digunakan untuk mengangkut penumpang umum disebut mobil penumpang umum (MPU). Bus kecil dicirikan dengan jumlah tempat duduk sekurang-kurangnya 9 (sembilan) sampai 19 (sembilan belas) tempat duduk, tidak termasuk tempat duduk pengemudi. Bus sedang adalah mobil bus yang dilengkapi sekurang-kurangnya 20 (dua puluh) sampai dengan 30 (tiga puluh) tempat duduk. Bus besar adalah bus yang dilengkapi sekurang-kurangnya 31 (tiga puluh satu) tempat duduk, tidak termasuk tempat duduk pengemudi. 25 Aset berupa kendaraan mobil bus/MPU yang dipertanggungjawabkan perusahaan, baik yang dalam keadaan siap guna maupun dalam konservasi disebut dengan armada. Konservasi adalah sejumlah bus/MPU yang merupakan sebagian dari armada, yang tidak lagi dioperasikan untuk pelayanan penumpang umum karena bus/MPU dalam keadaan rusak atau tidak layak jalan.

Pelayanan angkutan orang dengan kendaraan umum dapat diklasifikasikan berdasarkan wilayah pelayanan, operasi pelayanan, dan peranannya. Berdasarkan wilayah pelayanannya, angkutan penumpang umum terdiri atas angkutan pedesaan, angkutan perkotaan, angkutan antar kota, dan angkutan lintas batas negara. Berdasarkan operasi pelayanannya, angkutan penumpang umum dapat dilaksanakan dalam trayek tetap dan teratur serta tidak dalam trayek. Pembagian trayek tetap dan teratur adalah sebagai berikut ini:

1. Trayek Antar Kota Antar Provinsi (AKAP) dan lintas batas negara, trayek yang wilayah pelayanannya lebih dari satu provinsi.
2. Trayek Antar Kota Dalam Provinsi (AKDP), trayek yang wilayah pelayanannya melebihi satu wilayah kabupaten/kota namun masih dalam satu provinsi.
3. Trayek perkotaan dan pedesaan.
4. Pengangkutan dengan taksi.
5. Dengan cara sewa.
6. Pengangkutan pariwisata.

Angkutan dengan taksi dapat diklasifikasikan sesuai batasan wilayah pelayannya, seperti berikut ini:

a) Pelayanan taksi dengan wilayah operasinya hanya dalam wilayah administratif kota.
b) Pelayanan taksi dengan wilayah operasinya melampaui wilayah administratif kota/kabupaten dalam satu provinsi.
c) Pelayanan taksi dengan wilayah operasinya melampaui wilayah administratif kota/kabupaten dan melewati satu provinsi.

Sedangkan pengangkutan dengan cara sewa dan pariwisata tidak dibatasi wilayah pelayanannya.

Syarat dan Prosedur Angkutan Umum:

Jaringan trayek menurut pedoman teknis penyelenggaraan angkutan penumpang umum di wilayah perkotaan dalam trayek tetap dan teratur adalah kumpulan trayek yang menjadi satu kesatuan pelayanan angkutan orang. Faktor yang digunakan sebagai bahan pertimbangan dalam menetapkan jaringan trayek adalah sebagai berikut:

a. Pola tata guna tanah Pelayanan angkutan umum diusahakan mampu menyediakan aksesibilitas yang baik. Untuk memenuhi hal itu, lintasan trayek angkutan umum diusahakan melewati tata guna tanah dengan potensi permintaan yang tinggi. Demikian juga lokasi-lokasi yang potensial menjadi tujuan bepergian diusahakan menjadi prioritas perjalanan. Peraturan Menteri Perhubungan Nomor 26 Tahun 2017 tentang Penyelenggaraan Angkutan Orang Dengan Kendaraan Bermotor Umum Tidak Dalam Trayek 27.
b. Pola pergerakan penumpang angkutan umum Rute angkutan umum yang baik adalah rute yang mengikuti arah pola pergerakan penumpang angkutan sehingga tercipta pergerakan yang lebih efisien. Trayek angkutan umum harus dirancang sesuai dengan pola pergerakan penduduk yang terjadi, sehingga transfer moda yang terjadi pada saat penumpang mengadakan perjalanan dengan angkutan umum dapat diminimumkan.
c. Kepadatan penduduk Salah satu faktor yang menjadi prioritas angkutan umum adalah wilayah kepadatan penduduk yang tinggi, pada umumnya merupakan wilayah yang mempunyai potensi permintaan yang tinggi. Trayek angkutan umum yang ada diusahakan sedekat mungkin menjangkau wilayah tersebut.
d. Daerah pelayanan angkutan umum, selain memperhatikan wilayah potensial pelayanan, juga menjangkau semua wilayah perkotaan yang ada. Hal ini sesuai dengan konsep pemerataan pelayanan terhadap penyediaan fasilitas angkutan umum.
e. Karakteristik jalan Kondisi jaringan jalan akan menentukan pola pelayanan trayek angkutan umum. Karakteristik jaringan jalan meliputi konfigurasi, klasifikasi, fungsi, lebar jalan, dan tipe operasi jakur. Operasi angkutan umum sangat dipengaruhi oleh karakteristik jaringan jalan yang ada.

2.5 Taksi

Angkutan Taksi adalah angkutan dengan menggunakan mobil penumpang umum yang diberi tanda khusus, memenuhi syarat-syarat teknis, dilengkapi dengan argometer, untuk melayani angkutan dari pintu ke pintu (*door to door*) dalam wilayah operasi tertentu. Angkutan Taksi adalah jenis pelayanan angkutan orang dengan kendaraan bermotor umum tidak dalam trayek. Dalam arti tidak dalam

trayek yaitu angkutan yang dilayani dengan mobil penumpang umum dalam wilayah perkotaan dan atau kawasan tertentu atau dari suatu tempat ke tempat lain namun tidak mempunyai lintasan dan waktu tetap.

Pelayanan angkutan taksi diatur dalam Peraturan Menteri Perhubungan Republik Indonesia Nomor PM 26 tahun 2017 tentang Penyelenggaraan Angkutan Orang Dengan Kendaraan Bermotor Umum Tidak Dalam Trayek Pasal 5:

Pelayanan angkutan taksi wajib memenuhi pelayanan sebagai berikut:

a. Wilayah operasi pelayanan berada di dalam kawasan perkotaan.
b. Tidak berjadwal.
c. Pelayanan dari pintu ke pintu.
d. Tujuan perjalanan ditentukan oleh pengguna jasa.
e. Tarif angkutan berdasarkan argometer atau tertera pada aplikasi berbasis teknologi informasi.
f. Besaran tarif berdasarkan tarif batas atas dan batas bawah yang ditetapkan oleh pejabat yang berwenang sesuai ketentuan peraturan perundangundangan.
g. Wajib memenuhi Standar Pelayanan Minimal yang ditetapkan.
h. Pembayaran pada pelayanan angkutan taksi yang dilakukan berdasarkan argometer dilengkapi dengan alat bukti pembayaran yang tercetak.
i. Pemesanan dapat dilakukan melalui aplikasi berbasis teknologi informasi

Kendaraan yang dipergunakan untuk pelayanan Taksi juga wajib memenuhi persyaratan sebagai berikut:

a. Kendaraan yang dipergunakan meliputi:
    1. Mobil penumpang sedan yang memiliki 3 (tiga) ruang.
    2. Mobil penumpang bukan sedan yang memiliki 2 (dua) ruang.
b. Tulisan “TAKSI” yang ditempatkan di atas atap bagian luar kendaraan dan harus menyala dalam keadaan kosong dan padam apabila argometer dihidupkan.
c. Dilengkapi dengan tanda nomor kendaraan bermotor dengan warna dasar kuning tulisan hitam.
d. Argometer yang disegel oleh instansi yang berwenang dan dapat berfungsi dengan baik serta ditera ulang sesuai dengan ketentuan peraturan perundangundangan.

e. Nama perusahaan atau merek dagang, serta logo yang ditempatkan pada pintu depan bagian tengah, dengan susunan sebelah atas adalah logo perusahaan dan sebelah bawah adalah nama perusahaan/merek dagang.
f. Lampu bahaya berwarna kuning ditempatkan di samping kanan tanda taksi.
g. Identitas pengemudi ditempatkan pada kabin kendaraan, mudah terlihat jelas oleh penumpang, yang dikeluarkan oleh masing-masing perusahaan angkutan taksi.
h. Alat komunikasi sebagai penghubung antara pengemudi dengan pusat pengendali operasi dan atau sebaliknya, baik secara audio, visual, atau data.
i. Keterangan tentang biaya awal, kilometer, waktu, dan biaya tambahan yang ditempatkan pada sisi bagian dalam pintu belakang.
j. Dilengkapi dokumen perjalanan yang sah, berupa surat tanda nomor kendaraan atas nama badan hukum, kartu uji, dan kartu pengawasan.
k. Nomor urut kendaraan dari setiap perusahaan angkutan yang ditempatkan pada bagian depan, belakang, kanan atau kiri kendaraan, dan bagian dalam kendaraan.
l. Nomor pengaduan masyarakat yang dicantumkan di bagian dalam atau bagian luar kendaraan.

2.6 Optimalisasi

Optimalisasi adalah berasal dari kata dasar optimal yang berarti terbaik, tertinggi, paling menguntungkan, (Kamus Besar Bahasa Indonesia, 2011:345). Menjadikan paling baik, menjadikan paling tinggi, pengoptimalan proses, cara, perbuatan mengoptimalkan (menjadikan paling baik, paling tinggi, dan sebagainya), sehingga optimalisasi adalah suatu tindakan, proses, atau metodologi untuk membuat sesuatu (sebagai sebuah desain, system, atau keputusan) menjadi lebih/sepenuhnya sempurna, fungsional, atau lebih efektif.

Sedangkan dalam Kamus *Oxford* (2008:358) *"Optimization is the process of finding the best solution to some problem where "best" accords to prestated criteria".* Yang dimaksudkan adalah optimalisasi adalah sebuah proses, cara, dan perbuatan (aktivitas/kegiatan) untuk mencari solusi terbaik dalam beberapa masalah, dimana yang terbaik sesuai dengan kriteria tertentu. Menurut Machfud

Sidik, (2001:8) “Optimalisasi suatu tindakan/kegiatan untuk meningkatkan dan mengoptimalkan.”

Optimalisasi adalah upaya seseorang untuk meningkatkan suatu kegiatan atau pekerjaan agar dapat memperkecil kerugian atau memaksimalkan keuntungan agar tercapai tujuan sebaik-baiknya dalam batas-batas tertentu (Andri Rizki Pratama, 2013:6).

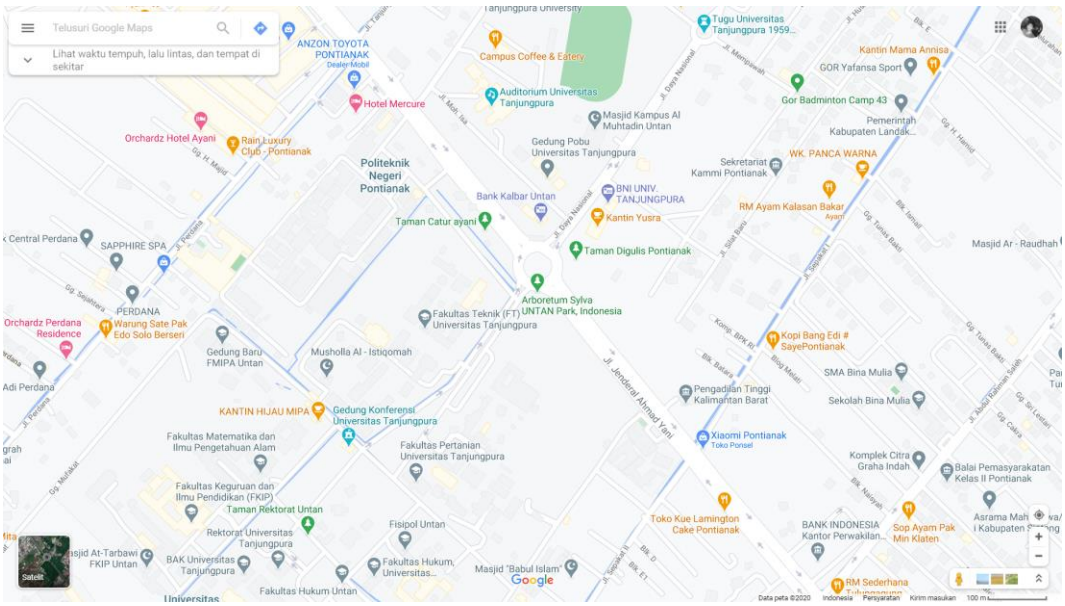
Dari pengertian tersebut dapat disimpulkan bahwa optimalisasi adalah suatu proses kegiatan untuk meningkatkan dan mengoptimalkan suatu pekerjaan menjadi lebih/sepenuhnya sempurna, fumgsional, atau lebih efektif serta mencari solusi terbaik dari beberapa masalah agar tercapai tujuan sebaik-baiknya sesuai dengan kriteria tertentu.

2.7 Sistem Informasi Geografis

Sistem Informasi Geografis (SIG) atau juga dikenal sebagai *Geographic Information System* (GIS) pertama pada tahun 1960 yang bertujuan untuk menyelesaikan permasalahan *geografis*. Empat puluh tahun kemudian GIS berkembang tidak hanya bertujuan untuk menyelesaikan permasalahan geografi 7 saja, tetapi sudah merambah ke berbagai bidang, seperti analisis penyakit epidemi (demam berdarah) dan analisis kejahatan (kerusuhan), termasuk analisis kepariwisataan. Kemampuan dasar dari SIG adalah mengintegrasikan berbagai operasi basis data seperti query, menganalisisnya serta menampilkannya dalam bentuk pemetaan berdasarkan letak geografisnya. Inilah yang membedakan SIG dengan sistem informasi lain (Prahasta, 2014).

2.8 Google Map API

Google Maps adalah sebuah jasa peta global virtual gratis dan online yang disediakan oleh perusahaan *Google. Google Maps* yang dapat ditemukan di alamat http://maps.google.com. *Google Maps* menawarkan peta yang dapat diseret dan gambar satelit untuk seluruh dunia.*Google Maps* juga menawarkan pencarian suatu tempat dan rute perjalanan. Dapat dilihat pada Gambar 2.1.



Gambar 2. 1 Tampilan Google Maps

Google Maps API adalah sebuah layanan *(service)* yang diberikan oleh *Google* kepada para pengguna untuk memanfaatkan *Google Map* dalam mengembangkan aplikasi. *Google Maps API* menyediakan beberapa fitur untuk memanipulasi peta, dan menambah konten melalui berbagai jenis *services* yang dimiliki, serta mengizinkan kepada pengguna untuk membangun aplikasi enterprise di dalam websitenya, *Google Maps API* adalah suatu *library* yang berbentuk *Javascript*. (Kindarto, 2008). Pada penelitian ini *Google Map API* akan menjadi *service* seperti *interface* bagi pengguna berupa *Map* yang akan dipakai.

2.9 Haversine Formula

Rumus Haversine adalah persamaan penting dalam navigasi, memberikan jarak lingkaran besar antara dua titik pada bola dari garis bujur dan garis lintangnya (Farid & Yunus, 2017).

Penggunaan rumus ini mengasumsikan pengabaian efek ellipsoidal, cukup akurat untuk sebagian besar perhitungan, juga pengabaian ketinggian bukit dan kedalaman lembah di permukaan bumi. Berikut adalah rumus haversine.

$x = (long2 - long1) * cos((lat2 + lat1) \div 2)$

$y = (lat2 - lat1)$

$d = \sqrt{x * x + y * y} * R$

Keterangan Rumus

x = Longitude / Lintang

y = Latitude / Bujur

d = Jarak

R = Radius Bumi = 6371 km

1 $derajat$ = 0,0174532925 radian

2.9.1 Penggunaan Formula Haversine

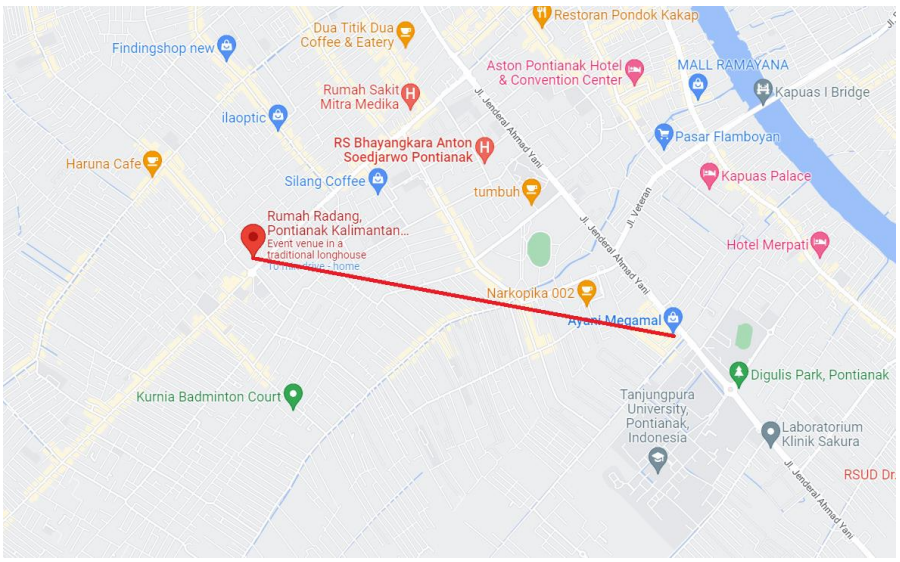
Formula Haversine dapat digunakan untuk mencari perhitungan jarak antar dua titik, umumnya pengukuran jarak akan dilakukan dengan menarik garis lurus pada peta untuk diukur jarak antar keduanya.

Tabel 2. 3 Contoh Kasus

Titik A (Rumah Radakng)		Titik B (Ayani *Megamall*)	
Long: 109.3171088	Lat: -0.0469081	Long: 109.3437163	Lat: -0.051849

Berdasarkan data dari Tabel 2.3, terdapat koordinat awal dan koordinat tujuan, yang akan dijadikan bahan ukur untuk pencarian jarak haversine.

Pada Gambar 2.2 merupakan ilustrasi pencarian jarak menggunakan metode haversine. terlihat pada Google Maps jarak yang didapat jika menarik garis lurus dari Titik A Rumah Radakng menuju Titik B Ayani *Megamall* adalah 3 Km.



Gambar 2. 2 Ilustrasi Pengukuran

Setelah didapat hasil pengukuran google maps pada Gambar 2.2, berikut adalah penjelasan perhitungan jarak menggunakan formula haversine yang akan dikerjakan di Microsoft Windows Excel, hasilnya sebagai berikut:

Perhitungan Jarak Algoritma Haversine			
Lokasi	**Latitude**	**Longitude**	
Rumah Radakng	-0,046908	109,3171088	
Ayani Megamall	-0,051849	109,3437163	
Haversine =	3,0091965	Km	=(6371*((2*ASIN(SQRT((SIN((RADIANS(C5)-RADIANS(C4))/2)^2)+COS(RADIANS(C5))*COS(RADIANS(C4))*(SIN((RADIANS(D5)-RADIANS(D4))/2)^2))))))

Gambar 2. 3 Contoh PerhitunganMetode Haversine

Perhitungan jarak menggunakan metode haversine pada Gambar 2.2, didapati bahwa jarak antara 2 titik adalah 3 Km.

2.10 Global Positioning System (GPS)

Global Positioning System merupakan sistem navigasi menggunakan teknologi satelit yang dapat menerima sinyal dari satelit. Sistem ini menggunakan satelit yang mengirimkan sinyal gelombang mikro ke bumi. Sinyal ini diterima oleh alat penerima (*receiver*) di permukaan, di mana GPS *receiver* ini akan mengumpulkan informasi dari satelit GPS, seperti:

a. Waktu. GPS *receiver* menerima informasi waktu dari jam atom yang mempunyai keakurasian sangat tinggi.
b. Lokasi. GPS memberikan informasi lokasi dalam tiga dimensi yakni *Latitude*, *Longitude*, Elevasi
c. Kecepatan. Ketika berpindah tempat, GPS dapat menunjukkan informasi kecepatan berpindah tersebut.
d. Arah perjalanan. GPS dapat menunjukkan arah tujuan.
e. Simpan lokasi. Tempat-tempat yang sudah pernah atau ingin dikunjungi bisa disimpan oleh GPS *receiver*.
f. Kumulasi data. GPS *receiver* dapat menyimpan informasi *track*, seperti total perjalanan yang sudah pernah dilakukan, kecepatan rata-rata, kecepatan paling tinggi, kecepatan paling rendah, waktu/jam sampai tujuan, dan sebagainya. (Wishnu, 2012).

2.10.1 Akurasi GPS

Posisi yang ditunjukkan oleh suatu GPS mempunyai faktor kesalahan atau juga disebut tingkat akurasi. Sebagai contoh suatu alat GPS menunjukkan titik koordinat dengan tingkat akurasi 5 meter, itu berarti posisi pengguna bisa berada dalam range radius 5 meter dari titik yang ditunjukkan tersebut. Mengapa tingkat akurasi yang terlihat bisa berubah-ubah? Kadang terlihat 10 meter, 15 meter, atau 5 meter. Ada beberapa hal yang mempengaruhi tingkat akurasi tersebut, antara lain:

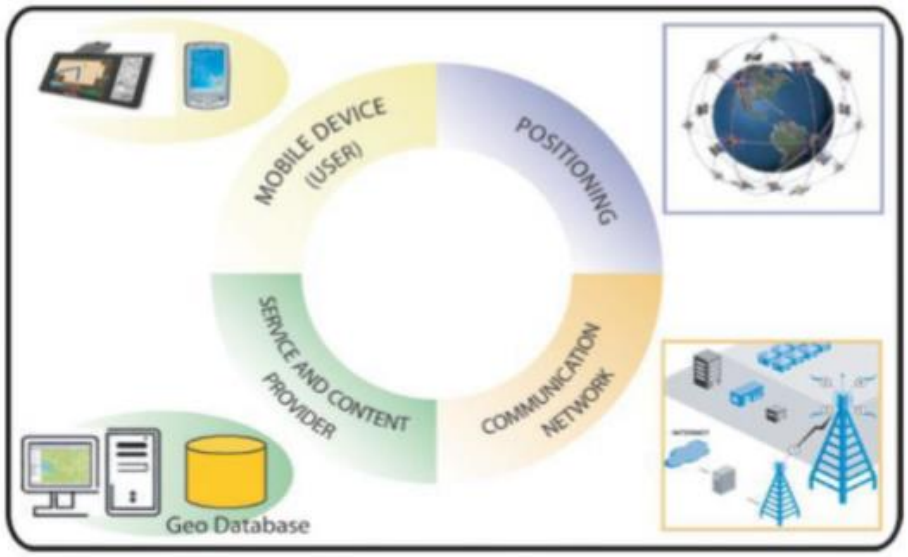
a. Kesalahan *Ephemeris*. Terjadi jika satelit tidak dapat mentransmisikan posisinya di orbit dengan tepat.
b. Keadaan *Ionosphere*. *Ionosphere* berada pada jarak sekitar 43-50 mil di atas permukaan bumi. Satelit yang melewati *ionosphere* akan menjadi lambat dikarenakan adanya plasma (gas dengan tingkat kepadatan rendah). Walaupun GPS *receiver* berusaha untuk mengkoreksi/memperbaiki faktor keterlambatan yang terjadi tetap saja aktivitas tertentu dari plasma bisa menyebabkan kesalahan perhitungan.
c. Keadaan *Troposphere*. *Troposphere* adalah bagian terendah dari *atmosfer* sampai dengan ketinggian sekitar 11 mil dari permukaan tanah. Variasi pada

temperatur, tekanan, dan kelembaban bisa menyebabkan perbedaan kecepatan penerimaan gelombang radio.

d. Kesalahan Waktu. Karena penempatan jam atom pada setiap GPS *receiver* tidak berjalan sebagaimana mestinya. Kesalahan waktu dari GPS *receiver* yang tidak presisi dapat menimbulkan ketidakakurasian.
e. Kesalahan *Multipath*. Terjadi karena sinyal satelit membentur permukaan keras (seperti bangunan atau tebing) sebelum mencapai GPS *receiver*. Hal tersebut bisa menyebabkan terjadinya delay sehingga perhitungan jarak menjadi tidak akurat.
f. Buruknya Sinyal Satelit. Keadaan langit yang terhalang akan menyebabkan GPS sulit menerima data satelit. Sebuah sinyal satelit yang pada hari tertentu diterima dengan sangat bagus belum tentu pada hari lain bisa diterima dengan kualitas yang sama walaupun user berdiri pada tempat yang sama. Hal tersebut dikarenakan posisi dari satelit yang terus bergerak atau bisa juga disebabkan faktor penghalang lain seperti pohon, gedung bertingkat, dan sebagainya. (Wishnu, 2012).

2.11 Location Based Service

Location Based Service (LBS) atau Layanan Berbasis Lokasi merupakan layanan informasi yang memanfaatkan kemampuan untuk menggunakan informasi lokasi dari perangkat bergerak dan dapat diakses dengan perangkat bergerak melalui jaringan telekomunikasi bergerak (Steiniger, 2006). Dalam layanan Berbasis Lokasi terdapat lima komponen penting seperti terlihat pada Gambar 2.2.



Gambar 2. 4 Komponen Dasar Location Based Service (LBS) (Steigner, 2006)

Setiap komponen mempunyai fungsi (Steigner, 2006).

1. *Mobile Devices*, merupakan suatu alat yang digunakan oleh pengguna untuk meminta informasi yang dibutuhkan. Informasi dapat diberikan dalam bentuk suara, gambar, dan *text*.
2. *Communication Network*, komponen ini mengirim data pengguna dan informasi yang diminta dari *Mobile* terminal ke *Service Provider* kemudian mengirimkan kembali informasi yang diminta ke pengguna. *Communication network* dapat berupa jaringan *seluler* (GSM, CDMA), *Wireless Local Area Network* (WLAN), atau *Wireless Wide Area Network* (WWAN).
3. *Positioning Component*, digunakan untuk memproses suatu layanan maka posisi pengguna harus diketahui.
4. *Service* dan *Application Provider*, penyedia layanan menawarkan berbagai macam layanan kepada pengguna dan bertanggung jawab untuk memproses informasi yang diminta oleh pengguna.
5. Data dan *Content Provider*, penyedia layanan tidak selalu menyimpan semua data yang dibutuhkan yang bisa diakses oleh pengguna. Untuk itu, *data* dapat diminta dari data dan *content provider*.

Pada penelitian ini memanfaat Layanan Berbasis Lokasi atau yang kita kenal dengan *LBS* merupakan layanan informasi yang memanfaatkan kemampuan untuk menggunakan informasi lokasi dari perangkat bergerak dan dapat diakses dengan

perangkat bergerak melalui jaringan telekomunikasi bergerak seperti smartphone yang akan dipakai pada penelitian ini.

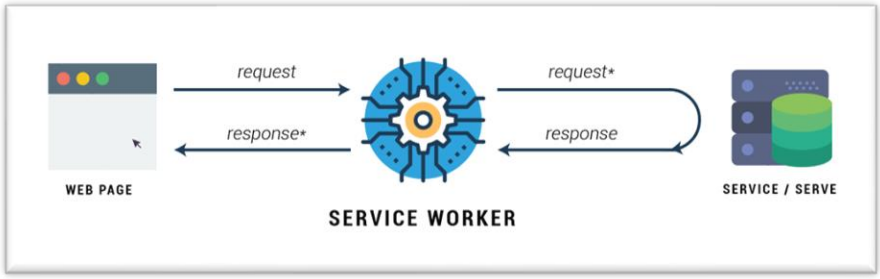
2.12 Progressive Web App (PWA)

Progressive Web App (PWA) adalah sebuah *website* yang dibangun menggunakan teknologi web modern, namun dapat berlaku seperti sebuah *mobile app*. Pada tahun 2015, *Google Engineer* Alex Russel dan Frances Berriman memberi istilah *Progressive Web App* pada konsep *web app* yang dapat memberikan *user experience* dalam keandalan *(reliability),* kecepatan *(speed)* dan keterlibatan pengguna *(user engagement).* Pada penelitian ini *Progressive Web App* (PWA*)* merupakan teknologi *modern* yang dapat diakses seperti aplikasi mobile pada umumnya tentu inikan mempermudah pihak truk ekspedisi yang akan menggunakannya Santoso (2019).

2.12.1 Service Worker

Service Worker merupakan salah satu jenis *web worker, javascript* yang berjalan di belakang layar (*background*) tanpa mempengaruhi kinerja halaman *web*. *Service Worker* pada dasarnya adalah *file javascript* yang berjalan *client side* secara terpisah dari rangkaian *browser* utama yang berfungsi untuk mencegah permintaan jaringan, melakukan *cache* atau mengambil sumber daya dari *cache* dan mengirimkan pesan.

Secara teknis, *Service Worker* menyediakan *script "network proxy"* di *web browser* untuk mengelola permintaan *web* (HTTP *request*) secara terprogram. *Service Worker* menggunakan mekanisme *cache* secara efisien dan memungkinkan perilaku *error-free* selama periode *offline*. Mekanisme kerja dari *Service Worker* dapat dilihat pada Gambar 2.3.



Gambar 2. 5 Mekanisme Kerja Service Worker

Pada gambar diatas dijelaskan bahwa mekanisme kerja dari *service worker* yaitu menerima *request* dari halaman *web* kemudian mengirimkan ke *server*. Selanjutnya *service worker* bertugas menerima *response* dari *server* untuk diteruskan kembali ke halaman *web*, sistem *service worker* akan membantu aplikasi ini agar bisa di akses dalam keadaan *offline* sekalipun.

2.12.2 Kelebihan dan Kekurangan PWA

Kemunculan teknologi PWA membawa pengaruh positif pada pengembangan aplikasi web. Namun aplikasi ini masih terbilang baru dan masih perlu pengembangan lebih lanjut. Berikut ini akan dibahas kelebihan dan kekurangan (atau keterbatasan saat ini) dari *Progressive Web App*.

1. Kelebihan PWA:
   - *Respon* Seperti *Mobile App*
   - *Layout Responsive*
   - Tersedia di *Mode Offline*
   - *Add to Home Screen* (A2HS)
   - *Push Notification*
2. Keterbatasan PWA:
   - Perlu *HTTPS*
   - Dukungan *Web Browser Modern*
   - Dukungan Pengguna

2.12.3 Karakteristik PWA

1. *Progressive:*

   Bekerja sama di semua *user*, tanpa melihat *web browser* yang dipakai, karena telah memiliki peningkatan progresif pada prinsipnya
2. *Responsive:*

   Cocok di segala bentuk *device*, (*desktop, mobile, tablet* lainya)
3. *Connectivity independent:*

   Ditingkatkan dengan *service worker* untuk bekerja secara *online* atau pada jaringan internet kualitas rendah
4. *App-like:*

   Terasa seperti *app*, karena model *App Shell* akan memisahkan fungsionalitas aplikasi dari kontennya
5. *Fresh:*

   Selalu *update* berkat adanya proses *update service worker*
6. *Safe:*

   Dilayani oleh HTTPS yang mencegah pengintaian (*snooping*) dan memastikan konten tidak rusak
7. *Discoperable:*

   Terientifikasi sebagai "*application*" berkat *manifest* W3C dan *registrasi service worker*, sehingga memungkinkan *search engine* untuk mengenalinya
8. *Re-engageable:*

   Memudahkan keterlibatan *user* dengan *fitur push notification.*
9. *Installable:*

   Memungkinkan user untuk menambahkan *apps* yang sering digunakan di layar home *screen* tanpa harus ke *app store*.
10. *Linkable:*

    Share aplikasi dengan mudah melalui *Uniform Resource Locator* (URL), tanpa repot-repot menginstalnya.

2.13 Model Proses Waterfall

"*Classic Life Cycle*" atau "*Linear Sequential Model*" atau lebih dikenal dengan model *waterfall* adalah sebuah model yang muncul pertama kali pada sekitar tahun 1970. Model waterfall merupakan salah satu model yang sering

dianggap kuno, tetapi merupakan model yang paling banyak di gunakan dalam *Software Enginering* (SE).

Pendekatan model waterfall dilakukan secara sistematis dan urut mulai dari level kebutuhan sistem sampai dengan tahap analisi, desain, coding, testing atau *verification* dan yang terakhir *maintenance* atau perawatan. Dimodelkan setelah siklus rekayasa konvensional, model sekuensial linier melingkupi aktivitas-aktivitas sebagai berikut (Pressman, 2002):

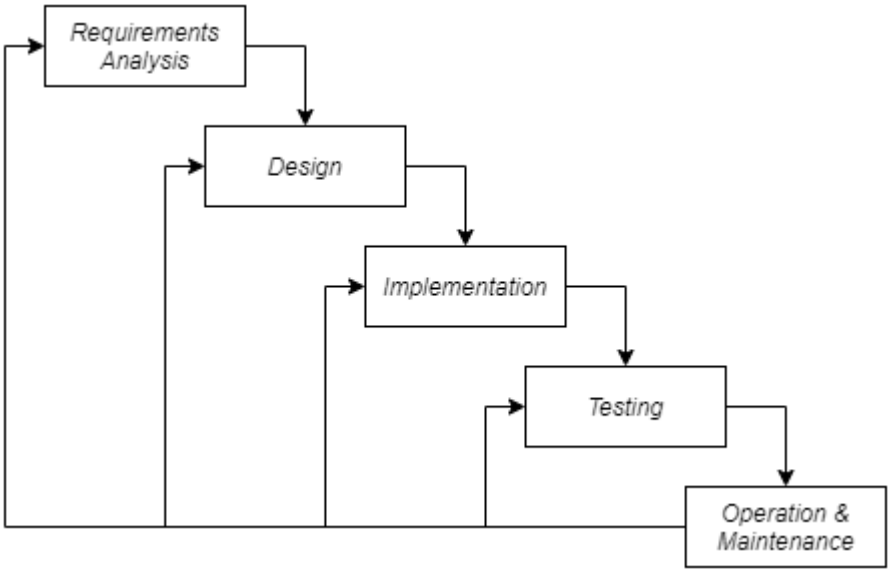
a. Analisis Kebutuhan *(Analysis Requirement)*
b. Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada perangkat lunak, untuk memahami sifat program yang dibangun, perekayasa perangkat lunak (analis) harus memahami domain informasi, tingkah laku, unjuk kerja dan antarmuka (*interface*) yang diperlukan.
c. Desain *(Design)*

   Desain perangkat lunak sebenarnya adalah proses *multi* langkah yang berfokus pada empat atribut sebuah program yang berbeda; struktur data, arsitektur perangkat lunak, representasi *interface*, dan detail (algoritma) prosedural.
d. Implementasi *(Implementation)*

   Desain harus diterjemahkan ke dalam bentuk mesin yang bisa dibaca. Dalam penelitian menggunakan bahasa pemrograman PHP untuk menterjemahkan perintah yang akan dieksekusi kedalam bahasa mesin.
e. Pengujian (*Testing)*

   Proses pengujian berfokus pada logika internal perangkat lunak, memastikan bahwa semua pernyataan sudah diuji dan pada eksternal fungsional, yaitu mengarahkan pengujian untuk menemukan kesalahan-kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.
f. Pemeliharaan *(Maintenance)*

   Pemeliharaan perangkat lunak mengaplikasikan lagi setiap fase program sebelumnya dan tidak membuat yang baru lagi. Fase-fase dalam *Waterfall* Model menurut referensi Pressman (2002):



Gambar 2. 6 Waterfall Model Pressman

2.14 Data Flow Diagram (DFD)

Menurut Kristanto (2008), *Data Flow Diagram (DFD)* merupakan suatu model logika data atau proses yang dibuat untuk menggambarkan dari mana asal data dan kemana tujuan data yang keluar dari sistem, di mana data disimpan, proses apa yang menghasilkan data tersebut dan interaksi antara data yang tersimpan dan proses yang dikenakan pada data tersebut. *Data Flow Diagram* atau dalam bahasa Indonesia menjadi Diagram Alir Data (DAD) adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengatur dari masukan dan keluaran.

Berdasarkan kedua pendapat di atas, dapat disimpulkan bahwa *Data Flow Diagram (DFD)* merupakan gambaran dalam bentuk diagram grafik yang menjelaskan tentang aliran data atau informasi yang berjalan pada suatu sistem.
Beberapa simbol yang terdapat pada *Data Flow Diagram (DFD)* sebagai berikut:

a. Kesatuan Luar *(External Entity)*

   Sesuatu yang berada di luar sistem, tetapi memberikan data ke dalam sistem atau memberikan data dari sistem. *External Entity* tidak termasuk bagian dari sistem. Bila sistem informasi dirancang untuk satu bagian atau departemen maka bagian lain yang masih terkait menjadi *external entity*.

Gambar 2. 7 Simbol Entitas

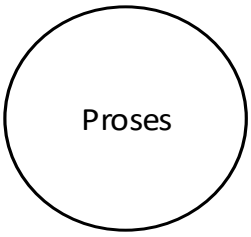
b. Arus Data *(*Data *Flow)*

Arus data merupakan tempat mengalirnya informasi dan digambarkan dengan garis yang menghubungkan komponen dari sistem. Arus data ditunjukkan dengan arah panah dan garis diberi nama atas arus data yang mengalir. Arus data ini mengalir di antara proses, data store dan menunjukkan arus data dari data yang berupa masukan untuk sistem atau hasil proses sistem.

Gambar 2. 8 Simbol Arus Data

c. Proses *(Process)*

Proses merupakan apa yang dikerjakan oleh sistem. Proses dapat mengolah data atau aliran data masuk menjadi aliran data ke luar. Proses berfungsi mengubah satu atau beberapa data masukan menjadi satu atau beberapa data keluaran sesuai dengan spesifikasi yang diinginkan. Setiap proses memiliki satu atau beberapa masukan serta menghasilkan satu atau beberapa keluaran.



Gambar 2. 9 Simbol Proses

d. Simpanan Data *(Data Store)*

Simpanan data merupakan tempat penyimpanan data pengikat data yang ada dalam sistem. *Data store* disimbolkan dengan sepasang dua garis atau dua garis

dengan salah satu sisi samping terbuka. Proses dapat mengambil data dari atau memberikan data ke basis data.

Gambar 2. 10 Simbol Simpanan Data

2.15 Entity Relationship Diagram (ERD)

Menurut Al-Bahra bin Ladjamudin (2005:142) *Entity Relationship Diagram* (ERD) adalah suatu model jaringan yang menggunakan susunan data yang disimpan dalam sistem secara abstrak. Jadi, jelaslah bahwa ERD ini berbeda dengan DFD yang merupakan suatu model jaringan fungsi yang akan dilaksanakan oleh sistem, sedangkan ERD merupakan model jaringan data yang menekankan pada struktur–struktur dan relationship data. Elemen–elemen dalam *Diagram* Hubungan Entitas meliputi:

a. Entity

Pada E-R *Diagram*, *Entity* dapat digambarkan dengan sebuah bentuk persegi panjang. *Entity* adalah sesuatu apa saja yang ada di dalam sistem, nyata maupun abstrak dimana data tersimpan atau dimana terdapat data. Entitas diberi nama dengan kata benda dan dapat dikelompokkan dalam empat jenis nama, yaitu orang, benda, lokasi, kejadian (terdapat unsur waktu di dalamnya).

b. Relationship

Pada E-R *Diagram*, *Relationship* dapat digambarkan dengan sebuah bentuk belah ketupat. *Relationship* adalah hubungan alamiah yang terjadi antara entitas. Pada umumnya penghubung (*Relationship*) diberi nama dengan kata kerja dasar, sehingga memudahkan untuk melakukan pembacaan relasinya (bisa dengan kalimat aktif atau pasif).

Penggambaran hubungan yang terjadi adalah sebuah bentuk belah ketupat dihubungkan dengan dua bentuk empat persegi panjang.

1. *Relationship Degree*

Relationship Degree atau Derajat *Relationship* adalah jumlah entitas yang berpartisipasi dalam suatu *Relationship*.

a. *Unary Degree* (Derajat Satu)

Gambar 2. 11 *Unary Degree*

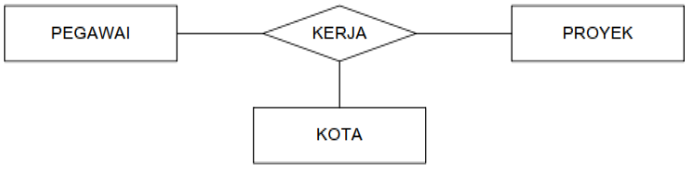
Sumber: Al-Bahra bin Ladjamudin (2005:142)

b. *Binary Degree* (Derajat Dua)

Gambar 2. 12 *Binary Degree*

Sumber: Al-Bahra bin Ladjamudin (2005:142)

c. *Ternary Degree* (Derajat Tiga)



Gambar 2. 13 *Ternary Degree*

Sumber: Al-Bahra bin Ladjamudin (2005:142)

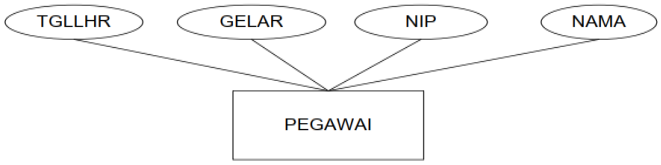
2. Atribut

Secara umum atribut adalah sifat atau karakteristik dari tiap entitas maupun tiap *Relationship*. Atribut adalah sesuatu yang menjelaskan apa sebenarnya yang dimaksud entitas maupun *Relationship*, sehingga sering dikatakan atribut adalah elemen dari setiap entitas dan *Relationship*. Atribut *Value* atau Atribut Nilai adalah suatu *occurrence* tertentu dari sebuah atribut di dalam suatu *entity* atau *Relationship*.

Ada dua jenis-jenis atribut:

1. *Key* adalah atribut yang digunakan untuk menentukan suatu *entity* secara unik.
2. Atribut *Simple* adalah atribut yang bernilai tunggal.

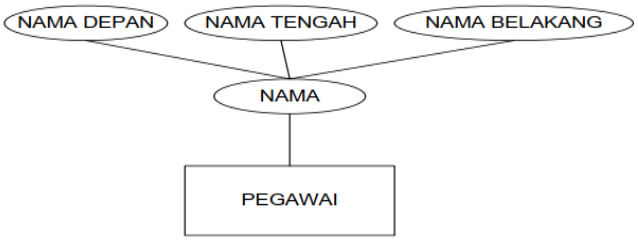
2. Atribut *Multivalue* adalah atribut yang memiliki sekelompok nilai untuk setiap *instan entity*



Gambar 2. 14 *Atribut Multivalue*

Sumber: Al-Bahra bin Ladjamudin (2005:142)

3. Atribut *Composite* adalah suatu atribut yang terdiri dari beberapa atribut yang lebih kecil yang mempunyai arti tertentu.



Gambar 2. 15 *Atribut Composite*

Sumber: Al-Bahra bin Ladjamudin (2005:142)

3. Kardinalitas

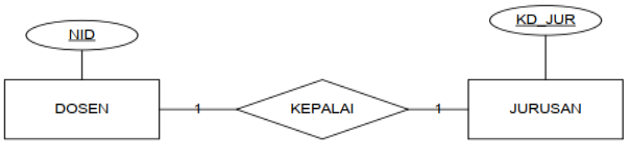
Kardinalitas Relasi menunjukkan jumlah maksimum tupel yang dapat berelasi dengan entitas pada entitas yang lain. Dari sejumlah kemungkinan banyaknya hubungan antra entitas tersebut, Kardinalitas Relasi merujuk kepada hubungan maksimum yang terjadi dari entitas yang satu ke entitas yang lain dan begitu juga sebaliknya. Terdapat tiga macam kardinalitas relasi, yaitu:

a. *One to One*

Tingkat hubungan satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas kedua dan sebaliknya. Yang berarti setiap tupel pada entitas A berhubungan

dengan paling banyak satu tupel pada entitas B, dan begitu juga sebaliknya setiap tupel pada entitas B berhubungan dengan paling banyak satu tupel pada entitas A.

Contoh: Adanya relasi antara entitas Dosen dengan entitas Jurusan. Relasinya diberi nama ‘Kepala’. Pada relasi ini, setiap dosen paling banyak mengepalai satu jurusan (walaupun memang tidak semua dosen yang menjadi ketua jurusan). Dan setiap jurusan dikepalai oleh paling banyak satu orang dosen.



Gambar 2. 16 Diagram Kardinalitas *One to One*

Sumber: Al-Bahra bin Ladjamudin (2005:142)

b. *One to Many* atau *Many to One*

Tingkat hubungan satu ke banyak adalah sama dengan banyak ke satu. Tergantung dari arah mana hubungan tersebut dilihat. Untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua. Sebaliknya, satu kejadian pada entitas yang kedua hanya dapat mempunyai satu hubungan dengan satu kejadian pada entitas yang pertama

c. *One to Many*

Satu tupel pada entitas A dapat berhubungan dengan banyak tupel pada entitas B, tetapi tidak sebaliknya, di mana setiap tupel pada entitas B, berhubungan dengan paling banyak satu tupel pada entitas A.

Contoh: Adanya relasi antara entitas Dosen dengan entitas Kuliah. Relasinya diberi nama ‘Ajar’. Setiap dosen dapat mengajar lebih dari satu mata kuliah, sedang setiap mata kuliah hanya oleh paling banyak satu orang dosen.

Gambar 2. 17 Diagram Kardinalitas *One to Many*

Sumber: Al-Bahra bin Ladjamudin (2005:142)

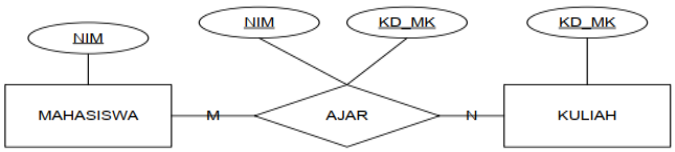
d. *Many to One*

Setiap tupel pada entitas A dapat berhubungan dengan paling banyak satu tupel pada entitas B, tetapi tidak sebaliknya, di mana setiap tupel pada entitas A berhubungan dengan paling banyak satu tupel pada entitas B.

e. *Many to Many*

Tingkat hubungan banyak ke banyak terjadi jika setiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainya. Baik dilihat dari sisi entitas yang pertama, maupun dilihat dari sisi yang kedua. Yang berarti setiap tupel pada entitas A dapat berhubungan dengan banyak tupel pada entitas B, dan demikian juga sebaliknya, dimana setiap tupel pada entitas B dapat berhubungan dengan banyak tupel pada entitas A.

Contoh: Adanya relasi entitas Mahasiswa dengan entitas Kuliah. Relasinya diberi nama ‘Belajar’. Setiap mahasiswa dapat mempelajari lebih dari satu mata kuliah. Demikian juga sebaliknya, setiap mata kuliah dapat dipelajari oleh lebih dari satu orang mahasiswa.

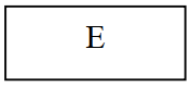


Gambar 2. 18 Diagram Kardinalitas *Many to Many*

Sumber: Al-Bahra bin Ladjamudin (2005:142)

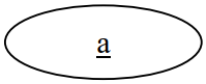
Notasi–notasi simbolik di dalam Diagram E-R:

1. Persegi panjang, menyatakan Himpunan Entitas/entitas.



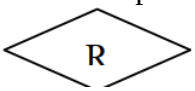
Himpunan Entitas/entitas E

2. Lingkaran/Elip, menyatakan Atribut (Atribut yang berfungsi sebagai key digaris bawahi).



Atribut a sebagai *key*

3. Belah ketupat menyatakan Himpunan Relasi/relasi.



Himpunan Relasi/relasi R

4. Garis, sebagai penghubung antara Himpunan Relasi dengan Himpunan Entitas dan Himpunan Entitas dengan Atributnya.

Link

Langkah–langkah teknis untuk menghasilkan *Entity Relationship Diagram* (ERD):

a. Melengkapi himpunan entitas dan himpunan relasi dengan atribut-atribut deskriptif (*non key*).
b. Mengidentifikasi dan menetapkan seluruh entitas yang akan terlibat.
c. Menentukan atribut-atribut key (*primary key*) dari masing-masing entitas.
d. Mengidentifikasi dan menetapkan seluruh relasi diantara entitas-entitas yang ada beserta *foreign-key*-nya (jika terjadi kardinalitas relasi *One to Many atau Many to Many*)
e. Menentukan derajat/kardinalitas relasi untuk setiap relasi.

2.16 Teknologi Pendukung

2.16.1 Web

World Wide Web (WWW) atau biasa disebut dengan web merupakan salah satu sumber daya Internet yang berkembang pesat. Pertama kali aplikasi web dibangun hanya dengan menggunakan bahasa yang disebut HTML (*HyperText Markup Language*) dan protokol yang digunakan dinamakan HTTP (*HyperText Transfer Protocol*). Pada perkembangan berikutnya, sejumlah skrip dan objek dikembangkan untuk memperluas kemampuan HTML yang sekarang ini terdapat banyak skrip seperti: PHP dan ASP, sedangkan contoh yang berupa objek antara

lain adalah *applet (java*). Jadi aplikasi *web* atau aplikasi berbasis *web* (*Web-based application*) adalah aplikasi untuk menyampaikan informasi kepada pengguna yang menggunakan layanan *Internet* berbasis *web*.

Dalam aplikasi tersebut, terjadi pertukaran antara *klien* (komputer yang meminta informasi) dengan *server* (komputer yang memasok atau menanggapi informasi). *Web* memberikan informasi secara *online* melalui internet langsung. *Klien* melakukan permintaan informasi dengan menggunakan *browser* (contoh *browser: Internet Explorer, Opera, Mozilla*, dan sebagainya). *Server* menerima informasi dan melayani permintaan dari *client*. Hal ini biasa disebut dengan web *server* (contoh *web server: Apache, IIS, Xitami,* dan sebagainya). Setelah itu, web *server* akan berkomunikasi dengan *middleware* (contoh *middleware*: ASP, JSP, PHP, dan sebagainya) untuk bisa berhubungan dengan basis data atau *database* (contoh *database: access, oracle, sql*, dan sebagainya). Setelah berinteraksi dengan *database*, *server* yang telah mendapatkan informasi akan memberikan tanggapan terhadap *klien* yang meminta informasi tadi Kadir (2005).

2.16.2 XAMPP

XAMPP merupakan paket PHP dan MySQL berbasis *open source*, yang digunakan sebagai alat pembantu pengembangan aplikasi berbasis PHP. XAMPP mengkombinasikan beberapa paket perangkat lunak berbeda dalam satu paket. Di dalam Paket XAMPP terdapat tiga paket penting yaitu *Apache* sebagai *web server*, PHP sebagai bahasa pemrograman dan MySQL sebagai *database*. *Apache* adalah *server web* (*web server*) yang dapat dijalankan di banyak sistem operasi.

Apache merupakan perangkat lunak open source yang dikembangkan oleh komunitas terbuka yang terdiri dari pengembang-pengembang dibawah naungan *Apache Software Foundation.*

2.16.3 MYSQL

MySQL adalah *software* atau program aplikasi database, yaitu *software* yang dapat kita pakai untuk menyimpan data berupa informasi teks dan juga angka. Semua informasi data yang dipakai di dalam satu program aplikasi disimpan dalam satu *software* database, yaitu MySQL.

Dengan menggunakan database MySQL, maka data yang tersimpan di dalam database dapat diakses secara bersama menggunakan beberapa komputer/laptop yang berbeda, konsep ini sering disebut dengan sistem *multi user.* Database MySQL juga dapat diinstal pada sebuah komputer pusat *(server)* yang tersimpan di dalam ruang *server*, kemudian datanya diakses melalui komputer/laptop yang terinstal dengan program *client* seperti program kasir toko atau program akademik siswa yang letaknya berbeda tempat, dan konsep tersebut disebut dengan sistem *client/server* (Nugroho, 2014).

2.16.4 Javascript

Javascript adalah bahasa *scripting* yang populer di internet dan dapat bekerja di sebagian besar *browser* populer seperti *Internet Explorer* (IE), *Mozilla Firefox*, *Netscape* dan *Opera*. Kode *Javascript* dapat disisipkan dalam halaman *web* menggunakan *tag SCRIPT* (Sunyoto,2007). Beberapa hal tentang *Javascript:*

1. *Javascript* didesain untuk menambah interaktif suatu *web*.
2. *Javascript* merupakan sebuah bahasa *scripting.*
3. Bahasa *scripting* merupakan bahasa pemrograman yang ringan.
4. *Javascript* berisi baris kode yang dijalankan di komputer (*web browser*).
5. *Javascript* biasanya disisipkan (*embedded*) dalam halaman HTML.
6. *Javascript* adalah bahasa *interpreter* (yang berarti *script* dieksekusi tanpa proses kompilasi).

2.16.5 HTML

HTML merupakan halaman yang berada pada suatu situs internet atau *web*. HTML merupakan metode yang menautkan (*link*) satu dokumen ke dokumen lain melalui teks. Menurut Deris Setiawan, HTML merupakan framework internet, hampir semua situs web yang ada menggunakan HTML untuk menampilkan teks, grafik, suara, dan animasinya (Kuswayatno, 2006). HTML adalah suatu bahasa yang dikenali oleh web browser untuk menampilkan informasi dengan lebih menarik dibandingkan dengan tulisan teks biasa (*plain text*) (Oktavian 2010).

HTML adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman *web*, menampilkan berbagai informasi di dalam sebuah penjelajah *web*

Internet dan pemformatan hiperteks sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi. Dengan kata lain, berkas yang dibuat dalam perangkat lunak pengolah kata dan disimpan dalam format ASCII normal sehingga menjadi halaman web dengan perintah-perintah HTML. Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan HTML (*Standard Generalized Markup Language*), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman *web.* HTML saat ini merupakan standar Internet yang didefinisikan dan dikendalikan penggunaannya oleh *World Wide Web Consortium (*W3C*).* HTML dibuat oleh kolaborasi Caillau TIM dengan Berners-lee Robert ketika mereka bekerja di *CERN* pada tahun 1989 (*CERN* adalah lembaga penelitian fisika energi tinggi di Jenewa).

2.16.6 Bootstrap

Bootstrap merupakan sebuah *framework* CSS yang memudahkan pengembang untuk membangun website yang menarik dan responsif. *Bootstrap* adalah CSS tetapi dibentuk dengan LESS, sebuah *pre-prosessor* yang memberi fleksibilitas dari penggunaan CSS biasa. *Bootstrap* dapat dikembangkan dengan tambahan lainnya karena ini cukup fleksibel terhadap pekerjaan *web* yang mengutamakan desain (Otto, 2011).

2.16.7 PHP

PHP merupakan bahasa scripting *server-side*, dimana pemrosesan datanya dilakukan pada sisi *server*. Sederhananya, *server* lah yang akan menerjemahkan *script program*, baru kemudian hasilnya akan dikirim kepada *client* yang melakukan permintaan.

PHP adalah bahasa yang sederhana namun kuat dirancang untuk membuat konten HTML. Sejak dimulai pada tahun 1994, PHP telah mampu menyaingi bahasa *web* karena keunggulan popularitas bahasa dan kemudahan penggunaan. PHP pertama kali dibuat oleh Rasmus Lerdorf. PHP sering digunakan untuk membangun web dinamis dimana proses keseluruhan berjalan pada *web server* dan menampilkan hasilnya pada web browser (Prasetyo, 2004).

PHP adalah bahasa pemrograman untuk membuat situs web dinamis dan interaktif. PHP berjalan di *server web* dan melayani pengunjung dengan halaman web sesuai permintaan (Doyle, 2009).

2.17 Pengujian Perangkat Lunak

Pengujian perangkat lunak merupakan suatu teknik yang digunakan menguji apakah sebuah perangkat lunak yang dihasilkan telah sesuai dengan yang diharapkan atau belum. Menurut Pressman (2002), pengujian adalah proses eksekusi suatu program untuk menemukan kesalahan sebelum digunakan oleh pengguna akhir (*end-user*).

2.17.1 Pengujian Black Box

Black Box testing berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineers* untuk memperoleh set kondisi input yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program (Pressman, 2010).

Black Box Testing berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program. *Black Box Testing* bukanlah solusi alternatif dari White Box Testing tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh White Box Testing. Black Box Testing cenderung untuk menemukan hal-hal berikut (Mustaqbal dan Firdaus, 2015).

1. Fungsi yang tidak benar atau tidak ada
2. Kesalahan antarmuka (*interface errors*)
3. Kesalahan pada struktur data dan akses basis data
4. Kesalahan performansi (*performance errors*)
5. Kesalahan inisialisasi dan terminasi

Adapun teknik pengujian secara *black box* menurut (Sukamto, 2009), adalah sebagai berikut:

1. *Boundary Value Analysis* (BVA) / *Limit Testing* adalah banyak kesalahan terjadi pada kesalahan masukan. BVA mengijinkan untuk menguji seleksi kasus uji yang menguji batasan nilai input. BVA merupakan komplemen

dari *equivalence partitioning*. Lebih pada memilih elemen-elemen didalam kelas ekivalen pada bagian sisi batas dari kelas.

2. *Comparison Testing* adalah pengujian pada beberapa aplikasi *reliability* dari sebuah perangkat lunak. Redundansi perangkat keras dan perangkat lunak mungkin digunakan untuk meminimalisir kesalahan (*error*). Untuk redundansi perangkat lunak, gunakan tim yang terpisah untuk mengembangkan setiap versi perangkat lunak yang independen. Uji setiap versi dengan data yang sama untuk memastikan semua versi menghasilkan keluaran yang sama. Jalankan semua versi dengan paralel dan perbandingan keluaran secara *real-time*.
3. *Sample Testing* adalah pengujian yang melibatkan beberapa nilai yang terpilih dari sebuah kelas ekuivalen. Mengintegrasikan nilai pada kasus uji dan nilai-nilai yang terpilih mungkin dipilih dengan urutan tertentu atau interval tertentu
4. *Robustness Testing* adalah pengujian dengan data input dipilih diluar spesifikasi yang telah didefinisikan. Tujuan dari pengujian ini adalah membuktikan bahwa tidak ada kesalahan jika masukan tidak valid
5. *Behavior Testing* adalah pengujian dengan hasil uji tidak dapat dievaluasi jika hanya melakukan pengujian sekali, tapi dapat dievaluasi jika pengujian dilakukan beberapa kali, misalnya pada pengujian struktur *data stack*.
6. *Requirement Testing* adalah spesifikasi kebutuhan yang terasosiasi dengan perangkat lunak (*input, output*, fungsi, performansi) diidentifikasi pada tahap spesifikasi kebutuhan dan desain. *requirement testing* melibatkan pembuatan kasus uji untuk setiap spesifikasi kebutuhan yang terkait dengan program.
7. *Performance Testing* adalah mengevaluasi kemampuan program untuk beroperasi dengan benar dipandang dari sisi acuan kebutuhan misalnya: aliran data, ukuran pemakaian memori, kecepatan eksekusi dan lain-lain. Untuk mencari tahu beban kerja atau kondisi konfigurasi program dan dapat digunakan untuk menguji batasan lingkup program.
8. Uji Ketahanan (*Endurance Testing*) adalah melibatkan kasus uji yang diulang-ulang dengan jumlah tertentu dengan tujuan untuk mengevaluasi program apakah sesuai spesifikasi kebutuhan.
9. *Equivalence partitioning* adalah membagi input menjadi kelas-kelas data yang

dapat digunakan untuk meregenerasi kasus uji dengan tujuan untuk menemukan kelas-kelas kesalahan. Selain itu, *equivalence partitioning* berdasarkan pada kesamaan kelas-kelas kondisi *input*. Sebuah kelas yang ekuivalen merepresentasikan kumpulan status/kondisi yang valid atau tidak valid. Sebuah kondisi input dapat berupa nilai numerik yang spesifik, rentan nilai, kumpulan nilai yang berkaitan, atau kondisi *boolean*.

10. Uji Sebab-Akibat (*Cause-Effect Relationship Testing*) adalah teknik yang merupakan suplemen dari *equivalence testing* dengan menyediakan cara untuk memilih kombinasi data input dan melibatkan kondisi *input* (*Cause*) dan kondisi *output* (*Effect*) untuk mencegah pendefinisian kasus uji yang terlalu banyak.

2.17.2 Pengujian Skala Likert

Perhitungan hasil dari kuesioner dilakukan dengan menggunakan cara *Skala Likert*. *Skala Likert* digunakan untuk mengukur sikap, pendapat, dan persepsi seseorang atau sekelompok orang tentang fenomena sosial (Sugiyono, 2014). Untuk setiap pilihan jawaban diberi skor, maka responden harus menggambarkan, mendukung pertanyaan dengan jawaban yang dipilih. Dengan *skala likert*, variabel yang akan diukur dijabarkan menjadi indikator variabel. Kemudian indikator tersebut dijadikan sebagai titik tolak ukur menyusun item-item instrumen yang dapat berupa pertanyaan atau pernyataan.

Tabel 2. 4 Skala Penilaian Untuk Pertanyaan Positif dan Negatif

Nilai	Kriteria
1	Sangat Buruk
2	Buruk
3	Cukup
4	Baik
5	Sangat Baik