



## Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

# **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## **BAB V**

#### SIMPULAN DAN SARAN

## 5.1 Simpulan

Dari hasil penelitian aplikasi keyboard pemblokir kata pornografi pada android dengan algoritma aho-corasick berhasil dirancang dan dibangun dengan menggunakan Bahasa pemrograman Java dan IDE Android Studio. Berdasarkan data kusioner yang disusun berdasarkan SUS, hasil perhitungan data yang didapat sebesar 82.74194 dan didapat standar deviasi sebesar 12.01 dari 100 yang berarti penyebaran data tiap responden dikatakan bagus. Dari hasil SUS skor tersebut juga didapat penjabaran bahwa *Acceptability ranges* dengan hasil *acceptable* menunjukan bahwa aplikasi NFKeyboard dapat diterima, *grade scale* dengan hasil B yang menunjukan bahwa aplikasi bernilai baik, dan *adjective ratings* dengan hasil *excellent*.

Hasil *percentile rank* yang didapatkan adalah nilai A. Hasil tersebut menunjukan bahwa aplikasi NFKeyboard mempunyai *perceived usability* yang 90% lebih tinggi dari produk yang pernah diuji. Berdasarkan data pencatatan pengukuran kecepatan pencarian kata berdasarkan urutan *index* dan panjang kata pada 2 *smartphone* yang berbeda spesifikasi, didapatkan kesimpulan bahwa panjang kata dan urutan kata tidak memengaruhi hasil kecepatan dari proses pencarian kata.

## 5.2 Saran

Penelitian ini jauh dari kata sempurna, namun berdasarkan kesimpulan di atas, saran yang dapat diberikan untuk pengembangan aplikasi selanjutnya adalah sebagai berikut.

- 1. Dilakukan pemblokiran *exact matching/partial/sub*. Sehingga pemblokiran tidak terpaku pada input *user*. Serta memakai algoritma yang bisa mendeteksi salah ketik pada saat *user* mengetik. Jadi jika ada salah ketik, walaupun ada *keyword* yang bermakna sama bisa di-*block*. Contoh salah satu algoritma terdsebut adalah Levenshtein Edit Distance.
- Dibuat gamifikasi sistem sehingga ada fitur *login* dan *notify* yang bisa menyambungkan antara *user* agar bisa dikirim pemberitahuan jika *user* mengetik kata yang di-*block*. Serta bisa mengetahui *memory* yang dipakai oleh aplikasi.