

Laporan Tugas Kecil 1

IF2211 Strategi Algoritma

Penyelesaian Permainan Kartu 24 dengan Algoritma
Brute Force



Disusun oleh:

Muhammad Habibi Husni
13521169
K01

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2023

1. Algoritma *Brute Force*

Program menggunakan algoritma Brute Force dengan langkah-langkah sebagai berikut.

1. Program menerima konfigurasi kartu yang akan dicari solusinya.
2. Mengiterasi setiap kemungkinan pemilihan bilangan pertama, kedua, ketiga, dan keempat dari keempat kartu. Secara berurutan bilangan tersebut akan dimisalkan sebagai a, b, c, dan d.
3. Mencari permutasi ketiga operator yang akan dipakai pada persamaan. Persamaan akan berbentuk sebagai berikut: $a \text{ op } b \text{ op } c \text{ op } d$, dengan op dapat berbentuk tanda tambah (+), kurang (-), kali (*), dan bagi (/).
4. Cek kebenaran hasil persamaan dari setiap kemungkinan pemasangan tanda kurung. Jika diasumsikan setiap urutan operator sudah ditentukan oleh tanda kurung, maka kemungkinan yang ada dari pemasangan tanda kurung adalah sebagai berikut :
 - a. $((a \text{ op } b) \text{ op } c) \text{ op } d$
 - b. $(a \text{ op } (b \text{ op } c)) \text{ op } d$
 - c. $a \text{ op } ((b \text{ op } c) \text{ op } d)$
 - d. $a \text{ op } (b \text{ op } (c \text{ op } d))$
 - e. $(a \text{ op } b) \text{ op } (c \text{ op } d)$Asumsi diberlakukan untuk memperjelas presedensi dari operasi serta mempersimpel permasalahan.
5. Apabila hasil dari sebuah persamaan sama dengan 24, maka string persamaan tersebut dimasukkan kedalam sebuah *set of string*. ADT *set* digunakan agar tidak terjadi duplikasi jawaban terhadap nomor kartu yang sama.
6. Tampilkan seluruh persamaan pada layar dan/atau file.

2. Source Code

Pada implementasi digunakan bahasa pemrograman c++ sebagai berikut :

```
#include <iostream>
#include <set>
#include <random>
#include <chrono>
#include <time.h>
#include <fstream>
using namespace std;
using namespace std::chrono;

const char opchar[4] = {'+', '-', '*', '/'};

double f(double operand1, int op, double operand2){
    // menghitung nilai a op b
    if (op == 0) return operand1 += operand2;
    if (op == 1) return operand1 -= operand2;
    if (op == 2) return operand1 *= operand2;
    if (op == 3) return operand1 /= operand2;
    return operand1;
}

bool isEqual(double a, double b){
    // Mengecek apakah bilangan riil a dan b sama
    return abs(a-b) <= 0.00000001;
}

void printKartu(int kartu[4], ostream &outstream){
    // Mencetak kartu pada output stream tertentu
    for (int i = 0; i < 4; i++){
        if (i != 0) outstream << " ";
        if (kartu[i] == 1) outstream << "A";
        else if (kartu[i] == 11) outstream << "J";
        else if (kartu[i] == 12) outstream << "Q";
        else if (kartu[i] == 13) outstream << "K";
        else outstream << kartu[i];
    }
    outstream << endl;
}

void printSolusi(set<string> ans, ostream &outstream) {
    // Mencetak solusi pada output stream tertentu
    if (ans.empty()){
        outstream << "No solutions found!\n";
    } else {
        outstream << ans.size() << " solutions found!\n";
        for (auto solusi : ans){
```

```

        ostream << solusi << endl;
    }
}

int main(){
    srand(time(0));
    string input;
    int kartu[4];
    cout << "***** 24-SOLVER *****\n";
    cout << "Pilih metode memilih kartu :\n";
    cout << "1. Generate kartu secara random\n";
    cout << "2. Input melalui terminal\n";
    do {
        cout << "Pilihan anda : ";
        cin >> input;
        if (input != "1" && input != "2"){
            cout << "Mohon masukkan pilihan yang sesuai.\n";
        }
    } while (input != "1" && input != "2");
    if (input == "1"){
        // input secara random
        cout << "Kartu Anda :\n";
        for (int i = 0; i < 4; i++) kartu[i] = rand() % 13+1;
        printKartu(kartu, cout);
    } else {
        // input oleh pengguna
        cout << "Masukkan nilai empat kartu yang valid\n";
        cout << "(A,2,3,4,5,6,7,8,9,10,J,Q,K) dengan dipisahkan spasi :\n";
        char tmp;
        getchar(); //flush newline char from cin stream
        while (true){
            string s;
            getline(cin, s);
            int inpCnt = 0;
            bool fail = false;
            int i = 0;
            while (i < s.length() && !fail){
                string word = "";
                while (i < s.length() && s[i] == ' ') i++;
                while (i < s.length() && s[i] != ' '){
                    word += s[i];
                    i++;
                }
                if (inpCnt == 4){
                    if (word.length() != 0) fail = true;
                    else break;
                } else if (word == "10"){

```

```

        kartu[inpCnt] = 10;
        inpCnt++;
    } else if (word.length() == 1){
        if (word[0] >= '2' && word[0] <= '9'){
            kartu[inpCnt] = word[0] - '0';
            inpCnt++;
        } else if (word[0] == 'A'){
            kartu[inpCnt] = 1;
            inpCnt++;
        } else if (word[0] == 'J'){
            kartu[inpCnt] = 11;
            inpCnt++;
        } else if (word[0] == 'Q'){
            kartu[inpCnt] = 12;
            inpCnt++;
        } else if (word[0] == 'K'){
            kartu[inpCnt] = 13;
            inpCnt++;
        } else {
            fail = true;
        }
    } else {
        fail = true;
    }
}
if (inpCnt != 4) fail = true;
if (!fail) break;
cout << "Input tidak valid! Mohon masukkan ulang.\n";
}
}

auto start = high_resolution_clock::now();
set<string> ans;
for (int a = 0; a < 4; a++){
    for (int b = 0; b < 4; b++){
        if (b == a) continue;
        for (int c = 0; c < 4; c++){
            if (c == a || c == b) continue;
            for (int d = 0; d < 4; d++){
                if (d == a || d == b || d == c) continue;
                int op[3];
                for (op[0] = 0; op[0] < 4; op[0]++){
                    for (op[1] = 0; op[1] < 4; op[1]++){
                        for (op[2] = 0; op[2] < 4; op[2]++){
                            if
(isEqual(f(f(f(kartu[a], op[0], kartu[b]), op[1], kartu[c]), op[2], kartu[d]), 24))
{
                                // Bentuk ((a op b) op c) op d
                                string tmp = "";

```

```

        tmp = tmp + "(" + to_string(kartu[a]) + " " + opchar[op[0]]
+ " " + to_string(kartu[b]) + ") ";
        tmp = tmp + opchar[op[1]] + " " + to_string(kartu[c]) + "
";
        tmp = tmp + opchar[op[2]] + " " + to_string(kartu[d]);
        ans.insert(tmp);
    }
    if
(isEqual(f(f(kartu[a],op[0],f(kartu[b],op[1],kartu[c]) ) ,op[2],kartu[d]),24))
{
        // Bentuk (a op (b op c)) op d
        string tmp = "";
        tmp = tmp + "(" + to_string(kartu[a]) + " " + opchar[op[0]]
+ " ";
        tmp = tmp + "(" + to_string(kartu[b]) + " " + opchar[op[1]]
+ " " + to_string(kartu[c]) + ") ) ";
        tmp = tmp + opchar[op[2]] + " " + to_string(kartu[d]);
        ans.insert(tmp);
    }
    if
(isEqual(f(kartu[a],op[0],f(f(kartu[b],op[1],kartu[c]) ,op[2],kartu[d]) ),24))
{
        // Bentuk a op ((b op c) op d)
        string tmp = "";
        tmp = tmp + to_string(kartu[a]) + " " + opchar[op[0]] + " ";
        tmp = tmp + "(" + to_string(kartu[b]) + " " + opchar[op[1]]
+ " " + to_string(kartu[c]) + ") ";
        tmp = tmp + opchar[op[2]] + " " + to_string(kartu[d]) + ")";
        ans.insert(tmp);
    }
    if
(isEqual(f(kartu[a],op[0],f(kartu[b],op[1],f(kartu[c],op[2],kartu[d]))),24)){
        // Bentuk a op (b op (c op d))
        string tmp = "";
        tmp = tmp + to_string(kartu[a]) + " " + opchar[op[0]] + " ";
        tmp = tmp + "(" + to_string(kartu[b]) + " " + opchar[op[1]]
+ " ";
        tmp = tmp + "(" + to_string(kartu[c]) + " " + opchar[op[2]]
+ " " + to_string(kartu[d]) + ") )";
        ans.insert(tmp);
    }
    if
(isEqual(f(f(kartu[a],op[0],kartu[b]),op[1],f(kartu[c],op[2],kartu[d])),24)){
        // Bentuk (a op b) op (c op d)
        string tmp = "";
        tmp = tmp + "(" + to_string(kartu[a]) + " " + opchar[op[0]]
+ " " + to_string(kartu[b]) + ") ";
        tmp = tmp + opchar[op[1]] + " ";

```

```

        tmp = tmp + "(" + to_string(kartu[c]) + " " + opchar[op[2]]
+ " " + to_string(kartu[d]) + ")";
        ans.insert(tmp);
    }
}
}
}
}
}
}
}
auto stop = high_resolution_clock::now();
auto duration = duration_cast<microseconds>(stop - start);
printSolusi(ans,cout);
cout << "Runtime : " << duration.count() << " microseconds\n";
do {
    cout << "Apakah Anda ingin mencetak solusi pada sebuah file (y/n)? ";
    cin >> input;
    if (input != "y" && input != "n"){
        cout << "Input tidak valid! Mohon masukkan ulang.\n";
    }
} while (input != "y" && input != "n");
if (input == "y"){
    string filename;
    ofstream file;
    cout << "Masukkan nama file beserta ekstensi : ";
    cin >> filename;
    file.open("../test\\"+filename);
    printKartu(kartu,file);
    printSolusi(ans,file);
    file.close();
    cout << "Solusi berhasil dicetak pada test\\" + filename + ".!\n";
}
}

```

3. Input dan Output

<div>Masukkan nilai empat kartu yang valid (A,2,3,4,5,6,7,8,9,10,J,Q,K) dengan dipisahkan spasi : 8 8 Q 9 4 solutions found! 1. (12 * 8) - (8 * 9) 2. (12 * 8) - (9 * 8) 3. (8 * 12) - (8 * 9) 4. (8 * 12) - (9 * 8) Runtime : 335 microseconds</div>				
<div>Masukkan nilai empat kartu yang valid (A,2,3,4,5,6,7,8,9,10,J,Q,K) dengan dipisahkan spasi : 6 6 6 6 7 solutions found! 1. ((6 * 6) - 6) - 6 2. ((6 + 6) + 6) + 6 3. (6 * 6) - (6 + 6) 4. (6 + (6 + 6)) + 6 5. (6 + 6) + (6 + 6) 6. 6 + ((6 + 6) + 6) 7. 6 + (6 + (6 + 6)) Runtime : 299 microseconds</div>				
<div>Masukkan nilai empat kartu yang valid (A,2,3,4,5,6,7,8,9,10,J,Q,K) dengan dipisahkan spasi : K 6 2 Q 11 solutions found! 1. ((12 * 13) / 6) - 2 2. ((12 / 6) * 13) - 2 3. ((13 * 12) / 6) - 2 4. ((13 / 6) * 12) - 2 5. (12 * (13 / 6)) - 2 6. (12 / (6 / 13)) - 2 7. (13 * (12 / 6)) - 2 8. (13 * 2) - (12 / 6) 9. (13 / (6 / 12)) - 2 10. (2 * 13) - (12 / 6) 11. 12 / ((13 / 2) - 6) Runtime : 282 microseconds</div>				
<div>Masukkan nilai empat kartu yang valid (A,2,3,4,5,6,7,8,9,10,J,Q,K) dengan dipisahkan spasi : A 1 2 3 Input tidak valid! Mohon masukkan ulang. A 2 3 4 242 solutions found! 1. ((1 * 2) * 3) * 4 2. ((1 * 2) * 4) * 3 3. ((1 * 3) * 2) * 4 4. ((1 * 3) * 4) * 2 5. ((1 * 4) * 2) * 3 6. ((1 * 4) * 3) * 2 7. ((1 + 2) + 3) * 4 8. ((1 + 3) + 2) * 4 9. ((2 * 1) * 3) * 4 10. ((2 * 1) * 4) * 3 11. ((2 * 3) * 1) * 4 12. ((2 * 3) * 4) * 1 13. ((2 * 3) * 4) / 1 14. ((2 * 3) / 1) * 4 15. ((2 * 4) * 1) * 3 16. ((2 * 4) * 3) * 1 17. ((2 * 4) * 3) / 1 18. ((2 * 4) / 1) * 3 19. ((2 + 1) + 3) * 4 20. ((2 + 3) + 1) * 4 21. ((2 / 1) * 3) * 4 22. ((2 / 1) * 4) * 3 23. ((3 * 1) * 2) * 4 24. ((3 * 1) * 4) * 2 25. ((3 * 2) * 1) * 4 26. ((3 * 2) * 4) * 1 27. ((3 * 2) * 4) / 1 28. ((3 * 2) / 1) * 4 29. ((3 * 4) * 1) * 2 30. ((3 * 4) * 2) * 1 31. ((3 * 4) * 2) / 1 32. ((3 * 4) / 1) * 2</div>	<div>33. ((3 + 1) + 2) * 4 34. ((3 + 2) + 1) * 4 35. ((3 / 1) * 2) * 4 36. ((3 / 1) * 4) * 2 37. ((4 * 1) * 2) * 3 38. ((4 * 1) * 3) * 2 39. ((4 * 2) * 1) * 3 40. ((4 * 2) * 3) * 1 41. ((4 * 2) * 3) / 1 42. ((4 * 2) / 1) * 3 43. ((4 * 3) * 1) * 2 44. ((4 * 3) * 2) * 1 45. ((4 * 3) * 2) / 1 46. ((4 * 3) / 1) * 2 47. ((4 / 1) * 2) * 3 48. ((4 / 1) * 3) * 2 49. (1 * (2 * 3)) * 4 50. (1 * (2 * 4)) * 3 51. (1 * (3 * 2)) * 4 52. (1 * (3 * 4)) * 2 53. (1 * (4 * 2)) * 3 54. (1 * (4 * 3)) * 2 55. (1 * 2) * (3 * 4) 56. (1 * 2) * (4 * 3) 57. (1 * 3) * (2 * 4) 58. (1 * 3) * (4 * 2) 59. (1 * 4) * (2 * 3) 60. (1 * 4) * (3 * 2) 61. (1 + (2 + 3)) * 4 62. (1 + (3 + 2)) * 4 63. (1 + 3) * (2 + 4) 64. (1 + 3) * (4 + 2) 65. (2 * (1 * 3)) * 4 66. (2 * (1 * 4)) * 3 67. (2 * (3 * 1)) * 4 68. (2 * (3 * 4)) * 1 69. (2 * (3 * 4)) / 1 70. (2 * (3 / 1)) * 4 71. (2 * (4 * 1)) * 3 72. (2 * (4 * 3)) * 1</div>	<div>73. (2 * (4 * 3)) / 1 74. (2 * (4 / 1)) * 3 75. (2 * 1) * (3 * 4) 76. (2 * 1) * (4 * 3) 77. (2 * 3) * (1 * 4) 78. (2 * 3) * (4 * 1) 79. (2 * 3) * (4 / 1) 80. (2 * 3) / (1 / 4) 81. (2 * 4) * (1 * 3) 82. (2 * 4) * (3 * 1) 83. (2 * 4) * (3 / 1) 84. (2 * 4) / (1 / 3) 85. (2 + (1 + 3)) * 4 86. (2 + (3 + 1)) * 4 87. (2 + 4) * (1 + 3) 88. (2 + 4) * (3 + 1) 89. (2 / (1 / 3)) * 4 90. (2 / (1 / 4)) * 3 91. (2 / 1) * (3 * 4) 92. (2 / 1) * (4 * 3) 93. (3 * (1 * 2)) * 4 94. (3 * (1 * 4)) * 2 95. (3 * (2 * 1)) * 4 96. (3 * (2 * 4)) * 1 97. (3 * (2 * 4)) / 1 98. (3 * (2 / 1)) * 4 99. (3 * (4 * 1)) * 2 100. (3 * (4 * 2)) * 1 101. (3 * (4 * 2)) / 1 102. (3 * (4 / 1)) * 2 103. (3 * 1) * (2 * 4) 104. (3 * 1) * (4 * 2) 105. (3 * 2) * (1 * 4) 106. (3 * 2) * (4 * 1) 107. (3 * 2) * (4 / 1) 108. (3 * 2) / (1 / 4) 109. (3 * 4) * (1 * 2) 110. (3 * 4) * (2 * 1) 111. (3 * 4) * (2 / 1) 112. (3 * 4) / (1 / 2) 113. (3 + (1 + 2)) * 4</div>	<div>114. (3 + (2 + 1)) * 4 115. (3 + 1) * (2 + 4) 116. (3 + 1) * (4 + 2) 117. (3 / (1 / 2)) * 4 118. (3 / (1 / 4)) * 2 119. (3 / 1) * (2 * 4) 120. (3 / 1) * (4 * 2) 121. (4 * (1 * 2)) * 3 122. (4 * (1 * 3)) * 2 123. (4 * (2 * 1)) * 3 124. (4 * (2 * 3)) * 1 125. (4 * (2 * 3)) / 1 126. (4 * (2 / 1)) * 3 127. (4 * (3 * 1)) * 2 128. (4 * (3 * 2)) * 1 129. (4 * (3 * 2)) / 1 130. (4 * (3 / 1)) * 2 131. (4 * 1) * (2 * 3) 132. (4 * 1) * (3 * 2) 133. (4 * 2) * (1 * 3) 134. (4 * 2) * (3 * 1) 135. (4 * 2) * (3 / 1) 136. (4 * 2) / (1 / 3) 137. (4 * 3) * (1 * 2) 138. (4 * 3) * (2 * 1) 139. (4 * 3) * (2 / 1) 140. (4 * 3) / (1 / 2) 141. (4 + 2) * (1 + 3) 142. (4 + 2) * (3 + 1) 143. (4 / (1 / 2)) * 3 144. (4 / (1 / 3)) * 2 145. (4 / 1) * (2 * 3) 146. (4 / 1) * (3 * 2) 147. 1 * ((2 * 3) * 4) 148. 1 * ((2 * 4) * 3) 149. 1 * ((3 * 2) * 4) 150. 1 * ((3 * 4) * 2) 151. 1 * ((4 * 2) * 3) 152. 1 * ((4 * 3) * 2) 153. 1 * (2 * (3 * 4))</div>	<div>153. 1 * (2 * (3 * 4)) 154. 1 * (2 * (4 * 3)) 155. 1 * (3 * (2 * 4)) 156. 1 * (3 * (4 * 2)) 157. 1 * (4 * (2 * 3)) 158. 1 * (4 * (3 * 2)) 159. 2 * ((1 * 3) * 4) 160. 2 * ((1 * 4) * 3) 161. 2 * ((3 * 1) * 4) 162. 2 * ((3 * 4) * 1) 163. 2 * ((3 * 4) / 1) 164. 2 * ((3 / 1) * 4) 165. 2 * ((4 * 1) * 3) 166. 2 * ((4 * 3) * 1) 167. 2 * ((4 * 3) / 1) 168. 2 * ((4 / 1) * 3) 169. 2 * (1 * (3 * 4)) 170. 2 * (1 * (4 * 3)) 171. 2 * (3 * (1 * 4)) 172. 2 * (3 * (4 * 1)) 173. 2 * (3 * (4 / 1)) 174. 2 * (3 / (1 / 4)) 175. 2 * (4 * (1 * 3)) 176. 2 * (4 * (3 * 1)) 177. 2 * (4 * (3 / 1)) 178. 2 * (4 / (1 / 3)) 179. 2 / ((1 / 3) / 4) 180. 2 / ((1 / 4) / 3) 181. 2 / (1 * (3 * 4)) 182. 2 / (1 / (4 * 3)) 183. 3 * ((1 * 2) * 4) 184. 3 * ((1 * 4) * 2) 185. 3 * ((2 * 1) * 4) 186. 3 * ((2 * 4) * 1) 187. 3 * ((2 * 4) / 1) 188. 3 * ((2 / 1) * 4) 189. 3 * ((4 * 1) * 2) 190. 3 * ((4 * 2) * 1) 191. 3 * ((4 * 2) / 1) 192. 3 * ((4 / 1) * 2) 193. 3 * (1 * (2 * 4))</div>

<pre> 193. 3 * (1 * (2 * 4)) 194. 3 * (1 * (4 * 2)) 195. 3 * (2 * (1 * 4)) 196. 3 * (2 * (4 * 1)) 197. 3 * (2 * (4 / 1)) 198. 3 * (2 / (1 / 4)) 199. 3 * (4 * (1 * 2)) 200. 3 * (4 * (2 * 1)) 201. 3 * (4 * (2 / 1)) 202. 3 * (4 / (1 / 2)) 203. 3 / ((1 / 2) / 4) 220. 4 * ((3 + 1) + 2) 221. 4 * ((3 + 2) + 1) 222. 4 * ((3 / 1) * 2) 223. 4 * (1 * (2 * 3)) 224. 4 * (1 * (3 * 2)) 225. 4 * (1 + (2 + 3)) 226. 4 * (1 + (3 + 2)) 227. 4 * (2 * (1 * 3)) 228. 4 * (2 * (3 * 1)) 229. 4 * (2 * (3 / 1)) 230. 4 * (2 + (1 + 3)) 231. 4 * (2 + (3 + 1)) 232. 4 * (2 / (1 / 3)) 233. 4 * (3 * (1 * 2)) 234. 4 * (3 * (2 * 1)) 235. 4 * (3 * (2 / 1)) 236. 4 * (3 + (1 + 2)) 237. 4 * (3 + (2 + 1)) 238. 4 * (3 / (1 / 2)) 239. 4 / ((1 / 2) / 3) 240. 4 / ((1 / 3) / 2) 241. 4 / (1 / (2 * 3)) 242. 4 / (1 / (3 * 2)) Runtime : 537 microseconds </pre>	
<pre> Kartu Anda : 2 K Q 5 16 solutions found! 1. ((12 / 2) + 13) + 5 2. ((12 / 2) + 5) + 13 3. ((13 + 5) * 2) - 12 4. ((5 + 13) * 2) - 12 5. (12 / 2) + (13 + 5) 6. (12 / 2) + (5 + 13) 7. (13 + (12 / 2)) + 5 8. (13 + 5) + (12 / 2) 9. (2 * (13 + 5)) - 12 10. (2 * (5 + 13)) - 12 11. (5 + (12 / 2)) + 13 12. (5 + 13) + (12 / 2) 13. 13 + ((12 / 2) + 5) 14. 13 + (5 + (12 / 2)) 15. 5 + ((12 / 2) + 13) 16. 5 + (13 + (12 / 2)) Runtime : 211 microseconds </pre>	
<pre> Kartu Anda : 9 6 7 10 No solutions found! Runtime : 190 microseconds </pre>	
<pre> Kartu Anda : J Q 2 9 No solutions found! Runtime : 189 microseconds </pre>	

5. Link ke Repository Github

https://github.com/habibibi/Tucil1_13521169

6. Check List Program

Poin	Ya	Tidak
1. Program berhasil dikompilasi tanpa kesalahan	✓	
2. Program berhasil running	✓	
3. Program dapat membaca input / generate sendiri dan memberikan luaran	✓	
4. Solusi yang diberikan program memenuhi (berhasil mencapai 24)	✓	
5. Program dapat menyimpan solusi dalam file teks	✓	