

Pengembangan Aplikasi Pengingat Salat Dengan Konsep *Context-Aware* Menggunakan MVP Pada Platform Android

Ibrahim Yunus Muhammad Fiqhan¹, Adam Hendra Brata², Agi Putra Kharisma³

Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya
Email: ¹ibrahimfiqhan@student.ub.ac.id, ²adam@ub.ac.id, ³agi@ub.ac.id

Abstrak

Perangkat *smartphone* semakin populer di kalangan masyarakat. Semakin banyaknya fungsionalitas yang ditawarkan oleh *smartphone*, orang-orang semakin sulit untuk lepas darinya. Karena hal itu *smartphone* sangat cocok sebagai alat untuk mengingatkan tentang agenda-agenda yang dimiliki oleh penggunanya. Banyak sekali aplikasi pengingat untuk *smartphone* tetapi pengingat tersebut masih memiliki kekurangan yaitu aplikasi tidak mampu mendeteksi apakah pengguna telah melaksanakan kegiatan yang telah diagendakan. Hal yang sama terjadi pada aplikasi pengingat salat. Aplikasi pengingat salat yang ada saat ini hanya mampu untuk mengingatkan pengguna ketika memasuki waktu salat. Aplikasi tidak mampu untuk mendeteksi apakah pengguna sudah melaksanakan salat atau belum. Untuk itu diperlukan konsep *context-aware* yang diharapkan mampu mengatasi kekurangan dari aplikasi pengingat salat. Dengan mendapatkan informasi konteks yang dimiliki oleh pengguna berupa aktivitas pengguna yang didapatkan melalui sensor-sensor yang ada pada perangkat, aplikasi dapat mengetahui apakah pengguna telah selesai melaksanakan salat sehingga aplikasi dapat memberikan pengingat sesuai dengan aktivitas pengguna tersebut. Informasi konteks yang digunakan untuk mendeteksi aktivitas pengguna adalah jenis kelamin pengguna dan beberapa keadaan perangkat. Untuk mendapatkan keadaan perangkat digunakan sensor kedekatan, sensor akselerometer, sensor cahaya, dan sensor gps. Implementasi dari aplikasi ini menggunakan arsitektur *Model-View-Presenter* (MVP) yang tiap-tiap kebutuhan dijadikan *package* yang berbeda yang di dalamnya terdapat *view* dan *presenter*.

Kata kunci: *pengingat, salat, context-aware, sensor, mvp*

Abstract

The smartphone device is getting popular among the people. The more functionality offered by smartphones, the more difficult for people to get away of it. Because that, smarphone is very suitable as a tool to remind about the agendas owned by users. There are so many reminder apps for smartphone but the reminder still has a drawback that the app is not able to detect whether the user has performed the activities that have been scheduled. The same thing happened in the prayer reminder app. The current application of prayer reminder is only able to remind the user when entering prayer time. The application is unable to detect whether the user has performed the prayer or not. For that required context-aware concept that is expected to overcome the deficiency of the application of prayer reminder. By obtaining user-defined context information in the form of user activity obtained through existing sensors on the device, the app can determine whether the user has finished performing the prayers so that the app can provide reminders according to the user's activity. The context information used to detect user activity are the user's gender and the state of device. To get the state of the device used proximity sensor, acceleromometer sensor, light sensor, and gps sensor. Implementation of this application using Model-View-Presenter (MVP) architecture that each requirements separated as a different package in which there is a view and presenter.

Keywords: *reminder, prayer, context-aware, sensor, mvp*

1. PENDAHULUAN

Teknologi berkembang dengan sangat pesat. Teknologi yang berkembang sangat pesat saat ini adalah *smartphone/mobile* di mana

penggunanya yang telah mencapai angka 2,32 milyar untuk di seluruh dunia dan diperkirakan pada tahun 2018 pengguna smartphone di Indonesia mencapai seratus juta pengguna (kominfo.go.id). Dengan semakin populernya smartphone di Indonesia, smartphone dituntut untuk dapat memenuhi semua kebutuhan pengguna. Hal ini didukung oleh pengembang di seluruh dunia sehingga dapat meningkatkan fungsionalitas. Semakin banyaknya fungsionalitas yang ditawarkan oleh smartphone, orang-orang semakin sulit untuk lepas darinya. Karena hal itu smartphone sangat cocok sebagai alat untuk mengingatkan tentang agenda-agenda yang dimiliki oleh penggunanya. Sudah banyak sekali aplikasi pengingat untuk smartphone atau mobile tetapi pengingat tersebut masih memiliki kekurangan yaitu pengingat hanya mengingatkan berdasarkan waktu sehingga pengguna kurang termotivasi untuk melakukan agendanya tersebut walaupun telah diingatkan oleh aplikasi pengingat. Kekurangan yang lain adalah aplikasi tidak mampu mendeteksi apakah pengguna telah melakukan suatu kegiatan yang telah diagendakan tersebut.

Hal yang sama terjadi pada aplikasi pengingat salat. Aplikasi pengingat salat yang ada saat ini kebanyakan hanya mampu untuk mengingatkan pengguna ketika memasuki waktu salat. Aplikasi tidak mampu untuk mendeteksi apakah pengguna sudah melaksanakan salat atau belum sehingga aplikasi tidak bisa untuk terus mengingatkan ketika pengguna belum melaksanakan salat.

Dipaparkan dalam paper Andreas Hamper yang berjudul “A Context Aware Mobile Application for Physical Activity Promotion” untuk mengajak agar pengguna melakukan suatu kegiatan diperlukan beberapa tahapan, yang pertama adalah menentukan kegiatan apa yang akan ditawarkan kepada pengguna, kedua adalah menentukan kapan dan bagaimana untuk mengajak pengguna, dan yang ketiga adalah mendeteksi informasi konteks dari pengguna saat ini berdasarkan waktu, lokasi, aktivitas, dan identitas. Dari hasil penelitian di atas nantinya akan diterapkan untuk mengembangkan aplikasi pengingat salat yang lebih efektif. Untuk mendapatkan konteks-konteks yang sekiranya dapat dimanfaatkan untuk menjadikan aplikasi pengingat salat lebih efektif dilakukan analisis terhadap calon pengguna.

Pengembangan aplikasi android sendiri dapat dikembangkan menggunakan beberapa metodologi. Salah satu metodologi yang populer

untuk pengembangan aplikasi android adalah pengembangan menggunakan arsitektur MVP (Model-View-Presenter). Arsitektur MVP pertama kali muncul pada IBM dan terlihat lebih jelas pada Taligent tahun 1990. Ketika menggunakan arsitektur MVP: a) Background-task dipisahkan dari kegiatan / pandangan / fragmen untuk membuatnya tidak bergantung sepenuhnya peristiwa terkait siklus hidup, b) Tugas kompleks dibagi menjadi lebih sederhana dan lebih mudah dipecahkan, c) Membuat pandangan mandiri dari sumber data dan d) Memfasilitasi pengujian unit otomatis. Semua fitur ini membuat arsitektur MVP lebih nyaman Bagi siswa dengan keterampilan pemrograman terbatas atau minimal siapa ingin mengembangkan aplikasi seluler (Ojeda-Guerra, 2015).

Dengan beberapa keunggulan arsitektur MVP yang telah dipaparkan di atas maka arsitektur MVP dapat digunakan sebagai solusi pengembangan aplikasi android. Oleh karena itu untuk menyelesaikan permasalahan tentang kekurangan aplikasi pengingat salat yang belum mampu mendeteksi pengguna telah melaksanakan salat atau belum, penulis melakukan penelitian dengan judul “Pengembangan Aplikasi Pengingat salat dengan Konsep Context-Aware Menggunakan MVP pada Platform Android”.

2. KAJIAN TEORI

2.1. Context Aware

Dalam konsep *Context-Aware*, aplikasi menggunakan metode autentikasi yang berbeda, yaitu berdasarkan konteks perangkat yang digunakan. Hal yang memicu perbedaan autentikasi tersebut adalah suatu lokasi dari perangkat tersebut yang didapat dari nilai longitud dan latitudnya. Selain dari lokasi, metode autentikasi dapat diatur lebih lanjut untuk menambahkan parameter lain seperti orientasi dan gerakan dari sebuah perangkat tersebut (Ranasinghe, 2016).

Selain yang telah disebutkan di atas informasi konteks juga dapat berupa aktivitas pengguna, identitas pengguna, dan waktu. (Hamper, 2015). Dijelaskan bahwa masing-masing parameter tersebut adalah sebagai berikut:

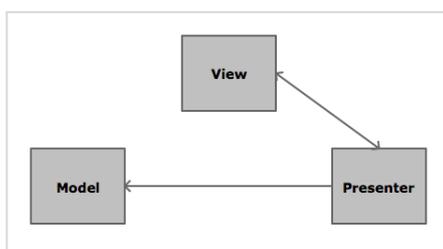
1. Waktu: Pagi/malam, Hari kerja/akhir pekan, waktu berlatih.

2. Lokasi: Tempat kerja, rumah, ataupun Gym.
3. Aktivitas: Steps, burned calories, activity types, intensity of activity
4. Identitas: Jenis kelamin, usia, komposisi tubuh, detak jantung, tekanan darah, kadar gula dalam darah, dan keadaan motivasional.

2.2. Arsitektur MVP

Arsitektur MVP (Model-View-Presenter) yang pertama kali muncul di IBM dan lebih terlihat di Taligent the 1990's. Arsitektur ini, sama seperti pola desain lainnya, decouples development dengan cara yang memungkinkan beberapa pengembang bekerja dan menguji secara bersamaan. Penting untuk disebutkan bahwa pengembang Android memiliki masalah yang timbul dari fakta bahwa aktivitas Android digabungkan erat dengan mekanisme akses antarmuka dan data. Agar aplikasi mudah dikembangkan dan pengembang yang mudah dipelihara perlu mendefinisikan lapisan yang terpisah dengan baik. Jadi, dengan menggunakan MVP: a) Tugas latar belakang dipisahkan dari aktivitas / pandangan / fragmen agar tidak terlepas dari sebagian besar kejadian terkait siklus hidup, b) Tugas kompleks dipecah menjadi tugas yang lebih sederhana dan mudah dipecahkan, c) Membuat pandangan independen dari sumber data dan d) Memfasilitasi pengujian unit otomatis. Semua fitur ini membuat arsitektur MVP lebih nyaman bagi siswa dengan keterampilan pemrograman terbatas atau minimal yang ingin mengembangkan aplikasi mobile (Ojeda-Guerra, 2015).

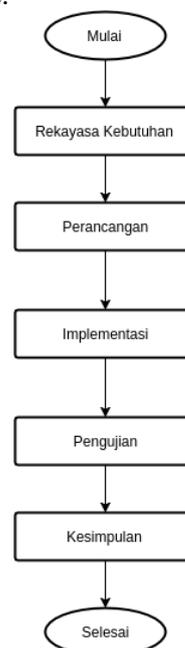
MVP (Model-view-presenter) adalah turunan dari pola MVC (Model-view-controller), perbedaan yang signifikan dari keduanya adalah pada MVP View tidak disarankan untuk berkomunikasi langsung dengan Model melainkan harus melalui perantara atau yang disebut juga dengan Contract (Suryanto, 2017). Representasi diagram dari pola MVP yang dapat dilihat pada Gambar 1.



Gambar 1. Diagram Pola MVP

3. METODOLOGI

Metodologi pada penelitian ini dapat dilihat pada Gambar 2.



Gambar 2. Metodologi

4. REKAYASA KEBUTUHAN

4.1. Gambaran Umum Sistem

Sistem Pengingat salat dengan konsep *context-aware* ini adalah sistem yang bertujuan untuk mengingatkan pengguna ketika memasuki waktu salat yaitu waktu salat subuh, zuhur, asar, magrib, dan isya. Penyampaian pengingat dipicu oleh waktu. Pengingat akan dimatikan jika sistem mendeteksi bahwa pengguna telah melaksanakan salat.

Untuk mengetahui pengguna telah melaksanakan salat dilakukan pengecekan pada beberapa kondisi yaitu perangkat berada pada mode *silent*, perangkat berada di masjid, perangkat sedang ditutup atau dibalik, perangkat berada di tas, perangkat berada dikantong atau saku, dan perangkat dibawa ke masjid. Jika kondisi-kondisi tersebut memenuhi selama waktu minimal malaksanakan salat, maka pengguna dideteksi telah melaksanakan salat dan sistem berhenti untuk menampilkan pengingat untuk waktu salat tersebut.

4.2. Analisis Kebutuhan

Tahap analisis kebutuhan pada penelitian ini dilakukan dengan menggunakan metode kuesioner. Kuesioner dilakukan untuk mendapatkan data tentang kebiasaan pengguna

terhadap perangkat, dalam hal ini smart phone dengan sistem operasi android, ketika pengguna melaksanakan salat. Selain itu juga didapatkan data berupa lama waktu pengguna dalam melaksanakan salat untuk masing-masing waktu salat.

Survey dilakukan pada sample 30 mahasiswa laki-laki dan 30 mahasiswa perempuan. Hasil dari survey tersebut adalah kebiasaan pengguna terhadap perangkat mereka pada saat mereka salat adalah sebagai berikut:

1. Perangkat di-silent
2. Perangkat berada di rumah atau ditinggal
3. Perangkat ditutup (dengan case atau diletakkan terbalik)
4. Perangkat dicas
5. Perangkat berada di tas
6. Perangkat berada di kantong
7. Perangkat dibawa ke masjid
8. Perangkat dimatikan

Untuk data lama waktu salat pengguna adalah diambil rata-rata masing-masing waktu salat yang didapatkan hasil berikut:

- a. Salat subuh: 8,1 menit untuk laki-laki dan 8,07 menit untuk perempuan
- b. Salat zuhur: 9,23 menit untuk laki-laki dan 8,5 menit untuk perempuan
- c. Salat asar: 9,23 menit untuk laki-laki dan 8,5 menit untuk perempuan
- d. Salat magrib: 9,17 menit untuk laki-laki dan 9 menit untuk perempuan
- e. Salat isya: 9,23 menit untuk laki-laki dan 8,4 menit untuk perempuan

Dari hasil survey tersebut dirumuskan untuk mendapatkan konteks berupa aktivitas pengguna atau secara khusus mendapatkan informasi apakah pengguna telah melaksanakan salat atau belum dengan aturan sebagai berikut:

Melakukan pengecekan terhadap keadaan perangkat

- a. Perangkat sedang di-silent
- b. Perangkat sedang ditutup atau dibalik
- c. Perangkat sedang berada di tas
- d. Perangkat sedang berada di kantong atau saku
- e. Perangkat sedang berada di masjid.

Dari keadaan-keadaan tersebut kemudian dikombinasikan untuk meningkatkan akurasi dengan kombinasi yang terdapat pada Tabel 1.

Tabel 1. Rancangan Analisis Komputasi

No.	Keadaan 1	Keadaan 2
1	Perangkat di-silent	Perangkat ditutup
2.	Perangkat ditutup	Perangkat berada di masjid

3.	Perangkat berada di tas	Perangkat berada di masjid
4.	Perangkat di-silent	Perangkat berada di masjid
5.	Perangkat berada di kantong atau saku	Perangkat berada di masjid

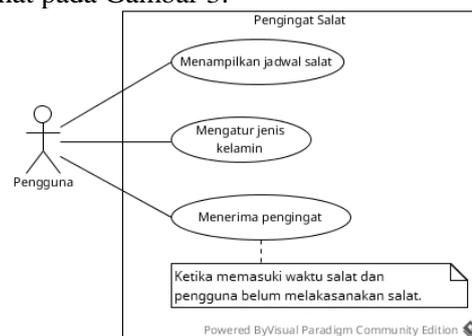
Jika kombinasi tersebut terpenuhi maka sistem akan mendeteksi bahwa pengguna sedang melaksanakan salat dan memulai hitung mundur sesuai dengan durasi salat. Dari hasil survey durasi salat tersebut disederhanakan menjadi:

- a. Salat subuh: 8 menit untuk laki-laki dan 8 menit untuk perempuan
- b. Salat zuhur: 9 menit untuk laki-laki dan 8 menit 30 detik untuk perempuan
- c. Salat asar: 9 menit untuk laki-laki dan 8 menit 30 detik untuk perempuan
- d. Salat magrib: 9 menit untuk laki-laki dan 9 untuk perempuan
- e. Salat isya: 9 menit untuk laki-laki dan 8 menit 30 detik untuk perempuan

Berdasarkan keadaan perangkat yang akan diambil atau dicek yang telah dijabarkan di atas dapat diketahui bahwa sistem akan membutuhkan beberapa sensor untuk mendapatkan keadaan-keadaan tersebut. Sensor-sensor yang dibutuhkan antara lain adalah sensor kedekatan (proximity sensor) untuk mendeteksi apakah perangkat sedang berada di kantong atau saku, sensor akselerometer (accelerometer sensor) untuk mendeteksi apakah perangkat sedang dibalik, sensor cahaya (light sensor) untuk mengecek apakah perangkat sedang berada di tas, sensor lokasi (GPS) untuk mengecek apakah perangkat sedang berada di masjid, dan Audio Manager dari sistem operasi android untuk mengecek apakah perangkat sedang dalam mode silent.

4.3 Kebutuhan Fungsional

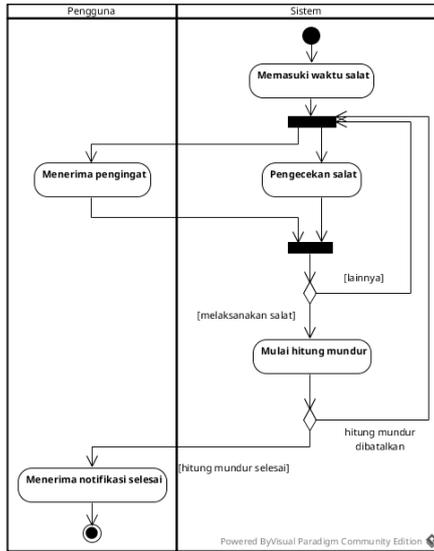
Perancangan kebutuhan sistem ini dapat dilihat pada Gambar 3.



Gambar 3. Use Case Diagram

Terdapat tiga *use case* pada perancangan sistem pengingat salat. Aktor dari sistem ini adalah aktor umum yang dapat menggunakan sistem. Terdapat tiga fungsi utama dari sistem ini yaitu menampilkan jadwal salat, mengatur data jenis kelamin, dan menerima pengingat.

5. PERANCANGAN



Gambar 4. Activity Diagram

5.1. Activity Diagram

Activity diagram yang terlihat pada Gambar 4 adalah activity diagram yang menjelaskan alur pemberian dan penghentian pengingat. Pada activity diagram tersebut dijelaskan bahwa pengecekan aktivitas pengguna yang berupa salat dilakukan ketika memasuki waktu salat. Setelah pengguna dideteksi telah melaksanakan salat maka sistem akan mematikan pengingat dan mengirimkan pemberitahuan selesai.

5.2. Perancangan Algoritma

Algoritma pengecekan salat ini akan dijalankan ketika ada perubahan state dari masing-masing sensor atau objek yang digunakan sebagai pengecek keadaan perangkat yang telah disebutkan di atas. Perancangan algoritma pengecekan salat dapat dilihat pada pseudocode berikut:

```

If inMosque is true and isSilent is true then
    isSalah set to true
else if inMosque is true and inBag is true then
    isSalah set to true
else if inMosque is true and inPocket is true then
    isSalah set to true
else if inMosque is true and isFacedown is true then
    isSalah set to true
else if isSilent is true and inPocket is true then
    isSalah set to true
    
```

```

else then
    isSalah set to false
end if
if isSalah is true and isStartCount is not true then
    isStartCount set to true
    startSalahCountDown
else if isSalah is not true and isStartCount is true then
    isStartCount set to true
    cancelSalahCountDown
end if
    
```

Algoritma pengecekan di masjid ini akan dijalankan ketika perangkat mengalami perubahan lokasi. Pengambilan lokasi masjid menggunakan API Google Place yang disediakan oleh Google. Perancangan algoritma ini dapat dilihat pada pseudocode berikut:

```

get masjid_location
If masjid_location not equals null then
    inMasjid set to true
else then
    inMasjid set to false
end if
    
```

Algoritma pengecekan ditutup menggunakan sensor akselerometer (*accelerometer sensor*). Algoritma ini akan dijalankan ketika sensor mendeteksi perubahan *state* dari sensor tersebut. Nilai dari *state* yang diambil adalah nilai sumbu z. Perancangan algoritma ini dapat dilihat pada pseudocode berikut:

```

get sensor_value
if sensor_value[z-axis] less than or equals -8.5 then
    isFacedown set to true
else then
    isFacedown set to false
end if
    
```

Algoritma pengecekan di kantong menggunakan sensor kedekatan. Algoritma ini akan dijalankan ketika sensor mendeteksi perubahan *state* dari sensor tersebut. Perancangan algoritma ini dapat dilihat pada pseudocode berikut:

```

get sensor_value
if sensor_value equals 0 then
    inPocket set to true
else then
    inPocket set to false
end if
    
```

Algoritma pengecekan di tas dijalankan menggunakan sensor cahaya. Algoritma ini akan dijalankan ketika sensor mendeteksi perubahan *state* dari sensor tersebut. Perancangan algoritma ini dapat dilihat pada pseudocode berikut:

```

get sensor_value
if sensor_value equals 0 then
    inBag set to true
else then
    inBag set to false
end if
    
```

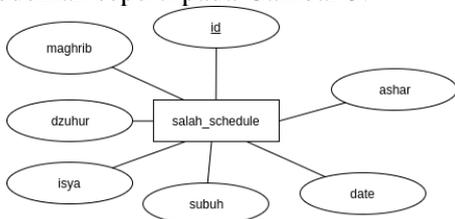
Algoritma pengecekan sedang dalam mode *silent* akan dijalankan ketika sistem mendeteksi perubahan *state* dari Android Audio manager yaitu *ringer* mode. Perancangan algoritma ini dapat dilihat pada *pseudocode* berikut:

```

get ringer_mode
if ringer_mode equals ringer_mode_silent then
    onSilent set to true
else then
    onSilent set to false
end if
    
```

5.3. Entity Relationship Diagram

Basis data pada sistem pengingat salat ini hanya terdapat satu entitas yaitu jadwal salat yang memiliki beberapa entitas yaitu *id*, *date*, *subuh*, *dzuhur*, *ashar*, *magrib*, *isya* yang dimodelkan seperti pada Gambar 5.



Gambar 5. Entity Relationship Diagram

5.4. Perancangan Antarmuka

Perancangan antarmuka jadwal salat dapat dilihat pada gambar 6. Tampilan ini adalah antarmuka yang pertama kali ditampilkan ketika pengguna membuka aplikasi. Pada antarmuka ini pengguna dapat mengetahui kota, tanggal, dan jadwal salat pada waktu itu. Pada antarmuka ini terdapat tombol setting pada toolbar yang berfungsi untuk menuju ke antarmuka pengaturan.



Gambar 6. Antarmuka Jadwal Salat

6. IMPLEMENTASI

6.1. Implementasi Algoritma

Tabel 2. Kode Program Method salahCheck
 salahCheck ()

```

1 private void salahCheck() {
2     if (inMosque && isSilent)
3         isSalah = true;
4     else if (inMosque && inBag)
5         isSalah = true;
6     else if (inMosque && inPocket)
7         isSalah = true;
8     else if (inMosque && isFacedown)
9         isSalah = true;
10    else if (isSilent && isFacedown)
11        isSalah = true;
12    else isSalah = false;
13
14    if (isSalah && !isCountStarted) {
15        isCountStarted = true;
16        salahCountDown.start();
17    } else if (!isSalah &&
18    isCountStarted) {
19        isCountStarted = false;
20        salahCountDown.cancel();
21    }
    }
    
```

Implementasi dari algoritma pengecekan salat terdapat pada method *salahCheck()* yang ada pada kelas *SalahService* dengan kode pada Tabel 2. Pada method tersebut dilkukan pengecekan nilai dari variabel *inMosque*, *isSilent*, *inBag*, *inPocket*, dan *isFacedown*. Jika *inMosque* dan *isSilent* bernilai benar maka *isSalah* diberi nilai benar, jika *inMosque* dan *inBag* bernilai benar maka *isSalah* diberi nilai benar, jika *inMosque* dan *inPocket* bernilai benar maka *isSalah* diberi nilai benar, jika *inMosque* dan *isFacedown* bernilai benar maka *isSalah* diberi nilai benar, jika *isSilent* dan *isFacedown* bernilai benar maka *isSalah* diberi nilai benar, selain itu *isSalah* diberi nilai salah. Kemudian melakukan pengecekan jika *isSalah* bernilai benar dan *isCountStarted* bernilai salah maka *isCountStarted* diberi nilai benar, memulai penghitungan mundur dengan menjalankan *salahCountDown.start()*, jika *isSalah* bernilai salah dan *isCountStarted* bernilai benar maka *isCountStarted* diberi nilai salah dan membatalkan penghitungan mundur dengan menjalankan *salahCountDown.cancel()*.

6.2. Implementasi Basis Data

Implementasi basis data pada sistem ini menggunakan basis data *SQLite* yang dapat dibuat secara lokal pada perangkat android. Bentuk basis data setelah diimplementasikan direpresentasikan pada *physical data model* yang terdapat pada Gambar 7.



Gambar 7. Physical Data Model Jadwal Salat

6.3. Implementasi Antarmuka

Efektifitas dari penggunaan konsep context-aware terhadap aplikasi pengingat salat masih belum dapat dibuktikan. Oleh karena itu perlu penelitian lebih lanjut untuk menganalisa efektifitas tersebut.



Gambar 8. Antarmuka Pengingat

Antarmuka pada Gambar 8 adalah antarmuka utama sistem yang akan ditampilkan saat sistem dijalankan yang berfungsi untuk menampilkan jadwal salat.

7. PENGUJIAN

7.1. Pengujian Unit

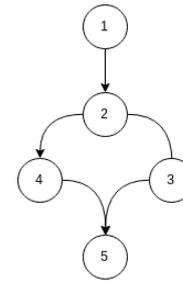
Pengujian unit dilakukan untuk memastikan bahwa implementasi pada kode program telah sesuai dengan algoritma yang direncanakan. Pada pengujian ini digunakan jUnit untuk menguji kode program yang telah diimplementasikan. Pengujian unit ini dilakukan pada method utama yang membangun sistem.

```
private void loadSalahSchedule() {
    Date currentDate = Calendar.getInstance().getTime();
    SimpleDateFormat simpleDateFormat = new SimpleDateFormat(
        pattern "yyyy-MM-dd"); // 1
    mSalahScheduleRepository.getSalahSchedule(simpleDateFormat.format(currentDate),
        new SalahScheduleDataSource().getSalahScheduleCallback() { // 2
        @Override
        public void onSalahScheduleLoaded(SalahSchedule salahSchedule) { // 3
            mSalahScheduleView.showSalahSchedule(mPreference
                .getString(key "location_preference", default "Kota Malang"),
                salahSchedule);
        }
        @Override
        public void onDataNotAvailable() { // 4
            mSalahScheduleRepository.refreshSalahSchedules(mPreference
                .getString(key "location_preference", default "Kota Malang")); // 4
        }
    }); // 5
}
```

Gambar 9. Kode Program

Pengujian unit method loadSalahSchedule adalah method yang digunakan untuk mengambil jadwal salat dari basis data kemudian ditampilkan pada view. Kode program method tersebut dapat dilihat pada Gambar 9.

Berdasarkan kode program tersebut dapat dibuat flow graph yang digambarkan pada Gambar 10.



Gambar 10. Flow Graph Method loadSalahSchedule

Dari graph pada Gambar 10 dapat ditentukan cyclomatic complexity dengan perhitungan sebagai berikut.

$$\begin{aligned}
 V(G) &= E-N+2 \\
 &= 5-5+2 \\
 &= 2
 \end{aligned}$$

Dari hasil cyclomatic complexity, dapat ditentukan jalur independen yaitu:

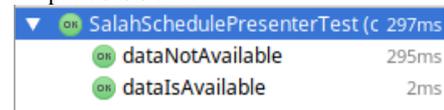
Jalur 1: 1-2-3-5

Jalur 2: 1-2-4-5

Sehingga dapat dibuat kasus uji seperti yang ditunjukkan pada Tabel 3 dan hasil pengujian menunjukkan bahwa pengujian ini valid seperti yang ditunjukkan Gambar 11.

Tabel 3. Kasus Uji

No.	Kasus Uji	Hasil yang diharapkan
1	Mengambil data yang sudah ada di basis data	Menjalankan method onSalahScheduleLoaded ()
2	Mengambil data yang tidak ada pada basis data	Menjalankan method onDataNotAvailable



Gambar 11. Hasil Pengujian

8. KESIMPULAN DAN SARAN

8.1. Kesimpulan

Berdasarkan hasil analisis kebutuhan yang dilakukan dengan survey menggunakan kuesioner didapatkan beberapa kebiasaan calon pengguna ketika melaksanakan salat. Kebiasaan-kebiasaan tersebut diambil kesimpulan untuk dijadikan konteks. Konteks-konteks yang didapatkan adalah lokasi, jenis kelamin, dan aktivitas. Aktivitas yang dimaksud adalah aktivitas yang berhubungan dengan perangkat android yaitu perangkat berada di masjid, perangkat dalam mode silent, perangkat dibalik, perangkat di dalam saku, dan perangkat di dalam tas.

Untuk mendapatkan konteks-konteks yang telah dijelaskan pada nomor satu digunakan beberapa cara. Pertama adalah dengan masukan

dari pengguna yaitu untuk mendapatkan konteks berupa jenis kelamin. Selanjutnya dengan menggunakan sensor-sensor yang ada antara lain yaitu sensor cahaya untuk mendapatkan konteks di dalam tas, sensor kedekatan untuk mendapatkan konteks di dalam saku, sensor akselerometer untuk mendapatkan konteks perangkat sedang dibalik, dan sensor GPS yang digunakan untuk mendapatkan konteks perangkat sedang berada di masjid.

Perancangan dan implementasi dari sistem pengingat salat menggunakan arsitektur MVP. Yang pada setiap fitur atau kebutuhannya dikelompokkan pada masing-masing package, pada masing-masing package tersebut terdapat view dan presenter tersendiri. Presenter berfungsi untuk komunikasi dengan model untuk mengelola data yang hasilnya disampaikan pada view. Pada arsitektur ini tidak terjadi komunikasi secara langsung antara view dengan model, melainkan melalui presenter.

Berdasarkan pengujian yang dilakukan dapat disimpulkan bahwa sistem telah berhasil mendapatkan konteks-konteks yang telah didefinisikan. Selain itu sistem juga sudah berhasil mengetahui aktifitas salat pengguna berdasarkan konteks yang didapat yang telah diimplementasikan sesuai algoritma dan dirancang sesuai dengan analisis.

8.2. Saran

Saran yang dapat disampaikan oleh penulis dari hasil penelitian ini untuk mendukung penelitian selanjutnya dengan topik yang sama adalah perlunya optimasi dalam penggunaan sensor untuk mendapatkan konteks agar sistem tidak memakan terlalu banyak daya.

9. DAFTAR PUSTAKA

- HAMPER, A., 2015. A Context Aware Mobile Application for Physical Activity Promotion. *Hawaii International Conference on System Sciences*, [e-journal] (48). 3197-3206. Tersedia melalui: IEEE Xplore Digital Library <<http://ieeexplore.ieee.org>> [Diakses 28 Mei 2017]
- OJEDA-GUERRA, C. N., 2015. A Simple Software Development Methodology Based on MVP for Android Applications in a Classroom Context. *International Conference on Computer and Information Technology*, [e-journal]. 1429-1434. Tersedia melalui:

IEEE Xplore Digital Library <<http://ieeexplore.ieee.org>> [Diakses 28 Mei 2017]

- RANASINGHE, Y.S., WALPOLA, M.J., 2016. Integrating Context-Awareness with Reminder Tools. *International Conference on Advances in ICT for Emerging Regions*, [e-journal] (1). 216-221. Tersedia melalui: IEEE Xplore Digital Library <<http://ieeexplore.ieee.org>> [Diakses 28 Mei 2017]
- SURYANTO, B. O. (2017, Januari 30). Implementasi MVP (Model-View-Presenter). Tersedia melalui: Medium: <<https://medium.com/@budioktaviyans/implementasi-mvp-model-view-presenter-2169a346acbb>> [Diakses 8 Januari 2018]