

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Pada zaman sekarang ini pengharum ruangan sudah menjadi kebutuhan bagi sebagian besar manusia, dikarenakan pengharum ruangan dapat membuat udara menjadi lebih segar dalam sebuah ruangan seperti dirumah, tempat kerja dan tempat lainnya yang sering dipakai untuk aktivitas manusia. Dengan adanya pengharum ruangan manusia bisa merasakan kenyamanan dan kesegaran udara ketika berada dalam suatu ruangan.

Pengharum Ruangan adalah produk konsumen yang biasanya memancarkan aroma dan digunakan di rumah atau interior komersial seperti toilet, foyer, lorong, ruang depan dan ruang dalam ruangan yang lebih kecil lainnya, serta area yang lebih besar seperti lobi hotel, dealer mobil, fasilitas medis, arena publik dan ruang interior besar lainnya. Beberapa jenis pengharum ruangan yang berbeda termasuk penyegar udara kipas angin listrik, sistem pembersihan kontrol bau tak sedap gravitasi, penyegar aroma penguapan non-mekanis pasif, dispenser kabut yang dioperasikan aerosol yang dioperasikan dengan waktu, semprotan, lilin, minyak, gel, manik-manik, dan colokan -di dalam. Beberapa penyegar udara mengandung bahan kimia yang memicu gejala alergi dan asma atau beracun. Penyegar udara tidak hanya terbatas pada semprotan modern, penyegar udara juga dapat melibatkan penggunaan barang-barang rumah tangga organik dan sehari-hari. Meskipun pengharum ruangan digunakan untuk menghilangkan bau, beberapa orang menggunakan pengharum ruangan untuk aroma menyenangkan yang mereka keluarkan.

Pada pengharum ruangan saat ini belum tersinkronisasi dengan android yang dapat melakukan pengontrolan pengharum ruangan dengan mudah seperti mendeteksi seberapa banyak sisa cairan yang ada pada pengharum ruangan tersebut.

Maka dari itu dibutuhkan sebuah sistem yang dapat memberikan informasi tentang jumlah cairan yang tersisa, prediksi kapan cairan pengharum ruangan akan habis dan riwayat pembaruan data pada pengharum ruangan. Serta menampilkan notifikasi jika cairan pengharum akan habis. Sehingga diusulkan sebuah sistem yang dapat menerapkan kebutuhan-kebutuhan tersebut dalam sebuah aplikasi berbasis IoT yaitu Smart Freshener.

## **1.2 Perumusan Masalah**

Berdasarkan latar belakang yang telah dipaparkan, maka rumusan masalah yang akan dibahas adalah:

1. Bagaimana cara menampilkan informasi jumlah cairan yang tersisa terkait pengharum ruangan ke dalam perangkat android?
2. Bagaimana cara menampilkan informasi terkait prediksi kapan cairan pengharum ruangan akan habis?
3. Bagaimana cara agar pengguna dapat mengetahui kapan data pengharum ruangan terakhir diperbarui?

## **1.3 Batasan Masalah**

Batasan masalah dari pembangunan sistem ini adalah:

1. Pengguna hanya dapat melakukan monitoring saja terhadap pengharum ruangan.
2. Sensor yang digunakan untuk mendeteksi sisa cairan yaitu sensor water level.
3. Sistem yang dibangun menggunakan firebase sebagai penyimpanan data sehingga pengguna harus selalu mempunyai jaringan internet untuk menggunakan sistem ini.

## **1.4 Tujuan**

Tujuan dari pembangunan sistem ini adalah:

1. Membantu pengguna dalam memeriksa jumlah cairan yang tersisa pada pengharum ruangan.
2. Mengetahui kapan cairan pada pengharum ruangan akan habis dengan menampilkan jumlah hari yang tersisa.
3. Membangun sistem yang dapat menampilkan waktu terkait pembaruan data yang ada pada pengharum ruangan dengan menampilkan detail tanggal beserta jamnya.

## 1.5 Metodologi Penyelesaian Masalah

Proses pembangunan aplikasi menggunakan sebuah metode penelitian yaitu metode pengumpulan data waterfall.

1. Metode waterfall, merupakan metode yang sering digunakan oleh penganalisa sistem pada umumnya. Inti dari metode waterfall adalah pengerjaan dari suatu sistem dilakukan secara berurutan atau secara linear. Jadi jika langkah ke-1 belum dikerjakan, maka langkah 2 tidak dapat dikerjakan. Jika langkah ke-2 belum dikerjakan maka langkah ke-3 juga tidak dapat dikerjakan, begitu seterusnya. Secara otomatis langkah ke-3 akan bisa dilakukan jika langkah ke-1 dan ke-2 sudah dilakukan. Secara garis besar metode waterfall mempunyai langkah-langkah sebagai berikut: Analisa, Desain, Penulisan, Pengujian dan Penerapan serta Pemeliharaan.

Tahapan pada metode waterfall antara lain:

- (a) Analisa kebutuhan, Merupakan langkah dalam melakukan analisa terhadap kebutuhan sistem. Dalam mengumpulkan data, pada tahap ini bisa melakukannya melalui wawancara, observasi atau studi literatur. Informasi sangat penting karena menyangkut dalam pembuatan dokumen user requirement.
- (b) Desain sistem, Pada tahap ini dilakukan pemodelan sistem dari gagasan yang sebelumnya telah dibuat, dalam diagram alir data (data flow diagram), diagram hubungan entitas (entity relationship diagram).
- (c) Penulisan data program Penulisan kode program adalah tahapan dalam mengaplikasikan bahasa pemrograman yang diterjemahkan oleh programmer. Tahap ini menjadi perwujudan nyata kebutuhan kedalam bahasa pemrograman yang menjadi sebuah program.
- (d) Pengujian program, Pengujian program adalah tahapan dalam menjalankan program, tahap ini menjadi tes kelayakan program. Program harus berfungsi sesuai dengan kebutuhan, jika terjadi kesalahan maka dapat diperbaiki sebelum program diberikan kepada user.
- (e) Penerapan dan pemeliharaan, Dalam tahap ini program diberikan kepada user, yang sebelumnya dilakukan pengujian program dan dilakukan perbaikan jika terjadi kesalahan pada program. Dan jika terjadi kendala seperti bug, force close dan sebagainya. Maka akan dilakukan perbaikan karena menjadi tanggung jawab dalam penerapan dan pemeliharaan program. [1]

## 1.6 Pembagian Tugas Anggota

Berikut pembagian tugas anggota tim proyek:

### 1. Fauzi Rizali Akbar

Peran : *Programmer*

**Tanggung Jawab** :

- (a) Membuat desain algoritma
- (b) Merancang program aplikasi
- (c) Analisis bug
- (d) Perbaiki bug
- (e) Pembuatan video promo
- (f) testing

### 2. Muhammad Juldhan Naufal

Peran : Desain dan Sistem Analisis

**Tanggung Jawab** :

- (a) Menganalisa masalah
- (b) Mencari referensi
- (c) Membuat *interface* program aplikasi
- (d) Mengembangkan produk
- (e) Pembuatan video promo
- (f) testing