

BAB IV KESIMPULAN DAN SARAN

4.1 Kesimpulan

Dari data yang telah didapatkan dari hasil analisis resiko yang kepada pembuatan APKD, berikut merupakan kesimpulan yang dapat diambil :

Tahap *communication / requirement*, telah dijalankan dengan sangat baik dalam proses pembuatan APKD, memang benar terdapat gap dengan teori yang ada, tetapi tidak perlu dilakukan perubahan karena tidak mempengaruhi kinerja dan hasil akhir.

Tahap *Data modeling*, dilakukan dengan catatan dan perlu perbaikan saat pembuatan selanjutnya. Karena data modeling yang tujuan akhirnya adalah mendapatkan rancangan basis data.

Tahap *Quick Plan / Analisis*, telah dijalankan dengan sangat baik dalam pembuatan APKD, dengan adanya penambahan *flow* model, maka pengertian bagaimana sistem harus berjalan akan lebih baik, mengingat *flow* model bukan bagian resmi dari UML.

Tahap *Architecture*, kurang baik dijalankan, hal ini karena proses analisis menggunakan metode penggambaran UML yang pada umumnya digunakan untuk pemrograman OO (*Object Oriented*), sedangkan saat proses pengkodean menggunakan pemrograman terstruktur sehingga beberapa langkah dalam tahap *architecture* tidak mungkin dilakukan. Langkah penting lain yang tidak dilakukan adalah testing sebelum dikodekan.

Tahap *Implementasi*, dilakukan dengan baik karena tujuan akhir tahap ini adalah *source code* dan basis data, serta terdapat dokumentasi untuk *source code* yang ada.

Tahap testing, dilakukan dengan sangat baik dalam pembuatan APKD.

4.2 Saran

Berikut merupakan saran – saran yang direkomendasikan untuk menghilangkan “gap” dalam pembuatan aplikasi yang dikembangkan oleh tim APKD :

Tahap *Communication / Requirement*, langkah mendapatkan kesepakatan semua *stakeholder* lebih baik jika dilakukan untuk mendapatkan pemahaman kebutuhan tiap – tiap *stakeholder* dengan baik, sehingga aplikasi yang dikembangkan tepat guna.

Tahap *Data Modeling*, langkah pembuatan ERD merupakan salah satu langkah penting dalam tahap data modeling, hal ini perlu dilakukan dalam pembuatan aplikasi agar data yang dibutuhkan dapat dijabarkan secara jelas.

Tahap *Quick Plan / Analisis*, langkah DFD lebih baik dilakukan agar tim pengembang lebih memahami alur sistem yang akan dibuat.

Tahap *Architecture*, langkah *reuse design component* lebih baik dilakukan saat melakukan desain menggunakan UML karena UML pada umumnya digunakan untuk pemrograman *object oriented*. Langkah testing sebelum dikodekan lebih baik dilakukan untuk meminimalisir kesalahan yang ada.

Tahap Implementasi, tidak ada saran dalam tahap ini karena langkah yang ada, dijalankan dengan baik

Tahap testing, tidak ada saran dalam tahap ini karena langkah yang ada dijalankan dengan baik