

PENGIMPLEMENTASIAN *CONTEXT AWARENESS* UNTUK MENGUBAH MODE BUNYI *ANDROID SMARTPHONE* TANPA USER INPUT

IMPLEMENTATION OF *CONTEXT AWARENESS* TO CHANGE *ANDROID SMARTPHONE'S RINGER MODE* WITHOUT USER INPUT

Arya Bagaskara¹, Dodi Wisaksono Sudiharto², Siti Amatulah Karimah³

Prodi S1 Informatika, Fakultas Informatika, Universitas Telkom Jl.
Telekomunikasi No.1, Terusan Buah Batu, Bandung 40257 Indonesia

¹aryagamis@gmail.com, ²dodiws@telkomuniversity.ac.id, ³karimahsiti@telkomuniversity.ac.id

Abstrak

Mode bunyi pada *smartphone* berfungsi untuk mengubah suara *smartphone* ketika ada notifikasi seperti adanya panggilan telepon, SMS, dan sebagainya. Biasanya *user* harus mengubah mode bunyi secara manual ketika akan melakukan aktifitas di tempat tertentu seperti pada tempat peribadatan. Kelalaian untuk mengubah mode bunyi, dapat menyebabkan *smartphone* berbunyi di saat yang tidak diinginkan, seperti saat beribadah

Pada dasarnya sudah ada beberapa aplikasi pihak ketiga yang menyediakan layanan tersebut, akan tetapi semuanya harus memasukkan lokasi secara manual. Sehingga ketika mengunjungi lokasi baru, *user* masih harus mengubah mode bunyi secara manual saat mengunjungi lokasi tersebut. Hal ini tidak nyaman untuk *user* yang sering berpindah-pindah ke lokasi baru, terutama manakala *user* berkunjung hanya sekali saja. Solusinya adalah dengan menggunakan sebuah aplikasi pengubah mode bunyi yang memiliki *location-awareness*. Aplikasi tersebut dapat mengubah mode bunyi secara otomatis tanpa campur tangan *user* sama sekali. *Android smartphone* telah memiliki API (*Application Programming Interface*) untuk menjalankan fungsi *context awareness* sebagai masukan yang dapat digunakan untuk mengubah mode bunyi secara otomatis. Sehingga pada studi ini akan dikembangkan aplikasi berbasis *Android smartphone* untuk mengubah mode bunyi secara otomatis dengan memanfaatkan inputan dari *context awareness*.

Kata kunci: *context-awareness, location-awareness, Android smartphone, mode bunyi*

Abstract

Smartphone ringer mode is used to change the phone mode when there are notifications such as phone call, SMS, etc. Habitually, the user needs to change manually the ringer mode when they are going to do an activity in certain place such as in a mosque. The user areless for changing the ringer mode can cause the smartphone rings inappropriately like when they are still in praying.

There are already third party applications which provide this kind service, but they still need for inputs. Furthermore, when the users come to a new location, they still need to set it up manually. This condition is uncomfortable for a mobile users who like to visit new place, especially when the users only visit a place for only one time. The solution is by using the application of ringer mode changer which is functioned by using a location-awareness input. It can change the ringer mode automatically without any user action. Android smartphone already have API (Application Programming Interface) of context awareness which can be used to utilize context awareness function as inputs to change ringer mode automatically. This study is going to develop Android smartphone application to change ringer mode automatically by utilizing the context awareness input.

Keywords: *context-awareness, location-awareness, Android smartphone, ringer mode*

1. Pendahuluan

1.1. Latar Belakang

Telepon genggam telah menjadi teknologi populer beberapa tahun belakangan ini. Perkembangan teknologi telepon genggam sangatlah pesat, sehingga saat ini telepon genggam telah berkembang menjadi telepon genggam cerdas atau yang biasa disebut *smartphone* [1].

Semenjak kehadirannya, telepon genggam memiliki fitur mode bunyi yang dapat diubah oleh pengguna untuk menyesuaikan dengan keadaan sekitarnya [2]. Pada umumnya mode bunyi tersebut adalah “bunyi”, “bergetar”, dan “diam”[2]. Meski dalam perkembangannya telepon genggam telah menjadi *smartphone*, fitur ini masih relevan digunakan karena manfaatnya dalam membantu memberikan notifikasi kehidupan sehari-hari, Sehingga fitur tidak dihilangkan pada fitur suatu *smartphone*.

Beberapa aplikasi pihak ketiga mencoba mengembangkan fitur mode bunyi seperti dengan memanfaatkan konteks waktu untuk mengubah mode bunyi secara terjadwal [3]. Aplikasi lainnya lagi, memanfaatkan konteks lokasi untuk mengubah mode bunyi sesuai dengan lokasinya [3]. Beberapa aplikasi lainnya melakukan kombinasi kedua konteks tersebut saat mengubah mode bunyi [4].

Dari semua studi yang ada, terdapat satu kesamaan, yaitu pengguna harus mengatur sendiri informasi lokasi dan kapan mode bunyi harus berubah, Sehingga saat pengguna memasuki suatu lokasi baru (saat mendatanginya untuk pertama kali), pengguna harus selalu mengatur mode bunyinya.

Pada studi ini diajukan sebuah aplikasi yang dapat memanfaatkan kemampuan *context-awareness* pada *Android smartphone* untuk mengubah mode bunyi secara otomatis tanpa ada masukan dari *user* sama sekali. Dengan penggunaan aplikasi ini, *user* dapat mendatangi suatu tempat yang baru dan *Android smartphone* akan secara otomatis menyesuaikan mode bunyinya, sehingga tidak terjadi hal-hal yang tidak diinginkan, seperti gangguan yang dapat ditimbulkan dari suara mode bunyi.

1.2. Identifikasi Masalah

Beberapa aplikasi pihak ketiga telah dikembangkan dalam memanfaatkan *context-awareness* pada *Android smartphone* untuk mengubah mode bunyi sesuai keadaan sekitar. Akan tetapi, pemanfaatannya belum sempurna, di mana pengguna masih perlu mengatur mode bunyi (secara manual) saat pertama kali mendatangi suatu tempat baru, atau saat melakukan kegiatan yang tidak terjadwal sebelumnya. Untuk menangani kasus-kasus tersebut, diperlukan pengembangan aplikasi *smartphone* dengan memanfaatkan *context-awareness* dengan pengaturan fitur secara otomatis (tanpa masukan dari pengguna).

1.3. Batasan Masalah

Batasan masalah pada rencana pengembangan aplikasi *context awareness* untuk mengubah mode bunyi *Android smartphone* tanpa *user input* adalah sebagai berikut:

1. Aplikasi berjalan pada perangkat *Android smartphone*.
2. *Android* yang digunakan merupakan versi *Lollipop (Android 5.1)*.
3. Perangkat *Android* memiliki sensor GPS.

1.4. Tujuan

Tujuan dari penelitian ini adalah :

1. Mengembangkan aplikasi yang memanfaatkan *context-awareness* pada *Android smartphone* untuk mengubah mode bunyinya.
2. Menerapkan otomatisasi pada perubahan mode bunyi pada aplikasi tersebut.
3. Menganalisis aplikasi yang dikembangkan.

2. Landasan Teori

2.1. Studi Literatur

Studi dilakukan menggunakan literatur dalam bentuk makalah dan artikel. Literatur yang digunakan membahas implementasi *context-awareness* dalam mengubah mode bunyi, seperti penggunaan secara kolaboratif [3] dan pemahaman konteks lokasi [5].

Akan tetapi dalam literatur yang tersedia tidak ada yang membahas otomatisasi perubahan tanpa adanya masukan dari pengguna, referensi aplikasi pihak ketiga yang tersedia dalam literatur juga memerlukan masukan pengguna untuk setiap lokasi dan waktu tertentu [3], [5].

2.2. Context-Awareness

Context-awareness adalah kemampuan dari suatu perangkat untuk mengetahui konteks yang tersedia di sekitarnya [6]. Konteks yang dimaksud bisa berupa tempat, waktu, cuaca, keramaian, kondisi perangkat, dan beragam konteks lainnya.

2.3. Android

Android adalah salah satu dari beberapa sistem operasi yang digunakan dalam *mobile device*. *Android* berjalan pada berbagai perangkat, mulai dari *smartphone*, *tablet*, *smart watch*, hingga perangkat *gaming*. *Android* memiliki populasi pengguna yang sangat banyak, memiliki perkembangan yang pesat sejak 2010 [7]. Sistem operasi *Android* dikembangkan oleh Google, dan saat ini telah mencapai versi 8.0 [8].

2.3.1. Android Studio

Android Studio adalah aplikasi yang dibuat secara resmi oleh *Google* untuk mengembangkan aplikasi untuk *Android* [9]. Aplikasi ini tersedia untuk *Windows*, *Linux*, dan *MacOS*. Saat ini *Android Studio* telah diluncurkan dalam versi 3.0 [10].

2.3.2. *Android API*

Android API adalah *interface* yang dapat digunakan untuk membantu memanfaatkan fitur-fitur yang tersedia dalam sistem operasi *Android* dan mengimplementasikannya ke dalam sebuah aplikasi. *API* memudahkan suatu aplikasi untuk digunakan di dalam berbagai perangkat selama perangkat tersebut menggunakan sistem operasi yang sama. *Android* menggunakan *API level* yang berbeda-beda untuk setiap versi sistem operasinya, dimana dilakukan pengembangan dalam fitur apa yang dapat diakses menggunakan *API* yang tersedia. *Android* memiliki kompatibilitas untuk menggunakan *API* pada versi sebelumnya, sehingga pada *Android* versi 8.0 masih bisa memanfaatkan aplikasi yang dibuat untuk *Android* versi 5.1.

2.4. *Google Awareness API*

Awareness API adalah *Android API* yang memungkinkan suatu aplikasi mengakses fitur *context-awareness* di dalam *Android* [11]. Dalam pengimplementasiannya, *Awareness API* terbagi menjadi 2, yaitu *Fence API* dan *Snapshot API*.

2.4.1. *Fence API*

Fence API merupakan bagian dari *Awareness API* yang dapat menjalankan suatu perintah saat suatu konteks terpenuhi [11]. Konteks yang disediakan oleh *Fence API* diantaranya adalah konteks lokasi berupa *latitude* dan *longitude*, aktifitas pengguna berupa aktifitas *idle*, berjalan, berlari, atau dalam kendaraan, status *headphone* terhubung atau tidak, dan pemancar terdekat.

2.4.2. *Snapshot API*

Snapshot API merupakan bagian dari *Awareness API* yang dapat memberikan konteks kepada sebuah aplikasi [11]. Konteks yang disediakan oleh *Snapshot API* diantaranya adalah konteks lokasi berupa *latitude* dan *longitude*, aktifitas pengguna berupa aktifitas *idle*, berjalan, berlari, atau dalam kendaraan, status *headphone* terhubung atau tidak, konteks tempat sesuai *Google Place* merujuk pada atribut tempat dalam *Google Places API*, serta konteks cuaca.

2.5. *Google Places API*

Places API adalah *Android API* yang dapat mengetahui informasi dari suatu tempat, seperti nama, alamat, lokasi berupa *latitude* dan *longitude*, nomor telpon, jenis tempat, *website* dari tempat tersebut, foto dari tempat tersebut, *review* tempat dan lainnya [12]. Data tempat disimpan dalam *Google database* dan pengembang aplikasi maupun pengguna dapat menambahkan tempat baru dan informasi di dalamnya ke dalam *database* dengan moderasi dari *Google*.

3. Perancangan Sistem

3.1. Perancangan Sistem

3.1.1. Analisis Kebutuhan Sistem

Aplikasi ini dibuat berdasarkan kebutuhan kenyamanan pengguna dalam penggunaan sebuah *smartphone*. Sebagai sebuah perangkat *mobile computing*, pengguna dari *Android smartphone* dapat menggunakan perangkatnya di berbagai tempat. Dengan seringnya perangkat *Android smartphone* berpindah tempat, pengguna harus teliti dalam mengubah mode bunyi agar perangkatnya tidak berbunyi di situasi yang tidak tepat. Atas dasar itulah dilakukan otomatisasi perubahan *ringer mode* untuk meningkatkan kenyamanan pengguna.

3.1.2. Gambaran Umum

Penerapan otomatisasi dilakukan dengan memanfaatkan *context-awareness* untuk mengubah *ringer mode*. Konteks yang tepat digunakan adalah konteks aktifitas, konteks tempat, serta konteks waktu. Konteks aktifitas digunakan untuk mengetahui aktifitas yang dilakukan oleh pengguna. Jika pengguna sedang berada dalam perjalanan maka tidak tepat rasanya jika *ringer mode* berubah-ubah karena dalam perjalanan lokasi pengguna berpindah dalam waktu cepat. Konteks tempat digunakan untuk mengetahui tempat yang tepat untuk *ringer mode* tertentu. Konteks waktu digunakan untuk membatasi waktu-waktu dimana perubahan *ringer mode* diperlukan.

Gambar 1. *Flow Chart* Proses Aplikasi

Gambar 2. *Use Case Diagram*

User berinteraksi dengan *MainActivity*. *MainActivity* digunakan untuk mengaktifkan *BackgroundService*. *BackgroundService* memiliki fungsi untuk mengecek apakah kondisi untuk menjalankan *RingerModeChanger* dipenuhi. *RingerModeChanger* adalah sebuah *BroadcastReceiver* yang berfungsi untuk mengubah *ringer mode* sesuai dengan konteks lokasi dan waktu, serta memberikan *feedback* berupa notifikasi ke user. *Google Play API*

adalah *actor* yang memberikan data konteks yang dimiliki oleh sistem untuk digunakan oleh aplikasi. *Audio Manager API* memberi izin pada *RingerModeChanger* untuk mengubah *ringer mode*.

3.1.3. Konteks

Dalam aplikasi ini *context awareness* dimanfaatkan untuk 2 fungsi yang berbeda. Fungsi pertama adalah untuk memberi syarat dijalankannya aplikasi bilamana konteks sesuai. Fungsi kedua adalah untuk mengetahui situasi dan mengubah *ringer mode* sesuai dengan data yang diterima.

a. Fence Context

Fence Context adalah konteks yang digunakan untuk memberi syarat terhadap aplikasi agar ia dapat mengubah *ringer mode*. Konteks yang digunakan untuk memberi syarat pada aplikasi ini adalah konteks aktifitas, di mana fungsi yang mengubah *ringer mode* baru akan dijalankan saat aktifitas *user* adalah diam (*idle*).

b. Snapshot Context

Snapshot context adalah konteks yang digunakan untuk mengetahui situasi di sekitar *smartphone*. Konteks pertama yang digunakan adalah konteks tempat yang disediakan oleh *Google Places API*. Konteks tempat ini bukanlah konteks lokasi berupa *latitude & longitude*, tetapi detail informasi tempat dalam *Google database*. Detail informasi ini berisi di antaranya nama tempat, nomor telepon, alamat, tipe, *website*. Konteks berikutnya yang digunakan adalah konteks lokasi berupa *latitude* dan *longitude* saat ini, konteks lokasi ini digunakan untuk meningkatkan akurasi dari aplikasi. Konteks ketiga yang digunakan adalah konteks waktu.

3.2. Desain Aplikasi

Gambar 3. Class Diagram

Untuk memulai *BackgroundService*, user menekan *start service*. Pada saat inisiasi, *BackgroundService* akan membuat *Activity Fence*. *Activity Fence* ini digunakan untuk mengirim broadcast ke *RingerModeChanger* apabila *Activity Fence* dalam kondisi diam (*idle*). Apabila tidak, *BackgroundService* akan menunggu sampai kondisi *fence* dipenuhi. Jika kondisi terpenuhi, *RingerModeChanger* akan menerima *broadcast* dan menjalankan *onReceive()*. *RingerModeChanger* kemudian meminta konteks kepada *Snapshot API*. Setelah konteks didapatkan, *RingerModeChanger* akan meminta tipe tempat yang kemudian dikembalikan dalam bentuk *integer list*. *RingerModeChanger* kemudian membandingkan *list* tersebut dengan *placeselection*. Jika tipe tempat termasuk pada *placeselection*, *RingerModeChanger* melakukan request ke *AudioManager* untuk mengubah *ringer mode* menjadi *vibrate*. Jika tidak termasuk, *ringer mode* diubah menjadi *normal*.

3.3. Skenario Pengukuran Performansi

3.3.1. Pengujian Fungsionalitas

Tujuan pengujian ini adalah untuk mengetahui apakah keseluruhan fungsionalitas aplikasi berjalan sesuai dengan rancangan. Pengujian dilakukan pada *Xiaomi Redmi Note 4*. Skenario yang digunakan adalah *Black Box Testing*.

3.3.2. Pengujian Akurasi

Pengujian akurasi dilakukan untuk mengetahui akurasi dari *context awareness* yang digunakan. Pengujian akurasi terutama difokuskan kepada akurasi identifikasi *Places API*.

3.3.3. Pengujian Kepuasan

Pengujian kepuasan dilakukan untuk mengetahui apakah aplikasi berjalan dengan baik. Target dari pengujian ini adalah kenyamanan dan keterbantuan user dalam penggunaan aplikasi. Pengujian ini akan dilakukan dengan menggunakan kuesioner.

4. Pengujian dan Analisis Sistem

4.1. Pengujian Fungsionalitas

Pengujian fungsionalitas dilakukan dengan *Black Box Testing* dan hasil pengujian ditampilkan pada tabel

Tabel 1. Skenario *Black Box Testing*

Kebutuhan Fungsional	Prosedur Pengujian	Ekspektasi Keluaran	Keluaran	Kesimpulan
Mengetahui aktifitas dan membuat <i>fence</i>	Bagian dari kode dibuat dalam sebuah aktifitas baru dan ditampilkan dalam bentuk <i>TextView</i>	<i>Fence</i> terbuat dan aktifitas dideteksi	<i>Fence</i> terbuat dan aktifitas dideteksi	Valid
Syarat <i>fence</i> dapat berjalan	Menggunakan <i>log tag</i> untuk melihat pada debug dan penggunaan pesan Toast untuk menampilkan notifikasi. Pengujian dilakukan tidak dengan menggunakan <i>ActivityFence</i> melainkan <i>HeadphoneFence</i> untuk efisiensi.	Logcat menampilkan pesan dan <i>smartphone</i> menampilkan Toast	Logcat menampilkan pesan dan <i>smartphone</i> menampilkan Toast	Valid
Dapat menerima konteks tempat, nama tempat serta tipe tempat	Membuat kode dalam aktifitas baru yang menampilkan nama tempat dan tipe tempat dalam <i>TextView</i>	Nama tempat dan tipe tempat dapat muncul pada aktifitas yang dibuat	Nama tempat dan tipe tempat dapat muncul pada aktifitas yang dibuat	Valid
Mengubah <i>Ringer Mode</i>	Memasukkan kode program dengan memanfaatkan <i>fence</i> yang sudah berjalan, pengamatan pada status bar dan pembuatan sebuah toast untuk menampilkan pesan sukses	Status bar menampilkan perubahan mode bunyi, serta munculnya pesan <i>toast</i>	Status bar menampilkan perubahan mode bunyi, serta munculnya pesan <i>toast</i>	Valid
Menggunakan konteks tempat untuk mengubah <i>ringer mode</i>	Menggabungkan kode sebelumnya dan menggunakan toast untuk notifikasi	Toast menampilkan nama lokasi dan tipe lokasi serta mengubah mode bunyi	Toast menampilkan nama lokasi dan tipe lokasi serta mengubah mode bunyi	Valid
Menggunakan konsep waktu untuk mengubah mode bunyi	Menggabungkan kode sebelumnya dan menggunakan toast untuk notifikasi	<i>Toast</i> menampilkan nama lokasi dan tipe lokasi serta mengubah mode bunyi sesuai waktu	<i>Toast</i> mengubah mode bunyi pada hari kerja	Valid

4.2. Pengujian Akurasi

Pada *Black Box Testing* telah dilakukan pengujian fungsionalitas. Pengujian Akurasi berfungsi untuk mengetahui keberhasilan dari *Snapshot API* untuk memunculkan hasil yang akurat. Hasil dari pengujian ditampilkan pada tabel berikut :

5. Kesimpulan dan Saran

5.1 Kesimpulan

Kesimpulan berdasarkan hasil analisis adalah :

1. Berdasarkan hasil pengujian fungsionalitas, fungsi dari aplikasi dapat berjalan dengan baik. Aplikasi berjalan sesuai dengan desain untuk mengetahui dan memproses konteks yang ada untuk mengubah *ringer mode* pada perangkat.
2. Berdasarkan hasil pengujian akurasi, aplikasi dapat mengetahui konteks lokasi dengan baik.
3. Berdasarkan pengujian akurasi, aplikasi masih sulit mendapat konteks tempat yang sangat berdekatan dengan akurat, hal ini dikarenakan aplikasi memiliki radius akurasi sebesar 20 meter.
4. Berdasarkan hasil pengujian akurasi, aplikasi sulit untuk mendapatkan konteks tempat di lokasi yang besar. Ini dikarenakan karena pada lokasi yang besar, marker penanda lokasi hanya berada pada satu titik.
5. Berdasarkan pengujian kepuasan, aplikasi telah memiliki tingkat kepuasan yang baik. 25 responden merasa terbantu dengan adanya aplikasi ini.

5.2 Saran

Saran untuk penelitian selanjutnya :

1. Peningkatan akurasi juga dapat dilakukan dengan mendaftarkan tempat-tempat yang belum terdaftar di *Google Places* serta dengan menggunakan *database* lokasi sendiri, dibandingkan hanya bergantung kepada *Google Places*.
2. Peningkatan kepuasan dapat dilakukan dengan memberikan *user* pilihan *ringer mode* yang diinginkan pada jenis lokasi yang sesuai.

Daftar Pustaka :

- [1] T. Page, "Smartphone technology , consumer attachment and mass customisation," Internal Journal of Green Computing, vol. 4, no. 12, pp. 38–40, December 2013.
- [2] Y. Chang, J. C. Tang, and A. Arbor, "Investigating mobile users ringer mode usage and attentiveness and responsiveness to communication," in Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services., 2015, pp. 6–15.
- [3] R. A. Dobrican, G. I. F. Neyens, and D. Zampunieris, "A context-aware collaborative mobile application for silencing the smartphone during meetings or important events," International Journal On Advances in Intelligent Systems, vol. 9, no. 1, pp. 171–180, 2016.
- [4] "Auto Silent Mode." [Online]. Available: <https://play.google.com/store/apps/details?id=com.silentonlocations>. [Accessed: 25-Apr-2018].
- [5] "Silent on Locations." [Online]. Available: <https://play.google.com/store/apps/details?id=com.silentonlocations>. [Accessed: 15-Apr-2018].
- [6] G. D. Abowd, A. K. Dey, R. Orr, and J. Brotherton, "Context-awareness in wearable and ubiquitous computing," in Wearable Comput., 1997, pp. 200–211.
- [7] M. Butler, "Android: changing the mobile landscape" in IEEE Pervasive Computing, 2010, pp. 4–7.
- [8] S. R. Pawar and M. Kulati, "Recent versions on android and survey on the versions lollipop , marshmallow and nougat," Int. Res. J. Eng. Technol., vol. 5, no. 3, pp. 1186–1189, 2018.
- [9] E. Protalinski, "Google releases Android Studio 1.0, the first stable version of its IDE," VentureBeat, 2014. [Online]. Available: <https://venturebeat.com/2014/12/08/google-releases-android-studio-1-0-the-first-stable-version-of-its-ide/>. [Accessed: 24-Jul-2018].
- [10] F. Lardinois, "Google announces the first preview of Android Studio 3.0, puts emphasis on speed and smarts," TechCrunch, 2017. [Online]. Available: <https://techcrunch.com/2017/05/17/google-announces-the-first-preview-of-android-studio-3-0-puts-emphasis-on-speed-and-smarts/>. [Accessed: 24-Jul-2018].
- [11] B. Vigliarolo, "Google Awareness API: The smart person's guide," TechRepublic, 2017. [Online]. Available: <https://www.techrepublic.com/article/google-awareness-api-the-smart-persons-guide/>. [Accessed: 24-Jul-2018].
- [12] M. Singhal and A. Shukla, "Implementation of location based services in android using GPS and web services," Int. J. Comput. Sci. Issues (IJCSI), vol. 9, no. 1, pp. 237–242, 2012.