

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Penggunaan *smartphone* dalam kehidupan sehari-hari tidak dapat dipisahkan dalam kehidupan manusia *modern* saat ini. Dengan *smartphone* manusia dapat saling terhubung satu sama lain di manapun dan kapanpun. Salah satu *system* operasi perangkat *mobile* yang populer digunakan adalah Android. Android telah menjadi perangkat *mobile* yang paling populer dengan 675.000 aplikasi yang aktif di *Google Play Store* dengan pertumbuhan 12.000 aplikasi setiap bulannya [11]. Namun ada masalah yang dihadapi oleh Android, yaitu penggunaan baterai yang berlebih oleh banyak aplikasi sehingga membuat baterai Android lebih cepat habis [2].

Beberapa penelitian sudah dilakukan untuk mengatasi masalah ini, baik di sisi sistem, aplikasi maupun bahasa pemrograman yang dipakai. Aspek yang diteliti meliputi GPS [8][12], *Profile Energy* [1], *Network* [13], *Modeling System* [10], *Wakelock Bug* [6], *Memory Usage* [2][3] dan *Permission usage* [4]. Beberapa penelitian seperti *Battery Saver* menemukan bahwa penggunaan energi yang paling besar disebabkan oleh aplikasi *background* [9]. Sebab, aplikasi yang berjalan di *background* tersebut, ketika tidak digunakan oleh *user*, dapat menyebabkan penggunaan daya energi baterai *smartphone* Android yang besar.

Secara umum, aplikasi-aplikasi penghemat daya baterai yang berjalan pada *smartphone* Android menggunakan semacam *task killer* untuk mematikan proses-proses *background* yang dianggap tidak perlu. Bahkan dari penelitian-penelitian yang sudah ada, seperti pada [1][2][3][4][6][8][9][10][12][13] juga menggunakan mekanisme semacam ini. Namun, secara umum, penelitian-penelitian tersebut tidak menyusun *list* aplikasi pada proses *background* yang sudah ada berdasarkan prioritas yang berbasis pada perilaku *user*, di mana aplikasi penghemat daya baterai dapat menentukan aplikasi mana yang perlu ditutup. Kebanyakan mekanisme yang dipilih adalah dengan mematikan semua aplikasi *background* secara menyeluruh tanpa ada prioritas. Selain itu, pada penelitian-penelitian tersebut juga belum

menggunakan *user preference* untuk mematikan aplikasi *background* yang dianggap tidak perlu.

Kelemahan-kelemahan ini menunjukkan bahwa perlu adanya suatu penelitian lebih lanjut mengenai pengembangan aplikasi penghemat daya baterai pada *smartphone* Android dengan berbasis pada perilaku *user*, serta penggunaan *user preference* agar dapat melakukan terminasi pada proses-proses *background* tanpa mengganggu *user*, akibat matinya proses yang sesungguhnya tidak diinginkan *user*.

1.2 Identifikasi Masalah

Penggunaan baterai berlebih oleh banyak aplikasi-aplikasi yang mengambil sumber daya CPU, sehingga membuat baterai Android lebih cepat habis.

1.3 Rumusan Masalah

- 1) Bagaimana membuat aplikasi yang dapat menghemat daya baterai dengan mengatur konsumsi pemakaian CPU dari aplikasi-aplikasi *background* yang sedang berjalan.
- 2) Desain aplikasi seperti apa yang tepat untuk mengontrol *process* aplikasi-aplikasi di perangkat Android.

1.4 Batasan Masalah

- 1) Aplikasi berjalan pada perangkat sistem Android yang telah di-*root*.
- 2) *Memory process* pada tiap aplikasi hanya sebagai informasi. Tidak digunakan dalam proses *sorting*.
- 3) Data aplikasi yaitu berupa variabel-variabel disimpan pada Android *Local Storage*.

1.5 Tujuan

- 1) Membuat aplikasi yang dapat menghemat daya baterai dengan mengatur aplikasi-aplikasi yang sedang berjalan.

- 2) Mendesain aplikasi yang tepat untuk mengontrol proses-proses aplikasi-aplikasi di *smartphone* Android dan menerapkan parameter *user preference* kedalam aplikasi.

1.6 Metodologi

Adapun metodologi yang dipakai untuk pembuatan aplikasi *battery saver* yaitu studi literatur dan menerapkan *waterfall design* dalam perancangan aplikasi, adapun penjelasan mengenai metode pengerjaannya sebagai berikut :

1.6.1 Studi Literatur

Peneliti membaca beberapa literatur yang ada hubungannya dengan tugas akhir, setelah membaca beberapa referensi-referensi literatur, lalu sebagai perbandingan terhadap aplikasi yang akan peneliti buat.

1.6.2 Waterfall Design

Waterfall atau air terjun adalah model yang diperkenalkan oleh Dr. Winston W. Royce pada tahun 1970. Model ini menuntun *developer* untuk melakukan pengembangan *software* yang akan dibangun secara sistematis dan mempunyai keterkaitan antara prosesnya. Dalam arti tahapan proses berikutnya tidak akan dapat diwujudkan jika tahapan sebelumnya tidak terselasaikan. ada dua langkah penting yang harus ada dalam pembuatan *software computer*. pertama ialah *analysis* lalu diikuti oleh *coding*, jika di *breakdown* akan menjadi beberapa tahap yang spesifik yaitu *System Requirements -> Software Requirements -> Analysis -> Program Design -> Coding -> Testing* [5]