

BAB 6: KESIMPULAN DAN SARAN

Bab ini akan membahas kesimpulan dan evaluasi dari hasil pengembangan aplikasi beserta saran – saran dan kemungkinan perbaikan atau pengembangan aplikasi di masa yang akan datang.

6.1. Keterkaitan Kesimpulan Dengan Hasil Evaluasi

Tujuan dari pengembangan aplikasi ini sudah tercapai walaupun belum mencapai 100%, terutama untuk masalah *link HTML* yang tidak standar (misalnya link dari *javascript*). Tetapi hasil test case sudah menunjukkan bahwa proses offline browsing telah dapat dilakukan pada file – file hasil crawling.

6.2. Keterkaitan Saran Dengan Hasil Evaluasi

Dari hasil kesimpulan di atas, saran yang diajukan adalah perbaikan pada fungsi ekstraksi *link*, sehingga dapat mengenali *link* yang kompleks sekalipun.

6.3. Saran Pengembangan

Langkah – langkah yang dapat diambil untuk pengembangan aplikasi ini lebih lanjut, antara lain:

- Memperbaiki kemampuan ekstraksi link dari link processor.
- Penambahan fitur untuk mengenali lebih banyak tag link, tidak hanya link HTML `<a href>` dan ``.
- Implementasi fitur tree dari web hasil crawling yang belum rampung.

Sedangkan hal – hal lainnya yang mungkin untuk dikembangkan atau diimplementasikan pada aplikasi ini di masa yang akan datang, antara lain:

- Penerapan algoritma indexing atau information retrieval untuk memungkinkan crawler melakukan crawling pada halaman web berdasarkan query tertentu yang dimasukkan oleh user,
- Dukungan untuk link depth yang tidak terbatas, sehingga memungkinkan melakukan retrieval file dari web yang sangat besar sekalipun,
- Kemampuan threading yang lebih canggih (optimalisasi algoritma threading yang ada).

Saat ini model *multithreading* aplikasi menggunakan sistem antrian / *queue* yang efektifitasnya masih kalah dibandingkan model lainnya, seperti *thread pooling*.

Pada model *thread queue* yang digunakan *slot thread* yang sudah *finished* tidak akan langsung terpakai karena queue thread tidak akan di *flush* (membuang *thread* yang sudah selesai atau *thread* yang *error*) jika belum ada *flag FULL* dari *CapacityStatus*, sementara pada model *pooling* setiap *slot thread* yang *finished* akan langsung digantikan oleh *worker thread* baru.