

## **BAB V**

### **KESIMPULAN DAN SARAN**

Pada bab ini kesimpulan dari hasil penelitian dengan judul “Pengembangan Platform IoT Cerdas Dengan *Trigger Action Programming* dan Canvas API Berbasis Android”. Pembahasan dari bab ini yaitu kesimpulan dari setiap rumusan masalah dan saran untuk penelitian selanjutnya. Pada kesimpulan dibuat memiliki jumlah yang sama dengan rumusan masalah, pada saran diberikan semua masukan yang dapat dilanjutkan pada penelitian selanjutnya.

#### **5.1. Kesimpulan**

Berdasarkan uraian penelitian yang telah penulis paparkan pada bab-bab sebelumnya, dapat diketahui hasil dari penelitian ini pada kesimpulan. Pada kesimpulan dijawab pertanyaan yang ada pada rumusan masalah. Berdasarkan paparan yang dilakukan dapat ditarik kesimpulan sebagai berikut:

1. Perancangan platform IoT cerdas berbasis Android dapat dilakukan dengan memanfaatkan diagram *use case*, *use case scenario*, diagram activity, diagram *sequence*, dan diagram *class*. Pada tahap implementasi digunakan bahasa pemrograman Kotlin dan XML untuk membangun tampilan dari aplikasi Android pada implementasi aplikasi Android dengan Kotlin dan XML digunakan arsitektur MVVM untuk membuat struktur aplikasi yang mudah dipahami dan handal.
2. Integrasi model *machine learning* pada platform ini dilakukan dengan melakukan *deployment* dari tiap model yang telah dimasukkan oleh pengguna. Pengguna memasukkan model dengan format H5 atau SAV untuk dilakukan *deployment* pada server. Platform Android sebagai antar muka pengguna untuk menghubungkan komponen IoT dengan model *machine learning* milik pengguna dengan menampilkan semua model yang ada pada basis data model yang telah dimasukkan pengguna, model yang telah dimasukkan tadi akan ditampilkan dalam bentuk *dropdown* dari nama model yang dapat dipilih pada detail komponen IoT pada *dashboard* IoT.

3. Pada platform ini untuk membuat sistem dapat menangani banyak model yaitu dengan memanfaatkan *saved model* dari model *machine learning* yang terbatas pada model dengan format h5 dan sav. Memuat data model dari h5 digunakan Tensorflow dan untuk sav digunakan Pickle. Pada tahap akhir model tadi disimpan dan dibuat sebuah *endpoint* dengan FastAPI Python untuk *deploy* model tadi sehingga bisa diakses pada Android. *Endpoint* yang dibuat khusus untuk melakukan *deployment* pada model ini adalah */models/predict* dengan masukan berupa *query string* id dari model dan input yang berformat JSON.
4. Implementasi *trigger action programming* dilakukan pada komponen IoT *switch* dengan memanfaatkan topik untuk mengidentifikasi masukan yang digunakan untuk mengontrol komponen *switch*. Pada implementasinya *switch* diberikan kolom khusus pada *bottom sheet* detail komponen di mana terdapat 5 masukan yaitu *value* yang dapat diisi dengan nama topik yang akan dibandingkan, operator sebagai operator dari perbandingan tadi, *condition* di mana digunakan sebagai pembanding lalu kolom *true* dan *false* untuk masukan dari nilai yang akan di *push* ke topik yang telah diikuti oleh *switch* ketika perbandingan tadi bernilai benar atau salah.

## 5.2. Saran

Kendati dapat digunakan dengan cukup baik, namun masih dapat dikembangkan lebih lanjut karena masih memiliki beberapa kekurangan. Berikut adalah beberapa kekurangan tersebut:

1. Ke depannya dapat ditambahkan fitur untuk validasi *input* dan *output* dari model yang akan dihubungkan ke komponen IoT.
2. Pemanfaatan *bottom sheet* sebagai detail komponen terlalu kecil sebaiknya dimasukkan pada *fragment / activity* tersendiri.
3. Interaktivitas pada komponen dapat ditambahkan seperti memperbesar, memperkecil dan merubah rotasi dari komponen tadi.