

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pada penelitian ini telah berhasil dibangun suatu algoritma *switching* antara Lanczos *type* A_{12} , A_{13}/B_6 dan A_{13}/B_{10} . Keterbaharuan algoritma *switching* pada penelitian ini dibanding penelitian sebelumnya adalah ketika breakdown terjadi pada iterasi ke k , maka ganti algoritma Lanczos *type* yang digunakan dengan inisialisasinya diambil dari iterasi ke $k-1$ pada algoritma Lanczos *type* yang sebelumnya.

Dari hasil simulasi menggunakan matriks koefisien berdimensi 100 hingga 10000, terlihat bahwa strategi *switching* sangat merupakan metode yang baik digunakan untuk mengatasi permasalahan *breakdown* dalam algoritma Lanczos *type*. Seluruh permasalahan dalam berbagai dimensi berhasil dicari solusinya menggunakan program yang telah dibangun. Nilai $\|\mathbf{r}_k\|$ terkecil didapatkan oleh SPL berdimensi 500 yaitu $2,90 \times 10^{-5}$ dan nilai $\|\mathbf{r}_k\|$ terbesar didapatkan oleh SPL berdimensi 7000 yaitu $9,75 \times 10^{-5}$. Sedangkan total *cycle* terkecil dihasilkan oleh SPL berdimensi 1500, 2500, 8000 dan 9500 yaitu 3 serta total *cycle* terbesar dihasilkan oleh dimensi 500 yaitu 14. Untuk waktu komputasi program, didapatkan bahwa semakin besar dimensi SPL dan total *cycle* yang dijalankan maka waktu komputasi juga cenderung semakin besar. Waktu komputasi tercepat adalah 0,113 dan waktu komputasi terlama adalah 151,117 detik. Selain itu, perkembangan nilai $\|\mathbf{r}_k\|$ dalam setiap iterasi tidak selalu turun, melainkan cenderung dinamis dengan nilai yang tak menentu.

5.2 Saran

Pada penelitian ini, strategi *switching* dilakukan antara algoritma Lanczos *type* A_{12} , A_{13}/B_6 dan A_{13}/B_{10} . Namun algoritma A_{13}/B_{10} dianggap kurang bagus karena cenderung menghasilkan nilai $\|\mathbf{r}_k\|$ yang lebih besar dan iterasi yang lebih

sedikit sebelum terjadinya *breakdown*. Untuk penelitian selanjutnya, diharapkan bahwa strategi *switching* dapat digunakan pada algoritma Lanczos *type* yang lebih *robust* sehingga menghasilkan program penyelesaian SPL dengan nilai $\|\mathbf{r}_k\|$ yang lebih kecil dan waktu komputasi yang lebih efisien. Selain itu, aturan yang digunakan dalam melakukan *switching* dapat diganti dengan aturan yang lebih baik, seperti inisialisasi untuk Lanczos *type* selanjutnya diambil dari iterasi yang memiliki nilai $\|\mathbf{r}_k\|$ terkecil maupun aturan lainnya.

