

PEMANFAATAN PHONEGAP UNTUK MEMBANGUN APLIKASI DI ANDROID DAN IOS DALAM MENENTUKAN ARAH KIBLAT DAN WAKTU SHOLAT DENGAN MENGGUNAKAN GPS

Asep Wahyudi Zein¹, DR., Ir. Nazori Az, M.T², Mardi Hardjianto, M.Kom³

^{1,2,3}Program Studi Magister Komputer Program Pasca Sarjana Universitas Budi Luhur Jakarta
Jln. Ciledug Raya Petungkang Utara, Jakarta Selatan 12260

¹wawah.zein@gmail.com, ²nazori_agani@yahoo.com, ³mardi.hardjianto@budiluhur.ac.id

ABSTRAK

Melaksanakan ibadah sholat lima waktu sesuai dengan waktu yang telah ditetapkan dan menghadap kiblat adalah keutamaan dalam ibadah wajib ini. Pemanfaatan smartphone dalam kehidupan sehari-hari sudah sangat lumrah bagi sebagian orang. Perkembangan smartphone dewasa ini sangat pesat diiringi dengan semakin banyaknya platform. Hal ini dapat menyulitkan para programmer smartphone karena perbedaan lingkungan dan bahasa pemrograman untuk setiap platform. Dengan adanya masalah tersebut lalu bermunculan tools cross platform. Dengan tools ini, pengembang aplikasi tidak perlu mempelajari satu persatu platform yang ada. Mereka hanya butuh menguasai sebuah tools, yang nantinya dapat membantu aplikasi mereka berjalan di banyak platform (sekali coding jalan di banyak platform). Phonegap atau Cordova adalah salah satu tools cross platform yang banyak digunakan. Dalam penelitian ini akan dibuat aplikasi multiplatform yaitu pembuatan aplikasi jadwal sholat, arah kiblat dan lokasi masjid terdekat, yang memanfaatkan GPS, Kompas dan Google API. Penelitian ini merupakan jenis Penelitian Terapan (Applied Research). Metode pengembangan sistem menggunakan model Waterfall. Metode pengumpulan data dilakukan dengan observasi, studi pustaka dengan menggunakan metode convenience sampling. Metode yang digunakan dalam menganalisis dan merancang sistem adalah metode Analisis dan Perancangan Berorientasi Obyek (Object Oriented Analysis and Design) menggunakan Unified Modelling Language (UML). Teknik pengujian sistem dengan pendekatan white-box dan black-box testing. Aplikasi ini dapat berjalan di semua platform yang di dukung oleh phonegap, tetapi karena keterbatasan device aplikasi ini hanya akan dicoba di platform android dan iOS.

Kata Kunci : Cross platform, PhoneGap, API, GPS, multiplatform, jadwal sholat, arah qiblat, Waterfall.

I. PENDAHULUAN

Dengan segala aktifitas yang kian padat menjadikan sebagian orang memiliki tingkat mobilitas yang tinggi. Terkadang hal yang tidak menjadi prioritas namun suatu kewajiban terlalaikan. Salah satunya kewajiban ibadah sholat[1] fardhu lima waktu bagi umat Muslim yang kadang terlambat, terabaikan bahkan terlupakan. Salah satu faktor penyebabnya adalah terbatasnya informasi. Misalkan ketika seseorang berada di suatu tempat atau daerah yang suara adzan tidak terjangkau dari masjid terdekat atau suatu lokasi yang baru, maka ketika waktu shalat tiba ada kemungkinan orang tersebut tidak tahu. Atau ketika hendak melakukan sholat bukan di masjid atau musholah ada kemungkinan kesulitan dalam menentukan arah kiblat.

Salah satu pendukung mobilitas dalam kehidupan sehari-hari manusia adalah kehadiran *smartphone*[2]. Perkembangan *smartphone* semakin maju dan pesat, termasuk dalam pengembangan aplikasinya. Aplikasi-aplikasi *smartphone* yang berplatform android[3], blackberry, iOS[4], Symbian dan lain-lain, sekarang sudah banyak dijumpai baik yang

gratis maupun yang berbayar. Jika dilihat dari sisi prosedural developer pengembangan aplikasi, aplikasi-aplikasi tersebut dikembangkan sesuai dengan ketentuan prosedur dan bahasa pemrograman yang ditentukan platformnya. Untuk membuat aplikasi native berplatform android dan *blackberry* harus menggunakan bahasa pemrograman java, sedangkan untuk membuat aplikasi native iOS menggunakan bahasa pemrograman Objective-C dan sebagainya.

Namun banyaknya *platform* menjadi bumerang bagi *developer* aplikasi. Perbedaan bahasa pemrograman dan lingkungan development membuat tantangan developer aplikasi semakin bertambah. Belum lagi dengan perkembangan API dan tren kebutuhan pengguna yang berubah dengan sangat cepat. Tentu dibutuhkan lingkungan pengembangan aplikasi yang lebih bersahabat dengan percepatan perubahan dan kecepatan pengembangan aplikasi[5].

Lalu muncul aplikasi-aplikasi dengan menggunakan teknologi web (*webmobile*) baru seperti HTML5[6] yang dikembangkan untuk aplikasi *multiplatform*[7]. Karena aplikasi html5 dijalankan dalam komponen *browser* maka aplikasi ini tidak

bisa mengakses API perangkat *device*. Phonegap mungkin bisa menjadi jawaban tantangan tersebut. Phonegap (Cordova) adalah teknologi yang dikembangkan oleh Nitobi, sebuah komunitas pengembang yang terbiasa dengan project IBM, RIM dan Microsoft[8]. Dengan adanya framework yang bernama phonegap, seorang developer dapat membuat aplikasi multiplatform[9], atau aplikasi yang dapat dioperasikan di platform android, blackberry, iOS dan lain-lain. Developer android dapat membuat aplikasi iOS tanpa harus menguasai bahasa Objective C, begitu juga sebaliknya. Pengembangan framework phonegap ini menggunakan HTML5, CSS dan Java Script[10]. Phonegap mendukung akses pada native API di platform yang didukungnya. Termasuk untuk kemampuan akses input seperti touch screen, kamera, GPS[11], kompas dan lain sebagainya. Kemampuan phonegap dalam menghasilkan aplikasi hybrid[12] telah banyak diadopsi oleh para pengembang software. Dalam kesempatan ini pula, akan dibangun aplikasi di android dan iOS dalam menentukan arah qiblat dan waktu sholat yang memanfaatkan GPS[13]. Dan juga memanfaatkan Google API untuk mendapatkan lokasi kota, negara dan posisi masjid terdekat dengan lokasi pengguna sesuai dengan koordinat GPS. sehingga hanya dengan sebuah code, aplikasi ini dapat berjalan di beberapa platform mobile.

1.1 Identifikasi Masalah

Berdasarkan latar belakang permasalahan dalam penelitian ini, maka permasalahan dapat diidentifikasi sebagai berikut:

1. Jika berada di luar jangkauan suara adzan masjid maka akan kesulitan dalam menentukan waktu sholat tiba.
2. Jika melakukan sholat bukan di masjid atau mushola maka akan kesulitan dalam menentukan arah kiblat[14].

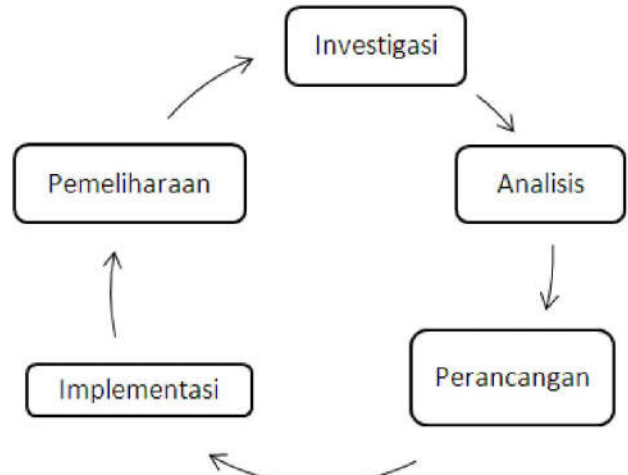
1.2 Rumusan Masalah

Berdasarkan latar belakang, identifikasi dan pembatasan masalah, maka dapat dirumuskan suatu permasalahan, yaitu bagaimana membangun suatu aplikasi yang dapat berjalan di android dan iOS yang dapat digunakan untuk menentukan arah kiblat, waktu sholat dan masjid sekitar pengguna.

II. METODE PENELITIAN

2.1 Systems Development Life Cycle (SDLC)

Menggunakan pendekatan sistem untuk mengembangkan solusi sistem informasi dapat dipandang sebagai proses multistep yang disebut siklus hidup pengembangan sistem informasi, yang dikenal juga sebagai *Systems Development Life Cycle* (SDLC). SDLC merupakan kerangka konseptual yang digunakan dalam manajemen proyek yang mendeskripsikan tahapan-tahapan yang dilakukan dalam proyek pengembangan sistem informasi[15]. Tahapan-tahapan dalam siklus pengembangan sistem menurut O'Brien meliputi: (1)investigasi, (2)analisis, (3)perancangan, (4)implementasi, dan (5)pemeliharaan[16].



Gambar 1. System Development Live Cycle

2.2 Model Waterfall

Dalam rekayasa perangkat lunak, konsep SDLC mendasari berbagai jenis metodologi pengembangan perangkat lunak. Metodologi-metodologi ini membentuk suatu kerangka kerja untuk perencanaan dan pengendalian pembuatan sistem informasi, yaitu proses pengembangan perangkat lunak. Beberapa proses pengembangan sistem dengan SDLC adalah: *waterfall*, *prototyping*, *incremental*, *spiral*, dan RAD[17].

Metodologi pengembangan sistem model *waterfall* menurut Pressman adalah proses pengembangan sistem yang menyiratkan pendekatan yang sistematis dan berurutan (sekuensial) pada pengembangan perangkat lunak, dimana proses pengembangan tersebut mengalir secara teratur ke bawah sehingga terlihat seperti air terjun[18]. Proses pengembangan dimulai dengan spesifikasi kebutuhan pengguna dan berlanjut melalui tahapan-tahapan perencanaan (*planning*), pemodelan (*modeling*), konstruksi (*construction*), serta penyerahan sistem perangkat lunak ke pelanggan/pengguna (*deployment*), yang diakhiri dengan dukungan berkelanjutan pada perangkat lunak yang dihasilkan



Gambar 2. Model Waterfall

Tahapan metodologi model waterfall tersebut dijelaskan sebagai berikut:

1. **Komunikasi.** Pada tahapan ini dilakukan komunikasi dan kolaborasi dengan pelanggan dan para pemangku kepentingan (*stakeholder*). Maksudnya adalah untuk memahami tujuan-tujuan stakeholder atas proyek perangkat lunak yang sedang dikembangkan dan mengumpulkan kebutuhan-kebutuhan yang akan membantu mengartikan fitur-fitur perangkat lunak beserta fungsinya.

2. **Perencanaan.** Tahapan perencanaan dilakukan dengan membuat rencana proyek perangkat lunak, untuk mengartikan ruang lingkup proyek rekayasa perangkat lunak dengan menggambarkan tugas-tugas teknis yang harus dilakukan, resiko yang mungkin muncul, sumber daya yang akan dibutuhkan, produk-produk kerja yang harus dihasilkan, dan jadwal kerja. Dapat juga dilakukan pelacakan terhadap perencanaan dan membuat penyesuaian jika hal ini memang diperlukan.

3. **Pemodelan.** Model digunakan untuk merepresentasikan informasi yang akan ditransformasi oleh perangkat lunak, fitur-fitur yang dikehendaki oleh pengguna, serta merepresentasikan perilaku sistem saat transformasi informasi tersebut benar-benar terjadi. Pada tahapan pemodelan, dibuat dua jenis model:

1) Model analisis. Untuk memperlihatkan spesifikasi kebutuhan pengguna dengan menggambarkan perangkat lunak dalam tiga ranah (*domain*) yang berbeda

2) Model perancangan. Menggambarkan karakteristik-karakteristik perangkat lunak

4. **Konstruksi.** Tahapan konstruksi mencakup penulisan kode program dan pengujian yang dibutuhkan untuk menemukan kesalahan-kesalahan dalam kode program yang sudah dibuat sebelumnya. Penulisan kode program dapat berupa: (1) pembuatan langsung kode program dalam bahasa pemrograman tertentu, (2) penulisan kode program secara otomatis menggunakan representasi mirip rancangan-rancangan yang akan dikembangkan, atau (3) pembuatan kode program menggunakan bahasa pemrograman generasi ke-4 yang langsung dapat dieksekusi. Pengujian yang dilakukan mencakup:

1) pengujian unit (*unit testing*), pengujian pada peringkat komponen.

2) pengujian integrasi (*integration testing*), yang dilakukan setelah sistem/perangkat lunak selesai dikonstruksi.

3) pengujian validasi (*validation testing*), yang melakukan penilaian apakah spesifikasi kebutuhan telah diakomodasi dalam sistem/perangkat lunak yang lengkap.

4) pengujian penerimaan (*acceptance testing*), yang dilakukan oleh pelanggan dengan tujuan untuk melakukan pemeriksaan atas semua fungsi dan fitur yang diinginkannya.

5. **Penyerahan Sistem.** Tahapan ini merupakan aktivitas penyerahan sistem/perangkat lunak kepada pelanggan (*deployment*), yang memiliki tiga aksi penting: pengiriman, dukungan, dan umpan balik. Perangkat lunak akan disajikan kepada pelanggan yang kemudian akan mengevaluasi produk

yang disajikan dan akan memberikan umpan balik berdasarkan evaluasi tersebut.

2.3 Hipotesis

Berdasarkan kerangka pemikiran yang menjadi panduan penelitian, maka diajukan Hipotesis (jawaban sementara) dengan pernyataan (statement) sebagai berikut :

1. Bahwa untuk pembuatan aplikasi di android dan Ios bisa dilakukan dengan menggunakan framework phonegap.
2. Bahwa untuk penentuan kiblat dan waktu sholat di butuhkan informasi GPS, tanggal, jam dan altitude matahari.
3. Bahwa untuk mendapatkan lokasi masjid terdekat dengan posisi kita bisa di manfaatkan google map dan GPS.

2.4 Metode Pemilihan Sampel

Metode pengambilan sample dalam penelitian ini adalah convenience sampling. Pengambilan sampel dengan convenience sampling merupakan teknik pengambilan sampel dengan sampel dipilih sesuai keinginan peneliti dengan alasan ketersediaan anggota atau yang mudah diperoleh[19] Adapun kriteria pemilihan responden dalam sampel penelitian ini adalah:

1. Seorang muslim.
2. Pengguna android dan iOS.

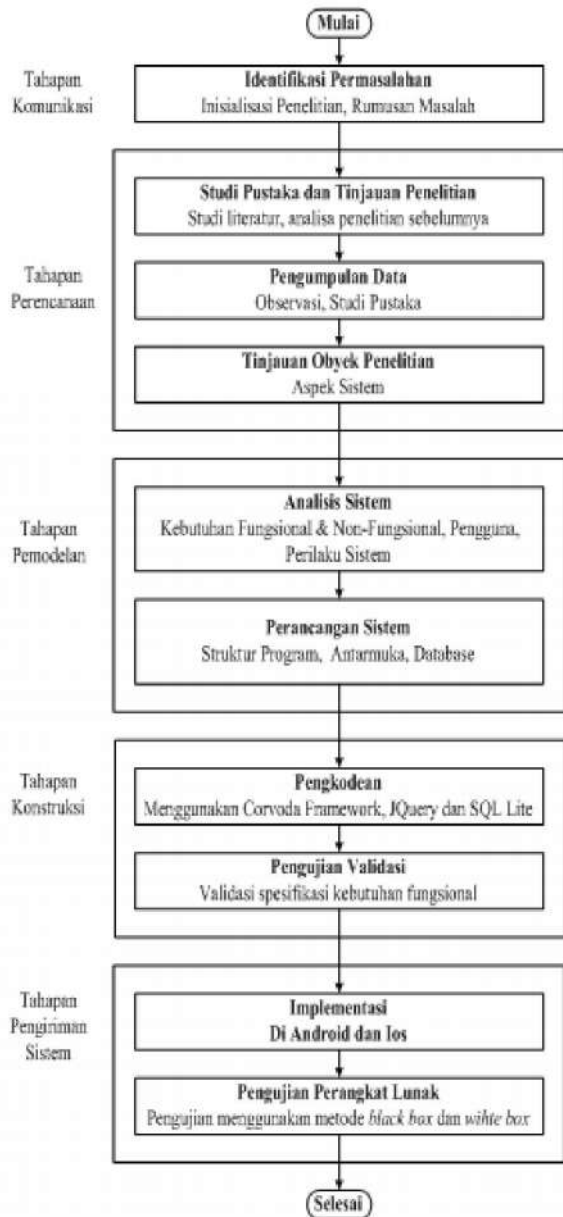
2.5 Metode Pengumpulan Data

Metode pengumpulan data yang digunakan dalam penelitian ini adalah:

1. Metode Observasi. Observasi atau pengamatan langsung terhadap lingkungan tempat penulis bekerja, disini diamati tentang kondisi lapangan apakah aplikasi yang dibangun nantinya bisa dipakai.
2. Studi Pustaka. Metode pengumpulan data yang diperoleh dengan mempelajari, meneliti, dan membaca buku, jurnal, skripsi, tesis yang berhubungan dengan aplikasi phonegap, jquery, perhitungan arah kiblat, jadwal sholat untuk membantu dalam pembuatan Aplikasi Pewaktuan Sholat[20]. Selain itu juga mempelajari web-web referensi seputar hal yang sama untuk membantu dalam penyajian informasi yang akan ditampilkan.

2.6 Langkah-langkah Penelitian

Dalam pengembangan sistem informasi, keseluruhan proses yang dilalui harus melalui beberapa tahapan. Dalam penelitian ini digunakan metode pengembangan sistem informasi model *Waterfall*. Tahapan yang dilakukan meliputi: komunikasi, perencanaan, pemodelan, konstruksi, dan pengiriman sistem. Langkah-langkah pada tahapan pelaksanaan penelitian dapat dilihat dalam bentuk diagram alir pada Gambar 3. berikut ini:



Gambar 3 Langkah-langkah Penelitian

1. Identifikasi Permasalahan. Tahapan ini merupakan inisialisasi dari penelitian, yaitu mencari permasalahan dalam membangun aplikasi *multiplatform* Berbasis mobile Untuk menentukan arah kiblat,waktu sholat dan masjid terdekat pada *framework cordova* dengan memanfaatkan *Global Position System* (GPS).

2. Studi Pustaka dan Tinjauan Penelitian. Tahapan ini dilakukan dengan mempelajari literatur yang berkaitan aplikasi *multiplatform* berbasis mobile,*framework cordova* dengan perhitungan waktu sholat, perhitungan arah kiblat, pemanfaatan *google mapplace* untuk mencari masjid,

memanfaatkan *Global Position System* (GPS). Dalam tahapan ini juga dilakukan studi dan analisa dari beberapa penelitian sebelumnya, berupa jurnal nasional maupun tesis mengenai aplikasi menentukan arah kiblat, waktu sholat dan masjid terdekat.

3. Pengumpulan Data. Proses ini dilakukan dengan cara melakukan observasi dan studi pustakautukmempelajari kebutuhan pengguna dan kebutuhan sistem.

4. Tinjauan Obyek Penelitian. Pada tahap ini akan dilakukan identifikasi obyek penelitian aspek teknis. Kebutuhan *tools* yang akan di gunakan untuk membangun aplikasi tersebut.

5. Analisis Sistem.Berdasarkan hasil pengumpulan data, akan dilakukan analisa kebutuhan pengguna dan kebutuhan fungsional dan non-fungsional sistem, dan analisa perilaku sistem. Permodelan yang dilakukan dalam tahapan analisis ini adalah:

a. Pembuatan use case diagram untuk memodelkan kebutuhan fungsional dan pengguna.

b. Pembuatan activity diagram dan sequence diagram. Activity diagram untuk memodelkan *prosesuse case* yang berjalan di dalam sistem. *Sequence diagram* untuk memodelkan pengiriman pesan (*message*) antar object dan kronologinya.

6. Perancangan Sistem. Berdasarkan hasil analisis, kemudian dilanjutkan dengan melakukan perancangan sistem untuk menggambarkan bagaimana sistem dapat memenuhi kebutuhan yang sudah ditetapkan, dengan melakukan prosedur-prosedur berikut:

a. Perancangan struktur program dengan membuat class diagram.

b. Perancangan physical architecture dengan membuat deployment diagram.

c. Perancangan antarmuka (input, output, dan navigasi)

d. Perancangan basis data dengan membuat ER-Diagram.

e. Perancangan infrastructure architecture.

7. Pengkodean. Pada tahap ini dilakukan pengkodean (*coding*) berdasarkan hasil analisis dan perancangan sistem kedalam bahasa pemograman yang digunakan berupa aplikasi berbasis *framework cordova*, *html5*, *javascript* yang akan di koneksikan kedatabase SQL Lite.

8. Pengujian validasi. Pengujian untuk memvalidasi fungsi perangkat lunak berdasarkan spesifikasi kebutuhan yang sudah dianalisis dan dirancang, meliputi pengujian fungsionalitas sistem dan non fungsional.

9. Implementasi sistem pada android dan iOs Berdasarkan tinjauan obyek penelitian, dalam tahap implementasi untuk penelitian ini dilakukan pada platform android dan iOs.

III. HASIL DAN PEMBAHASAN

3.1 Analisis Sistem

Pada proses analisis sistem harus mendeskripsikan apa yang harus dilakukan oleh sistem untuk memenuhi kebutuhan informasi pengguna. Analisis sistem akan menjawab pertanyaan apa yang akan dikerjakan oleh sistem, siapa yang akan menggunakan sistem, dan dimana serta kapan sistem

tersebut akan digunakan. Kegiatan analisis sistem yang berjalan dilakukan dengan pendekatan analisis berorientasi objek untuk sistem yang dirancang, dimaksudkan untuk menitik beratkan kepada fungsionalitas sistem yang berjalan. Selanjutnya dari hasil analisis akan divisualisasi dan didokumentasikan dengan *Unified Modeling Language (UML)* melalui *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram* dengan pertimbangan diagram tersebut dianggap mewakili secara keseluruhan sistem yang berjalan yang dapat dimengerti oleh pengguna[21].

3.1.1 Gambaran Umum System Yang Akan Dikembangkan

Dzikro adalah aplikasi yang dikembangkan perangkat mobile. Dapat berjalan diplatform android dan iOS. Aplikasi ini dapat digunakan untuk melihat jadwal sholat, arah kiblat dan masjid sekitar pengguna.

Diharapkan dengan adanya aplikasi ini pengguna bisa dapat melihat jadwal sholat, arah kiblat dan masjid sekitar secara mudah dan realtime, sesuai dengan posisi pengguna saat ini.

3.1.2 Analisis Kebutuhan Fungsional

Tahap analisis kebutuhan fungsional sistem akan membahas mengenai fungsi-fungsi yang diperlukan dalam pembangunan sistem. Berdasarkan hasil analisis proses bisnis, identifikasi kebutuhan data dan informasi, maka dianalisis juga beberapa fungsi yang harus tersedia di dalam sistem. Hal ini dilakukan untuk memenuhi kebutuhan data dan informasi yang diperlukan oleh pengguna.

Berikut fitur-fitur yang ada dalam aplikasi dzikro antara lain sebagai berikut:

- 1 Mampu menampilkan jadwal waktu shalat yang sesuai dengan lokasi dimana pengguna berada.
- 2 Mampu menampilkan arah kiblat secara otomatis dan real position berdasarkan posisi pengguna dengan bantuan kompas.
- 3 Terdapat pilihan pengaturan metode perhitungan waktu shalat sesuai yang diinginkan pengguna.
- 4 Mampu menampilkan masjid-masjid di sekitar posisi pengguna dengan bantuan google map.
- 5 Terdapat pilihan pengaturan penentuan waktu shalat Ashar berdasarkan mazhab pengguna.

3.1.3 Analisis Kebutuhan Nonfungsional

Setelah mendefinisikan kebutuhan fungsional yang harus dipenuhi oleh sistem maka langkah selanjutnya adalah mendefinisikan kebutuhan nonfungsional dari sistem yang akan dipenuhi. Kebutuhan ini adalah tipe kebutuhan yang berisi properti perilaku yang dimiliki oleh sistem, berikut ini adalah daftar kebutuhan nonfungsional sistem selengkapnya:

1. Kebutuhan Operasional
Sistem yang dibangun bisa digunakan pada platform sistem android dan iOS.
2. Kebutuhan Performansi
Dzikro merupakan aplikasi yang berjalan dismartphone yang berplatform android dan iOS, yang mempunyai

memori terbatas (256 Mb- 1Gb) untuk itu di rancang aplikasi dengan penggunaan memori seefektif mungkin, memanfaatkan sumber daya seefisien mungkin, dan merancang aplikasi antar muka yang sederhana namun tetap menarik dan ramah bagi pengguna.

3. Kebutuhan Kemudahan Penggunaan

Sistem seharusnya mudah digunakan dan mudah dipelajari. Sistem harus menggunakan bahasa yang mudah dimengerti serta sistem seharusnya memiliki tampilan menarik.

3.1.4 Analisis Pengguna

Pengguna aplikasi dzikro adalah pengguna *smartphone* android dan iOS yang menginstall aplikasi dzikro dalam hal ini disebut sebagai *user* (pengguna).

Spesifikasi kebutuhan pengguna bisa dilihat di table 3-1:

Tabel 1. Spesifikasi Kebutuhan Pengguna

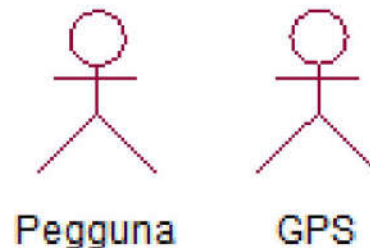
Proses	Kebutuhan Pengguna
Dzikro	1. Melihat Jadwal Sholat
	2. Melihat Lokasi Masjid Sekitar
	3. Melihat Arah Kiblat
	4. Menentukan Metode Perhitungan
	5. Menentukan Mazhab

3.2 Use Case Diagram

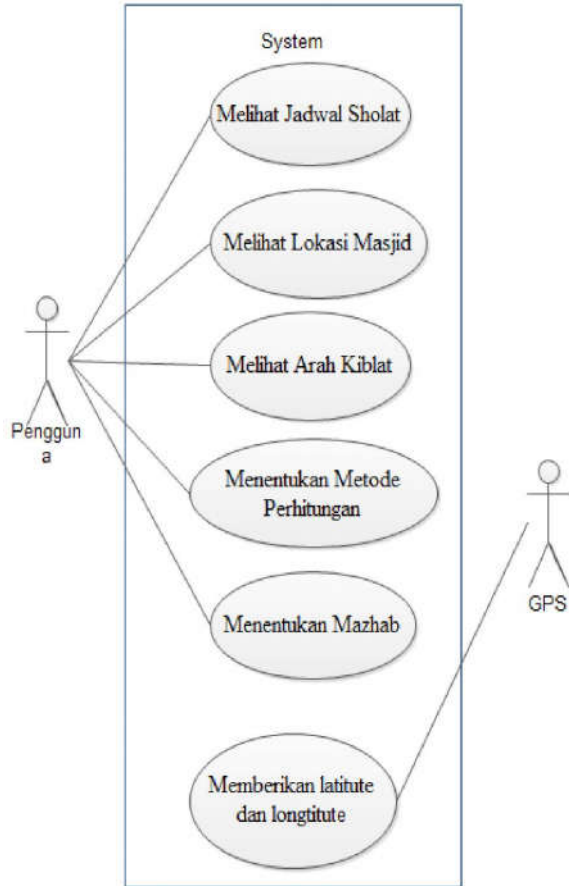
Berdasarkan spesifikasi kebutuhan fungsional dan actor yang terlibat dalam sistem, maka dapat dimodelkan dengan *use case diagram*. *Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah system. *use case diagram* menggambarkan interaksi antara actor dengan system.

1. Actor

Actor adalah segala sesuatu yang berinteraksi langsung dengan sistem aplikasi komputer, seperti orang, benda atau lainnya. Tugas actor adalah memberikan informasi kepada sistem dan dapat memerintahkan sistem agar melakukan suatu tugas. Berdasarkan identifikasi kebutuhan pengguna, maka actor yang terlibat dalam aplikasi dzikro ini bisa dilihat di gambar 4:



Gambar 4. Actor aplikasi dzikro



Gambar 5. Use Case Diagram

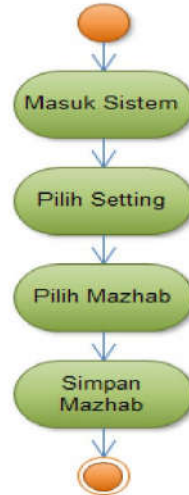
3.3 Analisis Perilaku Sistem

3.3.1 Activity Diagram

Activity Diagram merupakan model analisis yang digunakan atau menggambarkan sebuah proses aktivitas. Dalam penelitian ini, *activity diagram* digunakan untuk memodelkan suatu proses atau operasi, diagram ini dipakai untuk menggambarkan logika dari sebuah proses atau operasi.

Activity Diagram menjelaskan tentang alir aktifitas di dalam sistem yang sedang dirancang, bagaimana alir berawal, bagaimana keputusan-keputusan dapat terjadi dan terakhir bagaimana sistem berakhir. Dalam penelitian ini peneliti menggambarkan *activity diagram* tersebut melalui interaksi antara actor dengan sistem yang akan dibangun. Dengan demikian, *activity diagram* diturunkan dari setiap use case yang sudah dibuat sebelumnya, untuk menggambarkan masing-masing prosedur dalam aplikasi dzikro.

A. Activity Diagram Menentukan Mazhab



Gambar 6 Activity Diagram Menentukan Mazhab

Deskripsi *Activity Diagram* Menentukan Mazhab:

1. Pengguna membuka sistem.
2. Pengguna memilih setting.
3. Pengguna memilih mazhab.
4. Sistem akan menyimpan mazhab yang di pilih user.
5. Pengguna keluar/logout dari sistem

B. Activity Diagram Menentukan Pengaturan Sholat

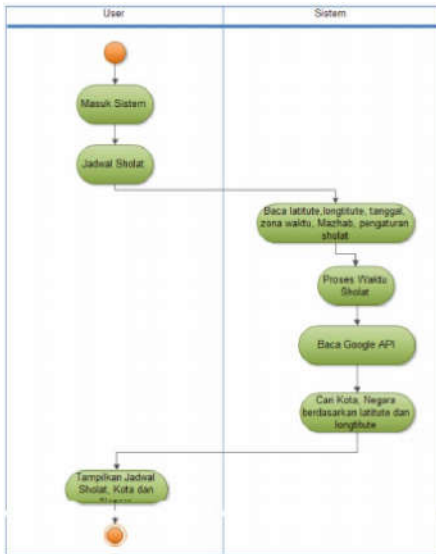


Gambar 7 Activity Diagram Menentukan Pengaturan Sholat

Deskripsi *Activity Diagram* Menentukan Pengaturan Sholat:

1. Pengguna membuka sistem.
2. Pengguna memilih *setting*.
3. Pengguna memilih pengaturan sholat.
4. Sistem akan menyimpan pengaturan yang di pilih *user*.

C. Activity Diagram Melihat Jadwal Sholat

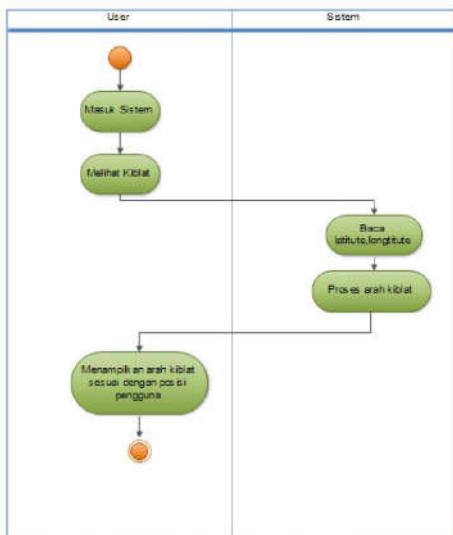


Gambar 8. Activity Diagram Melihat Jadwal Sholat

Deskripsi Activity Diagram Melihat Jadwal Sholat:

1. Pengguna membuka sistem.
2. Pengguna Melihat jadwal sholat.
3. Sistem akan membaca latitude, longitude, zona waktu, tanggal, pengaturan sholat dan mazhab.
4. Sistem akan memproses waktu sholat
5. Sistem akan membaca google API, guna mendapatkan nama kota dan negara.
6. Sistem akan menampilkan ke user informasi jadwal sholat dan kota dan negara.

D. Activity Diagram Melihat Arah Kiblat



Gambar 9 Activity Diagram Melihat Arah Kiblat

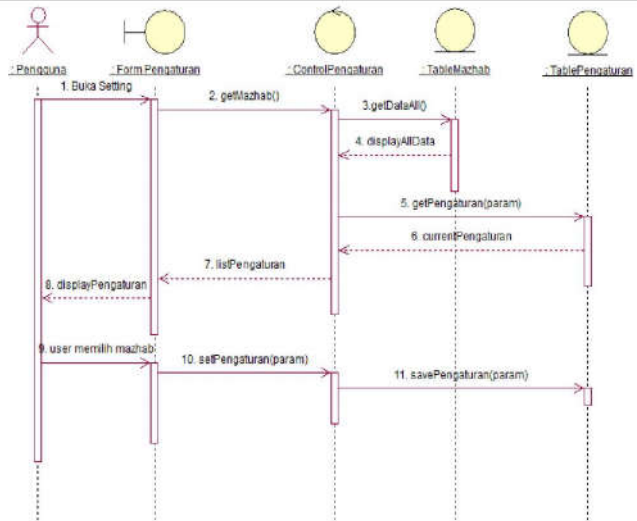
Deskripsi Activity Diagram Melihat Kiblat:

1. Pengguna membuka sistem.
2. Pengguna melihat kiblat.
3. Sistem akan membaca latitude, longitude.
4. Sistem akan memproses arah kiblat sesuai dengan posisi pengguna.
5. User akan melihat petunjuk arah kiblat berupa grafik.

3.3.2 Sequence Diagram

Pada Sequence Diagram menjelaskan aliran fungsionalitas dalam use case. Sebuah sequence diagram menunjukkan urutan interaksi object yang disusun dalam urutan waktu. Ini menggambarkan object dan class-class yang terlibat dalam skenario dan urutan pesan yang dipertukarkan antara object yang dibutuhkan untuk melaksanakan fungsi scenario. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antara object, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. Berikut beberapa contoh sequence diagram sistem yang akan dikembangkan

A. Sequence Diagram Pengaturan Mazhab

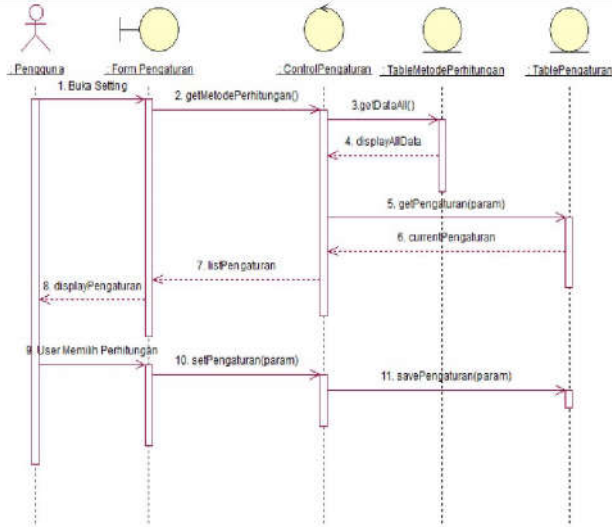


Gambar 10. Sequence Diagram Pengaturan mazhab

Deskripsi Sequence Diagram Pengaturan Mazhab:

1. Pengguna membuka aplikasi,
2. Program akan membaca metode perhitungan.
3. Program akan membaca data dari table mazhab.
4. Didapat semua data dari table mazhab.
5. Program membaca settingan mazhab saat ini.
6. Didapat mazhab saat ini.
7. Program akan mengolah mazhab.
8. Program akan menampilkan data-data tersebut.
9. User memilih pengaturan.
10. Program akan mengirim data mazhab yang di pilih user.
11. Sistem akan menyimpan mazhab ke dalam database

B. Sequence Diagram Metode Perhitungan Sholat

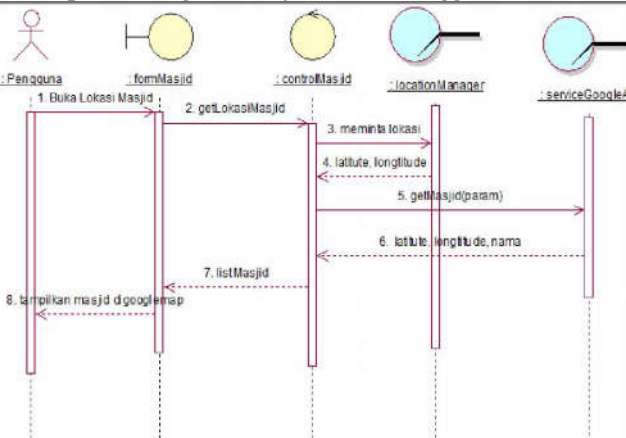


Gambar 11 sequence diagram Perhitungan Sholat

Deskripsi Sequence Diagram Pengaturan Metode Perhitungan Sholat:

1. Pengguna membuka aplikasi,
2. Program akan membaca metode perhitungan.
3. Program akan membaca data dari table metode perhitungan.
4. Didapat semua data dari table metode perhitungan.
5. Program membaca settingan metode perhitungan saat ini.
6. Didapat metode perhitungan saat ini.
7. Program akan mengolah metode perhitungan.
8. Program akan menampilkan data-data tersebut.
9. User memilih pengaturan.
10. Program akan mengirim data pengaturan yang di pilih user.
11. Sistem akan menyimpan pengaturan ke dalam database

C. Sequence Diagram Masjid Sekitar Pengguna

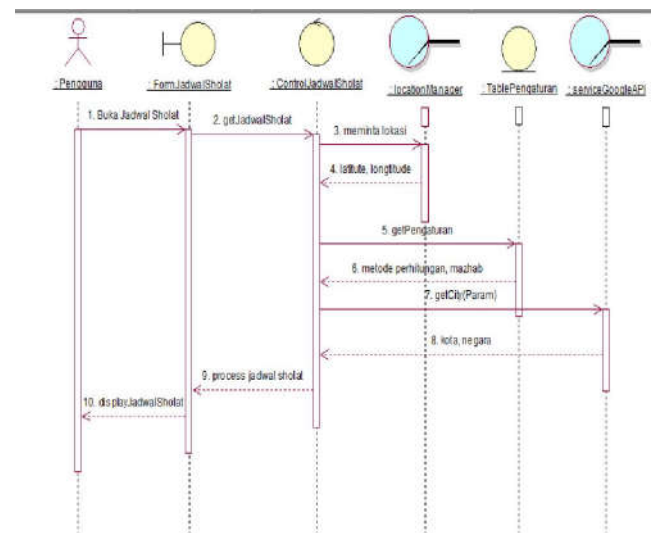


Gambar 13 Squence diagram masjid sekitar pengguna

Deskripsi Sequence Diagram Masjid Sekitar Pengguna:

1. Pengguna membuka aplikasi, kemudian memilih lokasi masjid
2. Program akan meminta lokasi masjid.
3. Program meminta lokasi ke location manager.
4. Location manager akan memberikan latitude dan longitude.
5. Program akan mengirimkan latitude dan longitude ke google API.
6. Google API akan memberikan latitude, longitude dan nama masjid.
7. Kemudian akan didapatkan list masjid.
8. Program akan menampilkan masjid-masjid dan posisi pengguna di google map.

D. Sequence Diagram Jadwal Sholat



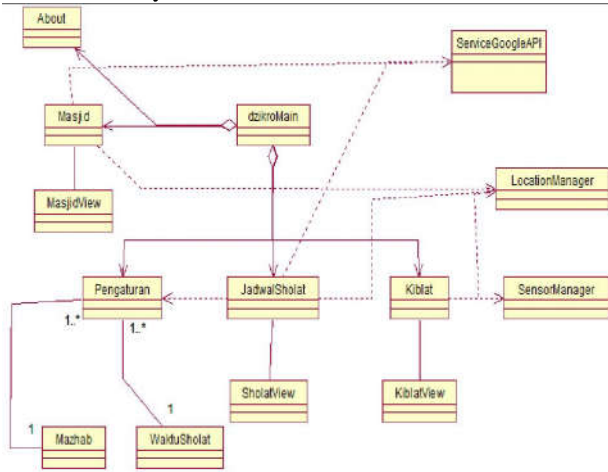
Gambar 14 Squence diagram Jadwal Sholat

Deskripsi Sequence Diagram Jadwal Sholat:

1. Pengguna membuka aplikasi, kemudian memilih jadwal sholat.
2. Program akan meminta jadwal sholat.
3. Kontrol program meminta lokasi ke location Manager.
4. Location manager akan mengirimkan latitude dan longitude.
5. Program akan membaca table pengaturan.
6. Akan didapat metode dan mazhab yang di pilih user saat ini.
7. Program akan meminta kota dan negara ke service google API dengan mengirimkan latitude dan longitude.
8. Googel API akan mengirim kan kota dan negara sesuai dengan parameter yang di kirim
9. Program akan mengolah jadwal sholat sesuai dengan parameter-parameter tadi.
10. Sistem akan menampilkan jadwal sholat untuk hari ini.

3.3.3 Class Diagram

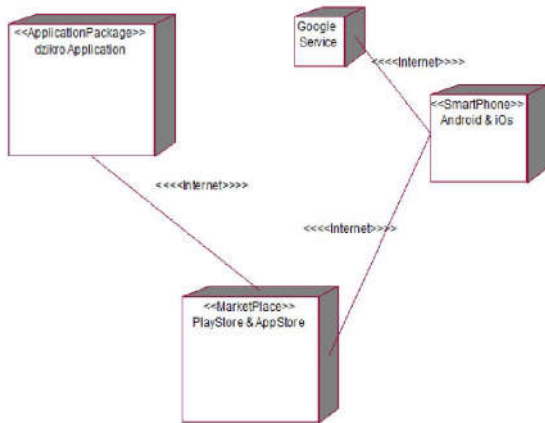
Class Diagram menampilkan beberapa class ada pada system ini dan memberikan gambaran tentang system dan relasi-relasi didalamnya.



Gambar 15. Class Diagram aplikasi Dzikro

3.3.4 Deployment Diagram

Deployment diagram merupakan suatu diagram yang dapat memberikan penjelasan tentang bagaimana berbagai elemen fisik menyusun dan menjalankan sistem di dalam suatu jaringan yang dibentuk. Arsitektur jaringan yang dibentuk merupakan kumpulan dari node-node yang berupa hardware dan software.

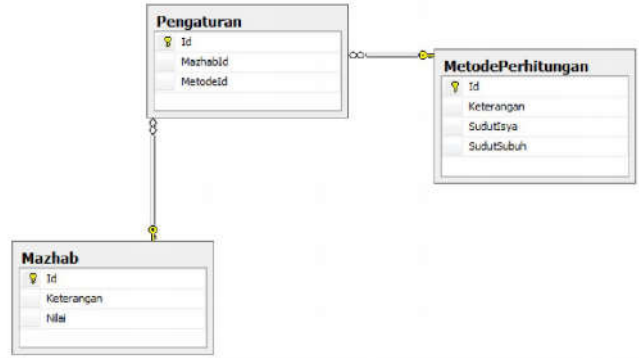


Gambar 16. Deployment Diagram

3.3.5 Perancangan Database

Database dalam aplikasi dzikro dibutuhkan untuk menyimpan konfigurasi yang di pilih oleh user. Adapun tahapan perancangan database terdiri dari dari ER-Diagram dan Stuktur table.

3.3.5.1 ERD



Gambar 17. ERD aplikasi Dzikro

3.3.5.2 Stuktur tabel

Tabel 2 menunjukan tabel yang dibangun untuk aplikasi dzikro beserta fungsi masing-masing:

Tabel 2: Daftar Tabel Database Sistem

No	Tabel	Fungsi
1	Mazhab	Menyimpan Master Mazhab
2	MetodePerhitungan	Menyimpan Master MetodePerhitungan.
3	Pengaturan	Menyimpan data MethodedanMazhab yang di pilihpengguna.

3.4 Konstruksi Sistem

Setelah tahapan analisa dan perancangan selanjutnya tabahap konstuksi sistem. Dalam konstruksi sistem ini dibutuhkan hardware dan software dalam pengembangan aplikasi.

3.4.1. Lingkungan Kontruksi

A. Hardware

Hardware atau perangkat keras yang digunakan untuk mengkonstruksi atau membangun aplikasi dzikro sebuah unit laptop thinkpad x201, dengan spesifikasi antara lain:

1. Processor Intel Core i5 I5-540M / 2.53 GHz
2. Chipset Type Mobile Intel QM57 Express
3. Memory Installed 4 GB DDR3 SDRAM - 1066 MHz PC3-8500
4. Hard Drive 320 GB - Serial ATA-300 - 7200 rpm
5. Display 12.1 in TFT active matrix Max Resolution 1280 x 800 (WXGA)
6. OS Provided Microsoft Windows 7 Professional 32-bit

B. Software

Dalam perancangan dan pengembangan aplikasi dzikro dibutuhkan beberapa *software*:

1. IDE (integrated development environment)

Untuk mempermudah pengembangan aplikasi digunakan IDE, IDE yang di pakai adalah Icenium Graphite.

 - a. Sistem Operasi

Sistem Operasi yang di pakai dalam pengembangan aplikasi adalah windows 7 Profesional 32 Bit.
 - b. Bahasa Pemograman

Dalam pengembangan dzikro ada beberapa bahasa pemograman dan framework yang di pakai antara lain:

 - Javascript
 - HTML5
 - Cordova 2.7
 - Kendo UI.
2. Database

Database yang di gunakan untuk aplikasi dzikro adalah sqllite. Sqllite merupakan database *embleded* yang ada di setiap *smarphone*.
3. Icenium Ion

Icenium Ion merupakan *tools* untuk menginstall dan testing aplikasi ke *device ios*, tanpa harus menggunakan *developer key*.

3.4.2. Kontruksi Antar Muka

Bagian ini akan menjelaskan implementasi atau kontruksi berdasarkan hasil rancangan tampilan aplikasi dzikro yang sudah dibahas dalam perancangan antarmukapengguna. Untuk menjelaskan hasil kontruksi tersebut akan diberikan prototype dari masing-masing tampilan, baik itu tampilan input,output, navigasi maupun halaman pada aplikasi yang dibangun.

1. Tampilan halaman Utama

Halaman utama aplikasi dzikro akan langsung menampilkan menu jadwal sholat. Hal ini tujuan untuk mempermudah pengguna untuk langsung mengetahui jadwal sholat untuk tanggal saat ini.

2. Tampilan Navigasi

Menu navigasi berupa toolbar, untuk iOS dan Android mempunyai perbedaan design. Untuk android *toolbar* ada di bagian atas dan untuk iOS ada di bawah. Perbedaan design ini tergantung dari platform tersebut. Untuk menghadle masalah perbedaan design digunakan kendo ui. Hal ini akan mempermudah pengembang *focus* pada aplikasi yang di kembangkan. Untuk tampilan *toolbar* aplikasi dzikro pada Ios bisa di lihat pada Gambar 18 dan Gambar 19 Untuk toolbar pada android.



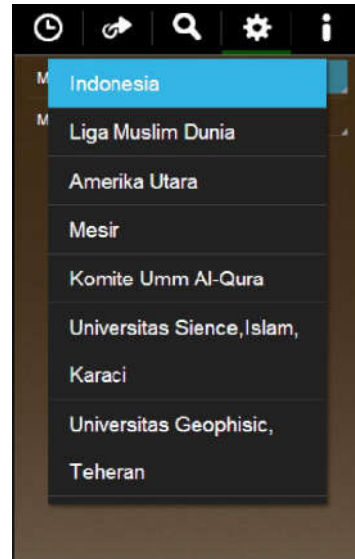
Gambar 18. *Toolbar* Aplikasi Dzikro untuk *Ios*



Gambar 19. *Toolbar* Aplikasi Dzikro untuk *Android*

3. Tampilan Layar Metode Perhitungan

Metode perhitungan merupakan pilihan metode-metode perhitungan sholat yang ada secara umum, pengguna bisa memilih metode perhitungan sholat sesuai dengan daerahnya atau berdasarkan keyakinannya. Untuk tampilan menu pada android bisa di lihat di gambar 20 dan untuk ios bisa dilihat di gambar 21 .



Gambar 20. Tampilan Setting Metode Perhitungan di *Android*



Gambar 21. Tampilan Setting Metode Perhitungan di *iOs*

4. Tampilan Layar Pengaturan Mazhab

Mazhab merupakan pengaturan yang digunakan untuk menentukan waktu sholat ashar. Di aplikasi dzikro terdapat 2 mazhab yang bisa di pilih. Tampilan untuk pemilihan mazhab

bisa dilihat pada gambar 22 untuk android dan gambar 23 untuk Ios



Gambar 22. Tampilan Setting Mazhab di Android



Gambar 23. Tampilan Setting Mazhab di Ios

5. Tampilan Jadwal Sholat

Tampilan jadwal sholat untuk platform android ditunjukkan pada gambar 24 dan gambar 25 untuk platform. Jadwal sholat ini menampilkan tanggal pada bagian atas, waktu sholat pada bagian tengah dan nama kota dan negara pada bagian bawah.



Gambar 24. Tampilan Jadwal Sholat Untuk Android



Gambar 25. Tampilan Jadwal Sholat Untuk ios.

6. Tampilan Arah Kiblat

Untuk tampilan arah kiblat bisa dilihat di gambar 26 pada android dan gambar 27 pada iOS. Tampilan berupa grafik kompas dengan jarum penunjukan untuk menunjukkan arah kiblat dan pada bagian bawah terdapat derajat arah kiblat.



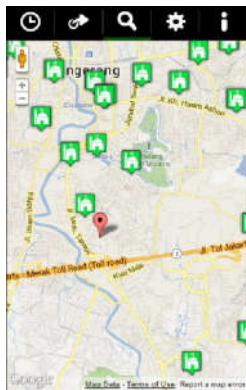
Gambar 26 Tampilan Arah Kiblat pada Android.



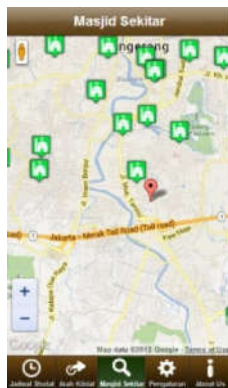
Gambar 27. Tampilan Arah Kiblat pada iOS

7. Tampilan Lokasi Masjid

Untuk masjid sekitar pengguna di tunjukan pada gambar 28 untuk android dan gambar 29. Disini terdapat 2 jenis *marker* pada *google map* yang berupa *icon* balon merah dan *icon* masjid hijau. *Icon* balon merah menunjukan posisi pengguna sekarang. Dan *icon* masjid merupakan masjid-masjid yang ada disekitar pengguna.



Gambar 28 Lokasi Masjid di Android



Gambar 29. Lokasi Masjid di ios

3.4.3. Pengujian Sistem

Tahap selanjutnya adalah proses pengujian perangkat lunak, proses pengujian ini dilakukan untuk memastikan perangkat lunak yang telah dibuat sesuai dengan kebutuhan.

A. Lingkungan Sistem

Lingkungan pengujian memberikan gambaran tentang spesifikasi *hardware*, *software*, dan jaringan yang digunakan oleh pengguna dalam proses pengujian sistem. Berikut ringkasan singkat spesifikasi perangkat keras dan perangkat lunak serta jaringan yang digunakan oleh pengguna untuk proses pengujian:

1) Hardware

Dalam perancangan, pengembangan dan implementasi Dzikro dibutuhkan beberapa perangkat keras, antara lain:

a. Android

Handphone yang digunakan dalam pengembangan aplikasi dzikro, adalah Ktouch W700, yang mempunyai spesifikasi sebagai berikut:

- General Network : Gsm 850/900/1800/1900 Mhz
- LayarTipe : Tft Capacitive 16 Juta Warna
- Ukuran : 3,8 Inchi, 480 X 800 Pixels
- Dimensi : Ukuran/Berat 18,7 X 60,1 X 13,7 Mm / 150g
- Memory : Internal 8 Gb
- Data : 3gHsdpa 14 Mbps / Hsupa 5,76 Mbps
- Usb/Port : Ya
- Os : Android 2.2 Froyo
- Cpu : 1ghz Nvidia Tegra 2, Ram 512 Mb
- Gps : Ya, A-Gps
- Digital Compass : Ya

b. Ios

Untuk platform ios digunakan ipad 3, 64 Gb, dengan spesifikasi sebagai berikut:

- GeneralNetwork : 2g Gsm 850 / 900 / 1800 / 1900 3g Hsdpa 850 / 1900 / 2100
- Layar Tipe : Led-Backlit Ips Tft, Capacitive Touchscreen, 16m Colors
- Ukuran : 768 X 1024 Pixels, 9.7 Inches
- Jack : 3.5 Mm Jack Audio
- Speakerphone : Ya
- Memory : Internal 64 Gb, 256 Mb Ram
- Data : 3g Hsdpa, Hsupa
- Edge : Ya
- Gprs : Ya
- Usb/Port : Ya, Usb V2.0
- Os : Ios 6.1
- Cpu : 1 Ghz Arm Cortex-A8 Processor, Powervr Sgx535gpu, Apple A4 Chipset
- Gps : Ya, With A-Gps Support
- Digital Compass : Ya

c. Kabel USB

Fungsi Kabel dari USB ini adalah untuk menghubungkan PC dengan Handphone.

d. Laptop

Sebuah unit laptop thinkpad x201, dengan spesifikasi antara lain:

- Processor Intel Core i5 I5-540M / 2.53 GHz
- Chipset Type Mobile Intel QM57 Express
- Memory Installed 4 GB DDR3 SDRAM - 1066 MHz PC3-8500
- Hard Drive 320 GB - Serial ATA-300 - 7200 rpm
- Display 12.1 in TFT active matrix Max Resolution 1280 x 800 (WXGA)
- OS Provided Microsoft Windows 7 Professional 32-bit

2) Icenium Graphite.

Berfungsi selain untuk IDE juga untuk mendeploy aplikasi.

3) Icenium Ion.

Icenium Ion merupakan tools untuk menginstall dan testing aplikasi ke device ios, tanpa harus menggunakan developer key.

4) Jaringan

Untuk menggunakan aplikasi dzikro dibutuhkan jaringan berupa internet. Internet digunakan untuk mendapatkan nama kota, negara dan masjid di sekitar posisi pengguna.

3.4.4. Metode Pengujian Sistem

1. White Box Testing

Tahap pengujian yang pertama adalah mengetahui cara kerja internal. Pengujian *white box* dilakukan dengan menguji atribut dan method yang ada pada kelas-kelas yang dibangun. Pengujian ini dilakukan pada proses pengembangan. Selain untuk pengujian cara kerja internal, hal ini juga untuk menguji hipotesa pertama yaitu bahwa untuk pembuatan aplikasi di android dan Ios bisa dilakukan dengan menggunakan *framework phonegap*. Hipotesa pertama terbukti bahwa dengan menggunakan *phonegap* atau *cordova*, aplikasi dapat berjalan dengan baik di Ios dan Android tanpa harus merubah source code sesuai platformnya.

2. Black Box Testing

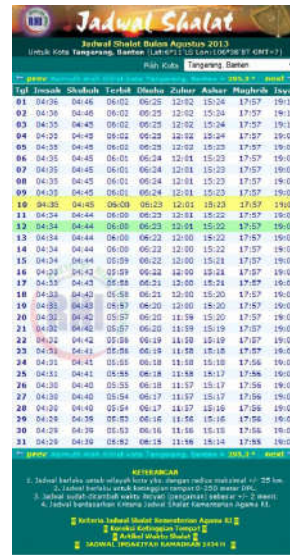
Teknik pengujian black-box testing berfokus pada ranah informasi dari perangkat lunak, menghasilkan test case dengan cara mempartisi ranah masukan dan keluaran dari sebuah program dengan cara mencakup pengujian yang menyeluruh.

a) Pengujian Black Box Jadwal Sholat.

Pengujian untuk jadwal sholat adalah dengan membandingkan hasil jadwal sholat dari dzikro dengan sumber terpercaya. Pada pengujian ini membandingkan dengan situs Rukyatul Hilal Indonesia (RHI) pada tanggal 10 Agustus 2013 dengan lokasi tangerang. Pada gambar 3-25 merupakan jadwal sholat yang ditampilkan dari aplikasi dzikro. Sedangkan gambar 3-26 merupakan gambar pada situs RHI.



Gambar 30. Jadwal Sholat di Aplikasi Dzikro



Gambar 31. Jadwal Sholat di Situs RHI

Berdasarkan hasil pengujian diatas waktu yang di tampilan dengan dzikro berbeda lebih kurang 2 Menit, hal ini dikarenakan aplikasi dzikro tidak ditambahkan waktu ihtiyati (waktu pengaman). Bisa di simpulkan hasil perhitungan waktu sholat antar aplikasi dzikro dan RHI sama.

b) Pengujian Black Box Arah Kiblat

Pengujian arah kiblat adalah dengan membandingkan derajat arah kiblat dzikro dengan situs Rukyatul Hilal Indonesia dikota tangerang. Gambar 32 arah kiblat pada aplikasi dzikro. Sedangkan gambar 33 arah kiblat yang di tunjukan pada situs RHI.



Gambar 32. Arah Kiblat di Aplikasi dzikro



Gambar 33. Arah Kiblat di situs RHI



Gambar 34. Arah Kiblat di Aplikasi dzikro posisi Horizontal

Jika di liat dari gambar derajat yang di hasil dzikro dan situ RHI sama yaitu 295.2o . dan jika posisi handphone di putar menjadi horizontal posisi penunjuk kiblat bisa mengikuti sesuai dengan arah kiblat. Berarti perhitungan arah kiblat dan RHI sama.

c) Rencana Implementasi Sistem

Dalam proses implementasi aplikasi dzikro diperlukan beberapa tahapan perencanaan untuk implementasi dan penerapan sistem. Tahapan tersebut adalah sebagai berikut:

Tabel 3. Tabel Rencana Implementasi

No	Kegiatan	Bulan 1				Bulan 2			
		1	2	3	4	1	2	3	4
1	Pembelian Developer Key								
2	Memasukan Aplikasi ke Market place								
3	Install Aplikasi di device								
4	Pelatihan Pengguna								
6	Ujicoba Sistem								
7	Evaluasi dan Perbaikan Sistem								

IV. PENUTUP

A. Kesimpulan

Kesimpulan Pemanfaatan Phonegap Untuk Membangun Aplikasi *Multiplatform* (android dan iOS) Dalam Menentukan Arah Kiblat Dan Waktu Sholat Dengan Menggunakan Gps

1. Phonegap sebagai cross framework platform bisa dimanfaatkan untuk membangun aplikasi-aplikasi yang dapat berjalan di beberapa platform hanya dengan menggunakan code yang sama.
2. Aplikasi dzikro sebagai cross aplikasi bisa dijadikan alternatif untuk melihat jadwal sholat, arah kiblat dan masjid sekitar.
3. Penggunaan Icenium dalam pembuatan aplikasi sangat membantu, terutama jika pengembang tidak mempunyai developer key, karena dengan memanfaatkan icenium, aplikasi bisa di install di device dengan bantuan icenium ion.

B. Saran

Saran yang dapat diberikan atas penelitian Pemanfaatan Phonegap Untuk Membangun Aplikasi *Multiplatform* (android dan iOS) Dalam Menentukan Arah Kiblat Dan Waktu Sholat Dengan Menggunakan Gps

1. Fitur dalam aplikasi dzikro masih belum banyak mungkin kedepannya bisa ditambahkan pengingat waktu sholat, dan widget yang dapat memudahkan pengguna untuk melihat waktu sholat tanpa harus membuka aplikasi.
2. Dapat ditambahkan rute terdekat ke tempat masjid yang dipilih, sehingga memudahkan pengguna jika ingin ke masjid yang di pilih.

3. Secara teori aplikasi dzikro bisa dipakai untuk platform yang lain, mungkin kedepannya bisa coba platform lainnya seperti blackberry dan windows phone.
4. Dengan semakin banyaknya platform mobile, mungkin aplikasi hybrid bisa menjadi solusi untuk membuat satu aplikasi yang bisa berjalan diberbagai platform.

DAFTAR PUSTAKA

- [1] Neni Nuraeni Dra. M.Ag. (2008,). *Tuntunan Shalat Lengkap dan Benar*. (T. P. ANDI, Penerj.) Mutiara Media.
- [2] Firtman, M. (2010). *Programming the Mobile Web*. Inc. Sebastopol, CA.: O'Reilly Media.
- [3] Sayed Hashimi, e. a. (2010). *Pro Android 2*. New York: Apress.
- [4] Ios Apple. (2013). *Ios*. Dipetik Februari 24, 2013, dari www.apple.com: <http://www.apple.com/ios/what-is>
- [5] Paananen, T. (2011). *Smartphone Cross-Platform Frameworks – A Case Study*. JAMK University of Applied Science.
- [6] Puputti, K. (2012). *Mobile HTML5:Implementing a Responsive Cross-Platform Application*. Aalto University.
- [7] Inderjeet Singh, M. P. (2011). *Comparison Of Cross-Platform Mobile Development Tools*. Malardalen University Innovation, Development and Technology.
- [8] Adobe Systems Inc. (2013). *Phonegap*. Dipetik Februari 24, 2013, dari Phonegap: <http://phonegap.com/>
- [9] Sarah Allen, e. a. (2010). *Pro Smartphone Cross-Platform Development: iPhone, Blackberry, Windows Mobile and Android Development and Distribution*. New York: Apress.
- [10] Inderjeet Singh, M. P. (2011). *Comparison Of Cross-Platform Mobile Development Tools*. Malardalen University Innovation, Development and Technology.
- [11] Geodesi, K. K. (2011). *Teknologi GPS*. Dipetik Februari 24, 2013, dari [geodesy: http://geodesy.gd.itb.ac.id/kkgd/?page_id=498](http://geodesy.gd.itb.ac.id/kkgd/?page_id=498)
- [12] IBM. (2012). Native, web or hybrid mobile-app development. *Native, web or hybrid mobile-app development*, 3.
- [13] Ibrahim, M., & Norashikin, M. (2009). *Universal Qibla and Prayer Time Finder*. Universiti Malaysia Pahang.
- [14] Abdali, S. K. (1997). *The Correct Qibla*. Washington, D.C. 20035: National Science Foundation.
- [15] Dawson, C. W. (2009). *Project in Computing and Information System: a Student Guide, 2nd Edition*. Addison-Wesley.
- [16] O'Brien, A. J. (2006). *Introduction to Information Systems, 12 th ed Penerj. Dewi Fitriyani dan A, Kwary Deny*. Jakarta: Salemba Empat.
- [17] Whitten, J. A., & et.al. (2004). *Systems Analysis and Design Methods, 1st ed*. (T. P. ANDI, Penerj.) Yogyakarta: ANDI.
- [18] Pressman, S. R. (2012). *Software Engineering: A Practitioner's Approach, 7th ed*. (A. Nugroho, J. L. George, & et.al., Penerj.) Yogyakarta: ANDI.
- [19] Nasution, S. (2009). *Metode Research*. Jakarta: Bumi Aksara.
- [20] Rinto Anugraha, D. (2009). *Cara Menghitung Waktu Shalat*. Dipetik Februari 24, 2013, dari Cara Menghitung Waktu Shalat: <http://www.erasuslim.com/peradaban/ilmu-hisab/cara-menghitung-waktu-shalat.htm>
- [21] Dennis, A., & at.al. (2009). *"Systems Analysis and Design with UML – 3rd Edition"*. John Wiley & Sons, Inc.