BAB II
TINJAUAN PUSTAKA

2.1 Teori Dasar

2.1.1 Masjid

Masjid atau mesjid adalah rumah tempat ibadah umat Islam atau Muslim. Masjid artinya tempat sujud, dan sebutan lain bagi masjid di Indonesia adalah musholla, langgar atau surau. Istilah tersebut diperuntukkan bagi masjid yang tidak digunakan untuk Sholat Jum'at, dan umumnya berukuran kecil. Selain digunakan sebagai tempat ibadah, masjid juga merupakan pusat kehidupan komunitas muslim. Kegiatan-kegiatan perayaan hari besar, diskusi, kajian agama, ceramah dan belajar Al Qur'an sering dilaksanakan di Masjid. Bahkan dalam sejarah Islam, masjid turut memegang peranan dalam aktivitas sosial kemasyarakatan hingga kemiliteran.(Wikipedia)

2.1.2 Android

Android adalah sistem operasi untuk perangkat mobile berbasis Linux yang mencakup sistem operasi, middleware, dan aplikasi. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat piranti lunak untuk ponsel. kemudian dalam pengembangan Android, dibentuklah *Open Handset Alliance, konsorsium* dari 34 perusahaan piranti keras, piranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, TMobile, dan Nvidia. Pada saat perilisan perdana Android, 5 November 2007, Android bersama *Open Handset Alliance* mendukung pengembangan *open source*pada perangkat *mobile*. Google juga merilis kode-kode Android di bawah lisensi Apache yaitu lisensi perangkat lunak dan open platform perangkat seluler. (Safaat H:2011)

2.1.3 Google Maps

Google Maps adalah sebuah jasa peta virtual gratis yang disediakan oleh Google. *Google Maps*menawarkan peta yang dapat diseret dan gambar satelit untuk seluruh dunia, juga menawarkan perencana rute dan pencari letak bisnis. Kita dapat menambahkan fitur *Google Maps* dalam web yang telah kita buat atau pada blog kita yang berbayar maupun gratissekalipun dengan *Google Maps API.Google Maps API* adalah suatu library yang berbentuk *JavaScript*. API sendiri adalah singkatan dari *Application Progrmming Interface*.

Perlu diketahui bahwa perkembangan penggunan *Google Maps* di Android yang ada saat ini dimulai dengan adanya *Google Maps* V1, yang penggunaannya telah dihentikan pada akhir tahun 2012. Mulai tahun 2013, aplikasi Android yang ingin menampilkan *Google Map* harus menggunakan layanan *Google Map* V2. Mulai dari penggunaan SHA1 yang menggantikan MD5 untuk mendapatkan *Google API Key* hingga penggunaan Fragment yang menggantikan *MapView*. Kita juga harus menginstal *library google-play-service*terlebih dahulu pada Android SDK. (Mufti:2015)

2.1.4 Sistem Informasi Geografis

Perkembangan teknologi komputer telah membuka wawasan dan paradigma baru dalam proses pengambilan keputusan dan penyebaran informasi. Data yang merepresentasikan ”dunia nyata” dapat disimpan dan diproses sedemikian rupa sehingga dapat disajikan dalam bentuk-bentuk yang lebih sempurna dan sesuai kebutuhan. Sejak pertengahan tahun 1970-an, telah dikembangkan sistem-sistem yang secara khusus dibuat untuk menangani masalah informasi yang bereferensi geografis dalam berbagai cara dan bentuk. Masalah-masalah ini mencakup : 1. Pengorganisasian data dan informasi 2. Penempatan informasi pada lokasi tertentu 3. Melakukan komputasi, memberikan ilustrasi keterhubungan satu sama lainnya (koneksi), beserta analisis-analisis spasial lainnya. Sebutan umum untuk sistem-sistem yang menangani masalah-masalah di atas adalah Sistem Informasi Geografis(SIG). Sistem Informasi Geografis adalah sistem komputer yang digunakan untuk memasukkan, menyimpan, memeriksa, mengintegrasikan, memanipulasi, menganalisa, dan menampilkan data yang berhubungan dengan posisi-posisi di permukaan bumi (Chang, 2008). Adapun komponen-komponen dari sistem informasi Seminar Nasional Aplikasi Teknologi Informasi 2010 (SNATI 2010) ISSN: 1907-5022 Yogyakarta, 19 Juni 2010 D-47 geografis adalah sistem komputer, software GIS, orang, data, dan infrastruktur. Pada awalnya, data geografis hanya disajikan di atas peta dengan menggunakan simbol, garis, dan warna. Elemen-elemen geometri ini dideskripsikan di dalam legendanya, misalnya garis hitam tebal untuk jalan utama, garis hitam tipis untuk jalan sekunder dan jalan-jalan berikutnya. Selain itu berbagai data juga dapat di-overlay-kan berdasarkan sistem koordinat yang sama. Akibatnya sebuah peta menjadi media yang efektif baik sebagai alat presentasi maupun sebagai bank tempat penyimpanan data geografis. (Prahasta, 2002). Bila dibandingkan dengan peta, SIG memiliki keunggulan karena penyimpanan data dan presentasinya dipisahkan. SIG menyimpan semua informasi deskriptif unsur-unsurnya sebagai atributatribut di dalam basisdata. Kemudian SIG

membentuk dan meyimpannya di dalam tabel-tabel (relasional). Setelah itu, SIG menghubungkan unsurunsur di atas dengan tabel-tabel yang bersangkutan. Dengan demikian, atribut-atribut ini dapat diakses melalui lokasi unsur-unsur peta, dan sebaliknya unsur-unsur peta juga dapat diakses melalui atributatributnya. Dengan demikian data dapat dipresentasikan dalam berbagai cara dan bentuk. (Prahasta, 2002)

2.1.5 PHP (*HyperText Preprocessor*)

PHP (*HyperText Preprocessor*) adalah sebuah bahasa utama *script* serverside yang disisipkan pada HTML yang dijalankan di server, dan juga bisa digunakan untuk membuat aplikasi desktop.

Menurut Betha Sidik, dalam bukunya yang berjudul Pemrograman Web Dengan *PHP* (2012 : 4), menyebutkan bahwa, *PHP* merupakan secara umum dikenal dengan sebagai bahasa pemrograman *script-script* yang membuat dokumen HTML secara *on the fly* yang dieksekusi di server web, dokumen HTML yang dihasilkan dari suatu aplikasi bukan dokumen HTML yang dibuat dengan menggunakan editor teks atau editor HTML, dikenal juga sebagai bahasa pemrograman *server side.*

2.1.6 Algoritma Floyd-Warshall

Algoritma Floyd-Warshall adalah sebuah algoritma analisis *graf* untuk mencari bobot minimum dari *graf* berarah. Dalam pengertian lain algoritma Floyd-Washall adalah suatu metode yang melakukan pemecahan suatu masalah dengan memandang solusi yang akan diperoleh sebagai suatu keputusan yang saling terkait. Artinya solusi-solusi tersebut dibentuk dari solusi yang berasal dari tahap sebelumnya dan ada kemungkinan solusi lebih dari satu.(Nur dan Setiawan:2013:21)

2.2 Referensi Jurnal

Menurut penelitian yang dilakukan oleh Wulandari,Pramono,dan Tajidun pada tahun 2017 dengan judul Aplikasi pencarian rute terpendek apotek di kota Kendari menggunakan algoritma floyd-warshallmenjelaskan bahwa untuk mencari rute terpendek dibutuhkan sebuah algoritma.Algoritma Floyd-Warshall dapat menghitung bobot terkecil dari semua rute yang menghubungkan pasangan titik dengan menghitung sekaligus bobot untuk semua rute yang mungkin dilewati.Algoritma ini efektif digunakan karena kesederhanaan dalam perhitungan matematik.(Wulandari:2017)

Menurut penelitian yang dilakukan oleh Ningrum dan Andrasto pada tahun 2016 dengan judul Penerapan Algoritma Floyd-Warshall dalam menentukan rute terpendek pada pemodelan jaringan pariwisata di kota Semarang. Dalam penelitian dijelaskan bahwa cara menerapkan Algoritma Floyd-Warshall dalam mencari rute terpendek untuk menyelesaikan suatu masalah(Ningrum:2016)

Menurut penelitian yang dilakukan oleh Triyanti dan Marleen pada tahun 2014 dengan judul Aplikasi android untuk pencarian lokasi tempat ibadah di wilayah Bekasi. Dijelaskan bahwa Pembuatan aplikasi ini yang berupa Augmented Reality (AR) untuk pencarian lokasi tempat ibadah dengan platform Layar untuk mendukung aplikasi tersebut.Hasil dari penelitian ini berupa aplikasi yang berjalan pada smartphone sistem operasi android, aplikasi ini diharapkan dapat memberikan informasi secara cepat untuk mencari tempat ibadah yang diinginkan oleh pemakai(Triyanti:2014)

Menurut penelitian yang dilakukan oleh Aquarizki, Irawan, dan Setianingsih. Pada tahun 2017 dengan judul perancangan dan implementasi aplikasi pencarian rute optimal untuk pemadam kebakaran berbasis android menggunakan algoritma Floyd-Warshall dijelaskan bahwa Algoritma Floyd-Warshall memberikan solusi pencarina rute yang optimal dengan memperhatikan kondisi jalan seperti kemacetan,ruas jalan.Proses pencarian didasarkan pada perhitungan jarak tiap simpul dan memilih jarak terkecil antar titik.(Aquarizky:2017).

Menurut penelitian yang dilakukan oleh Yusuf,Zahra,dan Apriyanti pada tahun 2017 dengan judul Implementasi algoritma Dijkstra dalam menemukan jarak terdekat dari lokasi pengguna ke tanaman yang dituju berbasis android dijelaskan bahwa dalam pembuatan aplikasi android dibutuhkan suatu metode/algoritma untuk melakukan perhitungan guna mendapat jarak terdekat. (Zahra: 2017).

2.3 Unified Modeling Language (UML)

Menurut Verdi (2012) dalam bukunya *Unified Modeling Language* (UML) adalah notasi bahasa pemodelan yang lengkap untuk membuat visualisasi suatu sistem atau perangkat lunak yang berorientasi objek. UML disebut sebagai bahasa pemodelan bukan sebagai metode. Bahasa pemodelan merupakan notasi dari metode yang digunakan untuk mendesain secara cepat.Menentukan bahasa pemodelan adalah cara untuk berdiskusi tentang desain dengan seseorang. Tujuan dari *Unified Modeling Language* (UML) diantara lain sebagai berikut:

1. Memodelkan suatu sistem (bukan hanya perangkat lunak) yang menggunakan konsep berorientasi objek.
2. Menciptakan suatu bahasa pemodelan yang dapat digunakan baik oleh manusia maupun mesin.
3. Memberikan bahasa yang bebas dari berbagai bahasa pemrograman.

Sebuah bahasa telah menjadi standar dalam industri untuk visualisasi, merancang, dan mendokumentasikan sistem piranti lunak. Denotasi yang lengkap untuk membuat visualisasi model suatu sistem. Sistem berisi informasi dan fungsi, tetapi yang secara normal digunakan untuk memodelkan sistem komputer. Keuntungan menggunakan *Unified Modeling Language* (UML), adalah sebagai berikut:

1. Software terdesain dan terdokumentasi secara professional sebelum dibuat.
2. Desain yang dibuat terlebih dahulu membuat reusable code dapat dikode dengan tingkat efisiensi yang tinggi.
3. Dengan membuat UML dapat melihat gambaran besar dari suatu software.

UML menjanjikan akan menghasilkan hasil dengan biaya rendah, software lebih efisien, lebih dapat dipercaya, dan hubungan antar bagian yang terlibat menjadi lebih baik.UML merupakan sintaks umum untuk membuat model logika dari suatu sistem dan digunakan untuk menggambarkan sistem agar dapatdipahami selama fase analisis dan desain. UML biasanya disajikan dalam bentuk diagram atau gambar yang meliputi class beserta atribut dan operasinya, serta hunbungan antar class yang meliputi inheritance, association dan komposisi. UML terdiri dari banyak diagram antara lain sebagai berikut (Sugiarti, 2013).

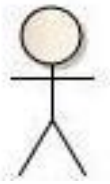
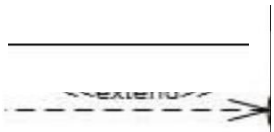
2.3.1 UseCase Diagram

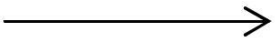
Use case diagram adalah gambaran dari beberapa atau seluruh aktor dan use case dengan tujuan mengenali interkasi mereka dalam suatu sistem. Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, yang ditentukan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interkasi antara aktor dengan sistem.Syarat penamaan pada use case adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada use case yaitu pendefinisian apa yang disebut aktor dan use case yaitu:

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.

2. Use case merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor (Sugiarti, 2013). Simbol-simbol yang digunakan pada use case diagram ditunjukkan pada tabel 2.1.

Tabel 2.1. Simbol pada *use case diagram* (Sugiarti, 2013).

Nama	Simbol	Deskripsi
Use Case		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor, biasanya dinyatakan dengan menggunakan kata kerja di awal di frase nama Use Case.
Aktor		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri. Aktor hanya memberikan informasi ke sistem, aktor hanya menerima informasi dari sistem, aktor memberikan dan menerima informasike sistem dan dari sistem.
Asosiasi		Komunikasi antara aktor dan *use case* yangberpartisipasi pada *use case* atau *use case* memiliki interaksi dengan aktor. Asosiasiyang menggambarkan elemen yang memiliki atribut berupa elemen lain
Esktensi		Relasi use case tambahan ke sebuah use case dimana use case yang ditambahkan dapat berdiri sendiri walaupun tanpa use case tambahan itu, mirip dengan prinsip inheritance pada pemrograman berorientasi objek. Biasanya use case tambahan memiliki

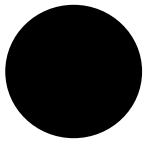
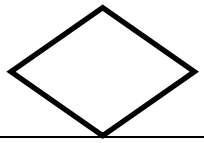
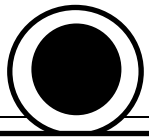
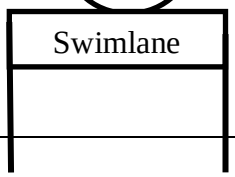
		nama depan yang sama dengan use case yang ditambahkan. Misalnya arah panah mengarah pada use case yang ditambahkan, biasanya use case yang menjadi extend-nya merupakan jenis yang sama dengan use case yang menjadi induknya.
Generalisasi		Hubungan generalisasi dan spesialisasi (umum-khusus) anatara dua buah use case dimana fungsi yang satu adalah fungsi yang lebih umum dari yang lainnya, misalnya : arah panah mengarah pada use case yang menjadi generalisasinya (umum). Generalisasi merupakan hubungan hirarkis antara elemen. Elemen dapat mewarisi semua atribut dan metode elemen asalnyadan menambah fungsionalitas baru.
Include	<<include>> 	memerlukan use case ini untuk menjalankan fungsinya atau sebagai syarat dijalankan use case ini Ada dua sudut pandang yang cukup besar mengenai include di use case : • Includeberarti use case yang ditambahkan akan selalu di panggil saat use case tambahan dijalankan • Includeberarti Usecase yang tambahan akan selalu melakukan pengecekan apakah usecase yang ditambahkan telah dijalankan sebelum usecase tambahan dijalankan

2.3.2. *Activity Diagram*

Rosa dan M. Shalahudin (2014:161), diagram aktivitas atau *activitydiagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuahsebuah sistem atau

proses bisnis atau menu yang ada pada perangkat lunak. Yangperlu di perhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Tabel 2.2. Simbol pada *Activity Diagram*

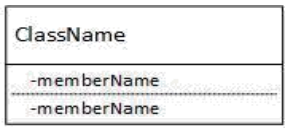
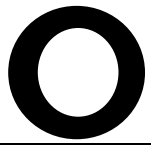
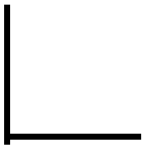
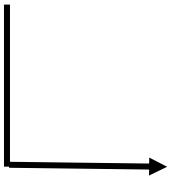
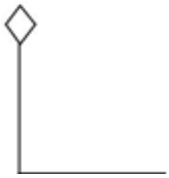
Status awal		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan		Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Status Akhir		Status akhir yang dilakukan oleh sistem,sebuah diagram aktivitas memiliki sebuah status akhir.
Swimline		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

2.3.3. *Class Diagram*

Rosa dan M. Shalahudin (2014:141), diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan *method* atau operasi. Berikut penjelasan atribut dan *method* :

1. Atribut merupakan variable-variabel yang dimiliki oleh suatu kelas.
2. Operasi atau *method* adalah fungsi-fungsi yang dimiliki oleh suatu kelas.

Tabel 2.3. Simbol pada *Class Diagram*

Kelas		Kelas pada struktur sistem
Antarmuka/*interface*		Sama dengan konsep interface dalampemrograman berorientasi objek
Asosiasi/*association*		Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan multipli city
Asosiasi berarah/ *directed*		Relasi antar kelas dengan makna kelas
Generalisasi		Association yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity
Kebergantungan/*dependensi*		Relasi antar kelas dengan makna generalisasi-spesialisasi(umum-khusus)
Agrasi		Relasi antar kelas dengan makna kebergantungan antar kelas

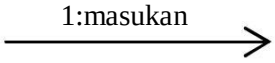
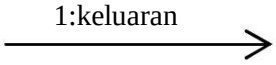
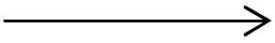
2.3.4. *Sequence Diagram*

Rosa dan M. Shalahudin (2014:165), diagram sekuen menggambarkan kelakuan objek pada *use case* dengan mendeskripsikan waktu hidup objek dengan *massage* yang dikirimkan dan diterima antar objek. Oleh karena itu untukmenggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Membuat

diagram sekuen juga dibutuhkan untuk melihat skenario yang ada pada *use case*. Banyaknya diagram sekuen yang harus digambar adalah minimal sebanyak pendefinisian *use case* yang memiliki proses sendiri atau yang penting semua *use case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup dalam diagram sekuen sehingga semakin banyak *use case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakinbanayak.

Tabel 2.4. Simbol pada *Sequence Diagram*

Aktor		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri,jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya dinyatakan dalam menggunakan kata benda diawal frase nama aktor.
Garis hidup/*lifeline*		Menyatakan kehidupan suatu objek
Objek	Nama Objek	Menyatakan objek yang berinteraksi pesan
Waktu aktif		Menyatakan objek dalam keadaan aktif dan berinteraksi, semuanya yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya
Pesan tipe *create*	<<Create>>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
Pesan tipe *call*	1:nama_metode()	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya

		sendiri,
Pesan tipe *send*	1:masukan 	Menyatakan bahwa suatu objek mengirimkan data/masukkan/informasi ke objek lainnya,arah panah mengarah pada objek yang dikirim
Pesan tipe *return*	1:keluaran 	Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian
Pesan tipe *destroy*	<<destroy>> 	Menyatakan suatu objek mengakhiri hidup objek yang lain,arah panah mengarah pada objek yang diakhiri, sebaliknya jika ada create maka ada destroy