

BAB 6

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Penelitian ini dilakukan dengan tujuan untuk membangun perangkat lunak yang mampu membangkitkan permainan teka-teki silang secara otomatis. Teknik yang digunakan dalam membangkitkan papan teka-teki silang yaitu dengan pencarian lokal *hill climbing random-restart*. Penggunaan teori pencarian seperti *exhaustive search* dan *particle swarm optimization* menemukan beberapa kesulitan pada saat dilakukan pemodelan masalah dan pembangkitan papan permainan oleh perangkat lunak. Kedua teori tersebut dinilai mampu menemukan *global optimum* atau solusi terbaik dari suatu kasus uji. Berbeda dengan *hill climbing random-restart* yang tidak akan menemukan *global optimum*. Tetapi dilihat dari hasil pengujian fungsional, papan teka-teki silang dengan ukuran 6x6, 20x20, dan 26x26 sudah menyerupai permainan teka-teki silang pada umumnya. Sehingga dengan mencapai *local maxima* pada hasil akhir pembangkitkan, *output* perangkat lunak ini sudah dinilai memuaskan.

Dalam menggunakan perangkat lunak, pengguna menyediakan sekumpulan kata yang merupakan petunjuk dan sekumpulan kata sebagai jawaban dari petunjuk tersebut. Kata-kata tersebut disimpan pada file terpisah dengan format penamaan file jawaban.txt dan pertanyaan.txt. Lalu pengguna diminta oleh perangkat lunak untuk memasukkan bilangan yang menjadi masukkan perangkat lunak sebagai ukuran dari teka-teki silang yang ingin dibangkitkan. Perangkat lunak telah diuji (dengan teknik *blackbox*) untuk mampu menghasilkan permainan teka-teki silang, dengan hasil keluaran yang dapat dilihat pada terminal IDE atau pada file dengan format hasilakhir.txt.

Pengujian dilakukan dengan teknik *blackbox* yang terbagi pada beberapa tipe pengujian. Pengujian dilakukan untuk mendapatkan kepastian bahwa perangkat lunak yang dibuat sudah mempunyai kebenaran secara fungsional dan nonfungsional. Selain dilakukan pengujian terhadap kebenaran atas perangkat lunak yang telah dibangun, dilakukan pula beberapa percobaan atau eksperimen terhadap perangkat lunak. Eksperimen yang dilakukan bertujuan untuk mendapatkan nilai yang tepat pada beberapa parameter algoritma pembangkitan. Kesimpulan dari beberapa eksperimen yang sudah dilakukan adalah:

- Nilai yang baik digunakan pada parameter jumlah kata adalah 50 sampai 70. Dengan nilai fungsi *fitness output* perangkat lunak berkisar 1,5 sampai 1,7 membutuhkan waktu komputasi 8 detik sampai 12 detik.
- Nilai yang baik digunakan pada parameter jumlah titik *start* algoritma *hill climbing random-restart* adalah berkisar 100 sampai 200 dengan waktu komputasi di bawah 20 detik untuk mendapatkan hasil *output* dengan fungsi *fitness* 1,2 sampai 1,3.
- Nilai yang baik digunakan pada parameter jumlah iterasi proses pembangkitan adalah 80 sampai 100 dengan waktu komputasi di bawah 10 detik untuk mendapatkan hasil *output* dengan nilai fungsi *fitness* 1,3 sampai 1,4.
- Nilai yang baik digunakan pada parameter ukuran papan teka-teki silang adalah antara 20

sampai 22. Tetapi angka tersebut tidak menjadi suatu patokan bagi pengguna, karena parameter ukuran papan tersebut adalah *input* bagi perangkat lunak serta pengguna menyesuaikan kebutuhannya dalam membangkitkan teka-teki silang dengan ukuran tertentu.

Penilaian terhadap baik tidaknya suatu teka-teki silang adalah penggunaan petak yang efisien. Penggunaan petak yang efisien dapat diamati pada perbandingan jumlah petak terisi dan jumlah petak kosong. Semakin banyak jumlah petak yang terisi oleh huruf maka semakin baik papan teka-teki silang tersebut. Pada penelitian ini, penilaian atas baik atau tidaknya suatu papan teka-teki silang dinyatakan sebagai fungsi *fitness*.

Petak yang menjadi lokasi perpotongan dinilai mempunyai bobot lebih besar dibandingkan petak tanpa perpotongan kata. Karena tempat perpotongan ini menyimpan 2 huruf dengan nilai String yang sama. Hasil akhir perangkat lunak yang telah dibangun tidak akan ditemukan kata yang diletakkan tidak berpotongan dengan kata lainnya. Karena pada proses pembangkitan, diprioritaskan kata yang diletakkan secara berpotongan daripada yang tidak berpotongan. Karena perpotongan ini yang dinilai mampu meningkatkan fungsi *fitness* papan.

6.2 Saran

Perangkat lunak yang dibangun masih memiliki beberapa kekurangan yaitu pada sisi tampilan. Hasil keluaran yang disimpan pada file hasilakhir.txt memiliki 4 bagian utama yaitu nilai fungsi *fitness* papan, lokasi petunjuk dilengkapi dengan kode alfanumerik dan kata atau frasa yang digunakan sebagai soal teka-teki silang, lokasi jawaban dilengkapi dengan kode alfa numerik dan kata atau frasa yang digunakan sebagai kunci jawaban menyelesaikan teka-teki silang, tampilan papan teka-teki silang dengan jawaban, dan tampilan teka-teki silang tanpa jawaban.

Seluruh hasil keluaran perangkat masih dalam format teks, untuk pengembangan perangkat lunak selanjutnya, dapat digunakan format tampilan non-teks yang dapat menampilkan secara grafik-visualisasi permainan teka-teki silang kepada pengguna. Sehingga hasil dari grafik-visualisasi tersebut, pengguna dapat mencetak papan permainan teka-teki silang untuk diselesaikan.

Penelitian dapat dilanjutkan pada pembuktian teoritis mengenai pencarian lokal *particle swarm optimization* dapat digunakan sebagai pendekatan alternatif dalam membangkitkan permainan teka-teki silang. Dilakukan pula perbandingan kinerja algoritma pembangkitan yang dihasilkan oleh pendekatan *hill climbing random-restart* dan yang dihasilkan oleh pendekatan PSO tersebut.

DAFTAR REFERENSI

- [1] Norvig, P. dan Russell, S. J. (1994) Constraint satisfaction problem. Bagian dari Peter Norvig, S. J. R. (ed.), *Artificial Intelligence: A Modern Approach*. Prentice Hall, USA.
- [2] Cheng, D. dan Dhulekar, N. (2009) Crossword puzzle generator. Technical Report UU-CS-2009-044. Rensselaer Polytechnic, USA.
- [3] Elum, C. R. (1996) *Crossword Puzzle Game and Method of Generating the Same*. Paten no. 5566942. USPTO. Ohio, USA.
- [4] Levitin, A. V. (2002) *Introduction to the Design and Analysis of Algorithms*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- [5] Russell, S. dan Norvig, P. (1994) Solving problems by searching. Bagian dari Hirsch, M. (ed.), *Artificial Intelligence A Modern Approach*. Prentice Hall, USA.
- [6] Apter, M. J. (1970) *The Computer Simulation of Behaviour*. Hutchinson University Library, Michigan, USA.
- [7] Eberhart, R. dan Kennedy, J. (2004) A new optimizer using particle swarm theory. *P Signals Systems and Computers 2004. Conference Record of the Thirty-Eighth Asilomar Conference*, Pacific Grove, CA, USA, 7-10 November, pp. 241–245. IEEE.
- [8] Engelbrecht, A. (2007) Particle swarm optimization. Bagian dari Wiley, J. dan Inc, S. (ed.), *Computational Intelligence*. John Wiley and Sons Inc, USA.
- [9] Berghel, H. dan Yi, C. (1989) Crossword compiler- compilation. *The Computer Journal*, **32**, 277–280.