

MEMBANGUN APLIKASI GAME “DUA SATU (21)” BERBASIS ANDROID MENGGUNAKAN METODE *DEPTH FIRST SEARCH*

Muhammad Fajar M ^{*1}, Ika Purwanti Ningrum², Rahmat Ramadhan³

^{1,2,3}Jurusan Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo, Kendari

e-mail: ^{*1}fajarm80@gmail.com, ²ika.purwanti.n@gmail.com, ³rahmat.ramadhan@innov-center.org

Abstrak

Kata *game* berasal dari bahasa Inggris yang berarti permainan. Saat ini *game* berkembang di aplikasi *mobile*. Salah satu sistem operasi untuk perangkat *mobile* adalah Android. Perkembangan sistem operasi *mobile* Android diakibatkan oleh kelebihan yang bersifat terbuka, sehingga kemudahan dalam menggunakannya menjadi nilai tambah. Hal ini menyebabkan pemakaian *mobile* Android terus meningkat. Untuk pencarian solusi dalam sebuah *game* dapat menggunakan metode pencarian, salah satunya adalah algoritma *Depth-First Search (DFS)*.

Dalam metode pencarian *Depth First Search (DFS)*, pencarian dilakukan pada satu node dalam setiap level dari yang paling kiri. Jika pada level yang paling dalam solusi belum ditemukan, maka pencarian dilanjutkan pada node sebelah kanan. Pada penelitian ini, algoritma pencarian *Depth First Search (DFS)* diimplementasikan dalam sebuah *game*. *Game* yang diciptakan dalam penelitian ini adalah *game* Dua Satu (21).

Hasil dari penelitian ini menunjukkan bahwa algoritma pencarian *Depth First Search* dapat diterapkan dalam aplikasi *game* Dua Satu (21), namun tidak lengkap karena dari 144 kombinasi angka yang menjadi masukan, terdapat 30 kombinasi angka yang tidak muncul solusinya. Hal ini merupakan salah satu kekurangan dari algoritma pencarian *Depth First Search* yang memungkinkan untuk tidak ditemukannya solusi. Selain itu, hasil pengujian aplikasi ini ke beberapa *smartphone* juga menunjukkan kondisi yang sama, yaitu masih terdapat kombinasi angka yang tidak muncul solusinya, jadi spesifikasi dan merek *smartphone* tidak mempengaruhi hasil keluaran.

Kata Kunci : *Game, Mobile, Depth First Search (DFS), Android dan Dua Satu (21).*

Abstract

The word game came from English. Game is currently developing in mobile applications. One of the operating system for mobile devices is Android. The development of Android mobile operating system is caused by its excess, which is open access. Ease of use is also an added value that makes the using of Android mobile is more increase. The way to search the solutions in a game is by using search methods, one of it is the algorithm Depth-First Search (DFS).

In Depth First Search (DFS) method, a search is performed on one node in each level of the far left. If the lowest level in a solution is not found, then the search is continued on the right side node. In this study, the algorithm Depth First Search (DFS) is implemented in a game. Game that had been created in this study was Two One (21) game.

Results of this study was the algorithm Depth First Search can be applied in game applications Two One (21), but it did not complete because of from 144 combinations of numbers that become the inputs, there were 30 combinations of numbers that did not give any solutions. This was one of the deficiencies of algorithm Depth First Search which allows to not finding any solution. This application had been tested to multiple smartphone. It showed the same conditions that there were still some combinations of numbers that did not appear solutions. Therefore, the specifications and brand of smartphone did not affect output.

Keywords : *Game, Mobile, Depth First Search (DFS), Android and Two One (21).*

1. PENDAHULUAN

Game merupakan sesuatu yang tidak asing lagi bagi setiap orang, dari kalangan anak-anak maupun dewasa. Perkembangan teknologi saat ini sangat pesat, menuntut pemikiran-pemikiran yang berkembang untuk membuat *game* yang beragam, salah satunya adalah *game* tentang edukasi, seperti *game* asah otak, tes psikologi, ataupun perhitungan matematika. *Game* berasal dari bahasa Inggris yang artinya permainan. Permainan adalah usaha olah diri (olah pikiran dan olah fisik) yang sangat bermanfaat bagi peningkatan dan pengembangan motivasi, kinerja, dan prestasi dalam melaksanakan tugas dan kepentingan organisasi dengan lebih baik.

Game juga berkembang di aplikasi *mobile*. Sistem operasi untuk perangkat *mobile* saat ini salah satunya adalah Android. Perkembangan sistem operasi *mobile* Android diakibatkan oleh kelebihan Android tersebut, yaitu bersifat terbuka. Kemudahan dalam menggunakannya juga menjadi nilai tambah sehingga membuat pemakaian *mobile* Android saat ini terus meningkat. Penggunaan kapan saja dan di mana saja menjadi salah satu faktor perkembangan *game* di Android.

Android adalah sistem operasi yang bersifat *open source* (sumber terbuka). Disebut *open source* karena *source code* (kode sumber) dari sistem operasi Android dapat dilihat, diunduh, dan dimodifikasi secara bebas. Paradigma *open source* ini memudahkan pengembangan teknologi Android, karena semua pihak yang tertarik dapat berkontribusi baik pada pengembangan sistem operasi maupun aplikasi. Pada awal Oktober 2013, tercatat ada lebih 850.000 aplikasi Android yang tersedia di *Google Play*.

Dari 850.000 aplikasi tersebut, 700.000 diantaranya adalah aplikasi tak berbayar atau gratis, sedang sisanya adalah aplikasi berbayar. Kategori aplikasi yang paling populer adalah kategori hiburan (*Entertainment*) dan personalisasi dengan hampir 90.000 aplikasi di kedua kategori tersebut. Aplikasi *Games* diletakkan dalam kategori tersendiri yang dibagi berdasarkan genre seperti *Action*, *Puzzle* dan *Sports*, dengan lebih dari 10.000 aplikasi [1].

Untuk pencarian solusi dalam sebuah *game* dapat digunakan algoritma pencarian. Salah satunya adalah algoritma *Depth-First Search (DFS)*. Algoritma ini adalah algoritma pencarian pada sebuah pohon dengan menelusuri satu cabang sebuah pohon sampai menemukan solusi. Pencarian dilakukan pada satu node dalam setiap level dari yang paling kiri dan dilanjutkan pada node sebelah kanan. Pada algoritma DFS pemakaian memori tidak terlalu banyak dan jika solusi yang dicari berada pada level yang dalam paling kiri, maka DFS akan menemukannya secara cepat [2].

2. METODE PENELITIAN

2.1 Game

Game berasal dari bahasa Inggris yang berarti permainan. Dalam setiap *game* terdapat peraturan yang berbeda-beda untuk memulai permainannya sehingga membuat jenis *game* semakin bervariasi.

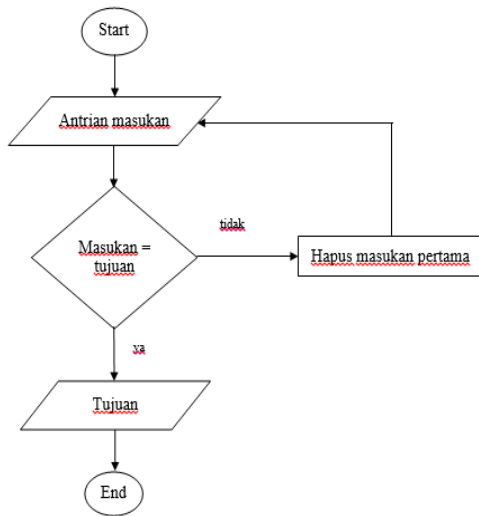
Game mempunyai dampak positif dan negatif pada kehidupan. Contoh dampak positif *game* adalah sebagai penghilang *stress* dari bekerja seharian sedangkan contoh dampak negatif *game* adalah dapat merusak mata jika terlalu lama bermain *game* di depan PC/Laptop [3].

2.2 Algoritma *Depth First Search (DFS)*

Pada algoritma DFS, pencarian dilakukan pada satu node dalam setiap level dari yang paling kiri. Jika pada level yang paling dalam solusi belum ditemukan, maka pencarian dilanjutkan pada node sebelah kanan. Node yang di kiri dapat dihapus dari memori. Jika pada level yang paling dalam belum ditemukan solusi, maka pencarian dilanjutkan ke level sebelumnya. Demikian seterusnya sampai ditemukannya solusi [4]. Gambar 1 menunjukkan *flowchart* algoritma DFS.

Algoritma pencarian *Depth First Search* diawali dengan membangkitkan antrian untuk menjadi masukan. Kemudian antrian pertama diproses dengan cara mencocokkan nilai masukan terhadap nilai tujuan. Jika iya, maka masukan pertama merupakan solusi dan proses berakhir. Jika tidak, maka antrian masukan pertama dihapus dan dimasukan antrian kedua. Proses

ini diulang sampai masukan sama dengan tujuan.



Gambar 1 Flowchart DFS

2.2 Analisis pemecahan masalah

a. Gambaran umum sistem

Penelitian ini dilakukan untuk membangun sebuah aplikasi dengan tema *game* berbasis mobile Android yang penyelesaian masalahnya menggunakan algoritma pencarian yaitu algoritma *Depth-First Search*.

Aplikasi *game* Dua Satu (21) merupakan *game* untuk *mobile* dengan sistem operasi Android. *Game* ini tidak memiliki jalan cerita, karena bukan *game* petualangan. Berdasarkan jenisnya, aplikasi *game* Dua Satu (21) termasuk *game educational*, karena pengguna dituntut untuk mengolah angka-angka yang disediakan dengan operasi dasar matematika sedemikian sehingga hasil akhirnya adalah 21.

Permainan ini dimulai dengan ditampilkannya 4 angka yang akan dimunculkan secara acak. Pemain dituntut untuk mengolah angka tersebut dengan memanfaatkan 6 operasi dasar matematika yang telah disediakan (tambah, kurang, kali, bagi, akar dan kuadrat). Hasil akhir dari pengolahan angka dalam permainan ini adalah 21. Gambar 2 menunjukkan tampilan menu utama *Game* Dua Satu (21).



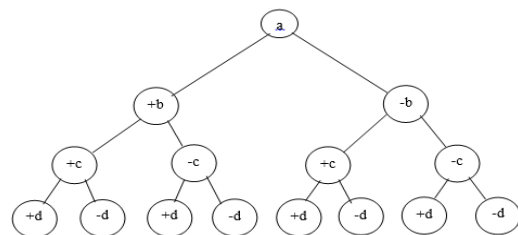
Gambar 2 Tampilan menu utama *Game* Dua Satu (21).

b. Penyelesaian menggunakan algoritma DFS

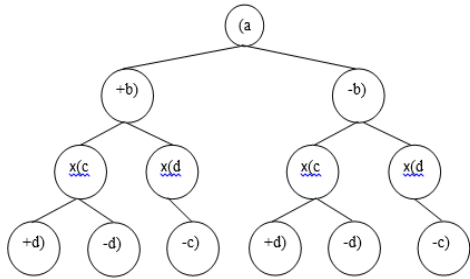
Untuk penggunaan algoritma DFS dalam aplikasi ini adalah dengan menentukan titik awal (*input*) dan tujuan (*goal*). Diketahui dalam aplikasi *game* Dua Satu ini mempunyai inputan berupa 4 buah kombinasi angka yang ditampilkan secara acak dan 6 operasi matematika yang telah disediakan dalam permainan. Untuk tujuan (*Goal*) aplikasi *game* Dua satu ini sesuai dengan nama aplikasinya adalah menghasilkan angka 21 di akhir pengolahan inputan.

Untuk mendapatkan solusi dari semua kemungkinan kombinasi angka yang muncul, dibangkitkan simpul dengan membuat pohon (*tree*) pencarian solusi. Dalam aplikasi ini membutuhkan banyak pohon penyelesaian dikarenakan masukan atau *inputan* dari 4 kombinasi angka yang akan muncul sangat banyak.

Berdasarkan 144 kombinasi angka yang menjadi masukan, perlu dibuat 110 simpul dari pohon pencarian. Salah satu pohonnya adalah seperti pada Gambar 3.



Gambar 3 Pohon pencarian (1)



Gambar 4 Pohon pencarian (2)

Gambar 4 menggunakan algoritma DFS menghasilkan solusi sebagai berikut:

1. Cara 1= $a + b + c + d$
2. Cara 2= $a + b + c - d$
3. Cara 3= $a + b - c + d$
4. Cara 4= $a + b - c - d$
5. Cara 5= $a - b + c + d$
6. Cara 6= $a - b + c - d$
7. Cara 7= $a - b - c + d$
8. Cara 8= $a - b - c - d$
9. Cara 9 = $(a + b) \times (c + d)$
10. Cara 10 = $(a + b) \times (c - d)$
11. Cara 11 = $(a + b) \times (d - c)$
12. Cara 12 = $(a - b) \times (c + d)$
13. Cara 13 = $(a - b) \times (c - d)$
14. Cara 14 = $(a - b) \times (d - c)$

Misalnya masukannya adalah 2,5,2,1, secara manual atau perhitungan biasa, kita bisa mendapatkan hasilnya dengan cara menambahkan angka pertama dan kedua kemudian dikalikan dengan hasil penjumlahan dari angka ketiga dan keempat. Namun, untuk algoritma DFS perlu dibangkitkan dulu simpul-simpul kemungkinan yang menjadi solusi. Seperti solusi yang dihasilkan pada gambar 2 dan 3 yaitu cara 1 sampai 14.

Untuk *inputan* 2,5,2,1 akan diuji dengan antrian simpul pohon solusi dari cara pertama sampai cara ke-14. Cara pertama, $a + b + c + d$ berarti $2 + 5 + 2 + 1$ hasilnya 10, tidak sama dengan tujuan (21) itu berarti cara pertama atau antrian pertama tidak memenuhi untuk menjadi solusi dalam aplikasi ini. Selanjutnya dimasukkan ke dalam cara kedua atau antrian kedua. Cara kedua, $a + b + c - d$ berarti $2 + 5 + 2 + 1$ hasilnya 9, tidak sama dengan tujuan. Begitu seterusnya cara ini dilakukan sampai mendapatkan solusi, yaitu pada cara kesembilan atau antrian kesembilan. Cara kesembilan $(a + b) \times (c + d)$ berarti $(2+5) \times (2+1)$

menghasilkan tujuan (21). Setelah itu berhentilah proses algoritma pencarian DFS ini.

3. HASIL DAN PEMBAHASAN

3.1 Skenario pertama

Pada penelitian ini, akan dilakukan pengujian sistem menggunakan beberapa perangkat keras yang berbeda, dalam hal ini adalah *mobile smartphone* Android dengan spesifikasi dan merek yang berbeda, yaitu pada Tabel 1.

Tabel 1 Spesifikasi dan merek *smartphone* yang digunakan

No		Samsung Galaxy Core Duos	Nokia XL Dual SIM	Asus T001
1	Ukuran Layar	4.3"	5.0"	4.0"
2	Resolusi	480 x 800 pixel	480 x 800 pixel	480 x 800 pixel
3	Sistem Operasi	Android v4.1.2 (Jelly Bean)	Android v4.1.2 (Jelly Bean)	Android v4.4.2 (Kit Kat)
4	CPU	Dual-Core 1.2 GHz	Dual-Core 1.2 GHz	Intel 1.2 GHz
5	Memori Internal	4 GB	4 GB	4 GB
6	RAM	1 GB	786 MB	1 GB

Dalam pengujian ini, dimasukkan 10 kombinasi angka untuk diproses dalam sistem pada setiap merek *smartphone* yang menjadi bahan uji. Hasil pegujian dapat dilihat pada Tabel 2, Tabel 3 dan Tabel 4.

Tabel 2 Hasil pengujian pada Samsung Galaxv Core Duos

No	Angka Masukkan	Perhitungan Sistem	Perhitungan Manual	Keterangan
1	2-5-6-1	$(5-2) \times (6+1)$	$(5-2) \times (6+1)$	Sesuai
2	7-2-4-1	$(1-2+4) \times 7$	$(1-2+4) \times 7$	Sesuai
3	7-2-2-3	$(3-2+2) \times 7$	$(3-2+2) \times 7$	Sesuai
4	1-2-6-5	$(1+6) \times (5-2)$	$(1+6) \times (5-2)$	Sesuai
5	1-4-6-1	$(4-1) \times (6+1)$	$(4-1) \times (6+1)$	Sesuai
6	7-4-6-1	$(1-4+6) \times 7$	$(1-4+6) \times 7$	Sesuai
7	5-4-2-3	$(5+4-2) \times 3$	$(5+4-2) \times 3$	Sesuai
8	1-2-6-3	$(6-1+2) \times 3$	$(6-1+2) \times 3$	Sesuai
9	1-2-4-5	Tidak muncul	$(4 \times 5) + (2-1)$	Tidak sesuai
10	5-5-6-1	Tidak muncul	$5^2 - ((6-1)+1)^2$	Tidak sesuai

Tabel 3 Hasil pengujian pada Nokia XL Dual SIM

No.	Angka Masukkan	Perhitungan Sistem	Perhitungan Manual	Keterangan
1	2-5-6-1	$(5-2) \times (6+1)$	$(5-2) \times (6+1)$	Sesuai
2	7-2-4-1	$(1-2+4) \times 7$	$(1-2+4) \times 7$	Sesuai
3	7-2-2-3	$(3-2+2) \times 7$	$(3-2+2) \times 7$	Sesuai
4	1-2-6-5	$(1+6) \times (5-2)$	$(1+6) \times (5-2)$	Sesuai
5	1-4-6-1	$(4-1) \times (6+1)$	$(4-1) \times (6+1)$	Sesuai
6	7-4-6-1	$(1-4+6) \times 7$	$(1-4+6) \times 7$	Sesuai
7	5-4-2-3	$(5+4-2) \times 3$	$(5+4-2) \times 3$	Sesuai
8	1-2-6-3	$(6-1+2) \times 3$	$(6-1+2) \times 3$	Sesuai
9	1-2-4-5	Tidak muncul	$(4 \times 5) + (2-1)$	Tidak sesuai
10	5-5-6-1	Tidak muncul	$5^2 - ((6-1)+1)^2$	Tidak sesuai

Tabel 4 Hasil pengujian pada Asus T001

No.	Angka Masukkan	Perhitungan Sistem	Perhitungan Manual	Keterangan
1	2-5-6-1	$(5-2) \times (6+1)$	$(5-2) \times (6+1)$	Sesuai
2	7-2-4-1	$(1-2+4) \times 7$	$(1-2+4) \times 7$	Sesuai
3	7-2-2-3	$(3-2+2) \times 7$	$(3-2+2) \times 7$	Sesuai
4	1-2-6-5	$(1+6) \times (5-2)$	$(1+6) \times (5-2)$	Sesuai
5	1-4-6-1	$(4-1) \times (6+1)$	$(4-1) \times (6+1)$	Sesuai
6	7-4-6-1	$(1-4+6) \times 7$	$(1-4+6) \times 7$	Sesuai
7	5-4-2-3	$(5+4-2) \times 3$	$(5+4-2) \times 3$	Sesuai
8	1-2-6-3	$(6-1+2) \times 3$	$(6-1+2) \times 3$	Sesuai
9	1-2-4-5	Tidak muncul	$(4 \times 5) + (2-1)$	Tidak sesuai
10	5-5-6-1	Tidak muncul	$5^2 - ((6-1)+1)^2$	Tidak sesuai

Hasil pengujian pada Samsung Galaxy Core Duos, Nokia XL Dual SIM, dan Asus T001, menunjukkan hasil yang sama bahwa ada 8 kombinasi angka masukkan yang sesuai antara perhitungan sistem dan perhitungan manual, dan terdapat 2 kombinasi angka masukkan yang tidak sesuai antara perhitungan sistem dan perhitungan manual.

Dari pengujian tersebut dapat disimpulkan bahwa spesifikasi dan merek

smartphone tidak mempengaruhi hasil keluaran.

3.2 Skenario kedua

Dalam pengujian scenario kedua ini dilakukan pengujian pada satu perangkat keras menggunakan masukkan berupa semua kombinasi angka yaitu sebanyak 144 kombinasi angka.

Hasil dari pengujian ini adalah dari 144 kombinasi angka yang menjadi masukkan, terdapat 30 kombinasi angka yang tidak muncul pemecahan masalahnya, yaitu: 1-2-2-1, 1-2-4-5, 1-3-2-1, 1-3-2-5, 1-3-4-3, 1-4-2-1, 1-4-4-5, 1-4-6-5, 1-5-4-5, 1-5-6-3, 2-2-2-1, 2-2-2-5, 2-2-6-1, 2-3-2-1, 2-3-2-3, 2-3-4-3, 2-4-6-3, 2-5-6-3, 5-2-4-1, 5-2-6-3, 5-3-4-5, 5-4-2-5, 5-4-4-1, 5-4-6-1, 5-5-6-1, 7-2-4-5, 7-3-2-1, 7-3-6-3, 7-4-6-3 dan 7-5-6-1.

Hal ini merupakan salah satu kekurangan dari Algoritma pencarian DFS, yaitu tidak *complete*, dimana algoritma DFS ini memungkinkan untuk tidak menemukan solusi jika pohon yang dibangkitkan mempunyai level yang dalam, seperti yang tertulis dalam buku yang berjudul *Artificial Intelligence* oleh Suyanto.

Jadi, tingkat keberhasilan penggunaan algoritma pencarian *Depth First Search* (DFS) dalam aplikasi *game* Dua Satu ini adalah:

$$\frac{144 - 30}{144} \times 100\% = 79.16\%$$

4. KESIMPULAN

Berdasarkan penelitian dan pengujian sistem yang dilakukan terhadap aplikasi *Game* Dua Satu (21), maka dapat disimpulkan:

1. Algoritma pencarian Depth First Search (DFS) dapat diterapkan dalam aplikasi *game* Dua Satu (21), yaitu dengan cara membangkitkan simpul-simpul solusi dari pembuatan pohon (*tree*) pencarian solusi.
2. Hasil pengujian aplikasi ini ke beberapa *smartphone* menunjukkan kondisi yang sama, yaitu masih terdapat kombinasi angka yang tidak muncul solusinya. Hal ini merupakan salah satu kekurangan dari algoritma DFS yaitu tidak *complete*,

- dimana sistem memungkinkan untuk tidak menemukan solusi permasalahannya.
3. Dari 144 kombinasi angka yang menjadi masukan, terdapat 30 kombinasi angka yang tidak muncul pemecahan masalahnya. Jadi, tingkat keberhasilan penggunaan algoritma pencarian *Depth First Search* (DFS) dalam aplikasi *game* Dua Satu ini adalah 79.16 %.

diakses 10 Juli 2015.

- [4] Kusumadewi, S., *Artificial Intelligence (Teknik dan Aplikasinya)*, Graha Ilmu, Yogyakarta.

5. SARAN

Adapun saran-saran yang dapat penulis berikan untuk pengembangan lebih lanjut terhadap penelitian Tugas Akhir ini adalah :

1. Agar pada penelitian selanjutnya dapat mengimplementasikan algoritma pencarian *Depth First Search* (DFS) untuk aplikasi *game* lainnya yang memungkinkan untuk pencarian solusi.
2. Agar penelitian berikutnya di lain aplikasi tidak menggunakan pohon pencarian solusi terlalu banyak, sehingga memungkinkan untuk algoritmapencarian *Depth First Search* (DFS) bekerja secara optimal.
3. Dilakukan penelitian yang lebih lanjut mengenai pembangunan aplikasi *game* lainnya menggunakanalgoritmapencarian lain selain *Depth First Search* (DFS), sehingga penggunaanalgoritmapencarian lebih optimal.
4. Untuk pengembangan sistem ini, selanjutnya ditambahkan *database* untuk nama, skor, akumulasi skor dan level.

DAFTAR PUSTAKA

- [1] Satyaputra dan Aritonang, E., 2014, *Beginning Android Programming with ADT Bundle*, Elix Media Komputerindo, Jakarta.
- [2] Suyanto., 2011, *Artificial Intelligence Searching-Reasoning-Planning-Learning*, Edisi Revisi, Bandung.
- [3] Widiyanto, A., 2014, Pengertian Game, <http://widi625.blogspot.co.id/2014/04/pengertian-game.html>,