

Tugas Kecil 1 IF2211 Strategi Algoritma

Semester II tahun 2022/2023

**Penyelesaian Permainan Kartu 24 dengan Algoritma
*Brute Force***

Disusun oleh:

Kenneth Dave Bahana 13521145



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

2023

1. Penjelasan Algoritma *Brute Force*

Algoritma *brute force* merupakan algoritma dengan pendekatan yang lempang(*straightforward*) untuk memecahkan suatu persoalan. Algoritma ini bersifat luas karena dapat berupa berbagai algoritma yang sifatnya sederhana dan caranya bersifat jelas dipahami manusia karena ide pembentukan algoritma ini didasarkan apa yang terlintas dalam pemikiran ketika suatu program ingin dibuat.

Pendekatan utama dari algoritma *brute force* adalah dengan menguji seluruh kemungkinan dalam mendapatkan hasil yang memenuhi syarat sebagai solusi dari suatu permasalahan. Pendekatan ini yang menyebabkan algoritma *brute force* seringkali bersifat tidak efisien dikarenakan algoritmanya tidak mengutamakan efisiensi dan kecepatan eksekusi program, namun mengutamakan program dapat berjalan dengan cara yang secara *straightforward* dan optimal sehingga algoritma ini dapat juga disebut algoritma naif (*naïve algorithm*) dikarenakan menggunakan usaha untuk menguji segala kemungkinan dalam pembentukan algoritmanya, dibandingkan dengan memikirkan pengujian cara yang efisien atau membutuhkan jumlah tahapan yang lebih sedikit dalam algoritmanya sehingga tidak perlu menguji setiap kemungkinan.

Pendekatan algoritma *brute force* yang digunakan dalam pembuatan program ini adalah konsep permutasi. Pada permainan 24 ini, untuk mencari solusi yang menghasilkan nilai 24 dari suatu empat kartu, dibutuhkan empat permutasi posisi dari keempat nilai kartu tersebut, serta kombinasi dari operator berupa operasi perkalian, penambahan, pengurangan, serta pembagian yang digunakan antara keempat nilai tersebut. Variasi untuk menguji seluruh kemungkinan dengan menggunakan permutasi didasarkan dengan urutan perhitungan yaitu dimulai dengan kartu pertama dengan kedua, atau dimulai dengan kartuma pertama dengan ketiga, dan seterusnya hingga seluruh variasi terpenuhi. Setiap variasi pengurutan nilai suatu kartu yang dihitung terlebih dahulu ditandai dengan pengurangan, dimana pengurangan tersebut mengindikasikan pengurutan perhitungan setiap nilai pada kartu dan diuji semua serta disimpan setiap nilai yang pada hitungan akhirnya menghasilkan nilai 24. Pada pengujian tersebut, perlu dibuat beberapa Batasan yang tidak memungkinkan, yaitu apabila terdapat operasi pertambahan ataupun pengurangan yang menghasilkan 0 tidak boleh menjadi penyebut dari operasi pembagian selanjutnya karena akan menghasilkan nilai yang tidak terdefinisi dan menyebabkan program error. Secara algoritma *brute force*, permutasi dilakukan dengan melakukan *for loop* sejumlah angka serta operasi

yang dilakukan menggunakan 5 variasi pengurutan perhitungan yang merupakan seluruh variasi kemungkinan urutan perhitungan dari keempat nilai kartu.

Hasil pencarian dari permutasi ini yang menguji untuk setiap nilai kartu (*slot*) dari keempat *slot* tersebut terdapat 13 kemungkinan nilai serta setiap kombinasi operasi, setiap solusi yang memiliki sistematika perhitungan yang benar dan mendapatkan nilai akhir 24 akan dimasukkan ke dalam *list* solusi – solusi yang akan ditampilkan serta ditunjukkan jumlah solusi yang ditemukan, yaitu berbagai cara untuk menyelesaikan permasalahan dari empat nilai yang telah terpilih dan memungkinkan tidak ada solusi sama sekali.

2. *Source Code* dengan Bahasa C++

```
src > main.cpp
1  #include <bits/stdc++.h>
2  #include <ctime>
3  #include <cmath>
4  using namespace std;
5  using namespace std::chrono;
6
7  int strToInt(string x){
8      // ALGORITMA
9      if (x == "A"){return 1;}
10     else if (x == "2"){
11         return 2;
12     }
13     else if (x == "3"){
14         return 3;
15     }
16     else if (x == "4"){
17         return 4;
18     }
19     else if (x == "5"){
20         return 5;
21     }
22     else if (x == "6"){
23         return 6;
24     }
25     else if (x == "7"){
26         return 7;
27     }
28     else if (x == "8"){
29         return 8;
30     }
31     else if (x == "9"){
32         return 9;
33     }
34     else if (x == "10"){
35         return 10;
36     }
37     else if (x == "J"){
```

```

38         return 11;
39     }
40     else if (x == "Q"){
41         return 12;
42     }
43     else if (x == "K"){
44         return 13;
45     }
46     else{
47         return 0;
48     }
49 }
50 string intToStr(int x){
51     // ALGORITMA
52     if (x == 1){
53         return "1";
54     }
55     else if (x == 2){
56         return "2";
57     }
58     else if (x == 3){
59         return "3";
60     }
61     else if (x == 4){
62         return "4";
63     }
64     else if (x == 5){
65         return "5";
66     }
67     else if (x == 6){
68         return "6";
69     }
70     else if (x == 7){
71         return "7";
72     }
73     else if (x == 8){
74         return "8";
75     }
76     else if (x == 9){
77         return "9";
78     }
79     else if (x == 10){
80         return "10";
81     }
82     else if (x == 11){
83         return "11";
84     }
85     else if (x == 12){
86         return "12";
87     }
88     else if (x == 13){
89         return "13";
90     }
91     else{
92         return " ";
93     }
94 }
95 double operasi(double a, double b, string c){
96     // ALGORITMA
97     if (c == "+"){
98         return (a + b);
99     }
100     else if (c == "-"){
101         return (a - b);
102     }
103     else if (c == "/"){
104         if (b != 0){
105             return (a / b);
106         }
107     }
108     else if (c == "*") {
109         return (a * b);
110     }

```

```

111 }
112
113 double hitung(double a, double b, double c, double d, string op1, string op2, string op3, int var){
114     // KAMUS LOKAL
115     double hasil, sub;
116     // ALGORITMA
117     if (var == 1){
118         if ((b == 0 && op1 == " / ") || (c == 0 && op2 == " / ") || (d == 0 && op3 == " / ")){
119             return 0;
120         }
121         else{
122             hasil = operasi(a, b, op1);
123             hasil = operasi(hasil, c, op2);
124             hasil = operasi(hasil, d, op3);
125             return hasil;
126         }
127     }
128     else if (var == 2){
129         if (c == 0 && op2 == " / "){
130             return 0;
131         }
132         else{
133             hasil= operasi(b, c, op2);
134             if ((hasil == 0 && op1 == " / ") || (d == 0 && op3 == " / ")){
135                 return 0;
136             }
137             else{
138                 hasil = operasi(a, hasil, op1);
139                 hasil = operasi(hasil, d, op3);
140                 return hasil;
141             }
142         }
143     }
144     else if (var == 3){
145         // (a op b) op (c op d) --> a ama b, c ama d, hasil ab sama hasil cd
146         if ((b == 0 && op1 == " / ") || d == 0 && op3 == " / "){
147             return 0;
148         }
149         else {
150             hasil = operasi(a, b, op1);
151             sub = operasi(c, d, op3);
152             if (sub == 0 && op2 == " / "){
153                 return 0;
154             }
155             else{
156                 hasil = operasi(hasil, sub, op2);
157                 return hasil;
158             }
159         }
160     }
161     else if (var == 4){
162         // a op ((b op c) op d) --> b ama c, hasil ama d, a sama hasil.
163         if ((c == 0 && op2 == " / ") || (d == 0 && op3 == " / ")){
164             return 0;
165         }
166         else {
167             hasil = operasi(b, c, op2);
168             hasil = operasi(hasil, d, op3);
169             if (hasil == 0 && op1 == " / "){
170                 return 0;
171             }
172             else {
173                 hasil = operasi(a, hasil, op1);
174                 return hasil;
175             }
176         }
177     }
178     else if (var == 5){
179         // a op (b op (c op d)) -- > c ama d, b ama hasil, a ama hasil.
180         hasil = operasi(c, d, op3);
181         hasil = operasi(b, hasil, op2);
182         hasil = operasi(a, hasil, op1);

```

```

183
184   if (d == 0 && op3 == " / "){
185       return 0;
186   }
187   else{
188       hasil = operasi(c, d, op3);
189       if (hasil == 0 && op3 == " / "){
190           return 0;
191       }
192       else{
193           hasil = operasi(b, hasil, op2);
194           if (hasil == 0 && op1 == " / "){
195               return 0;
196           }
197           else{
198               hasil = operasi(a, hasil, op1);
199               return hasil;
200           }
201       }
202   }
203 }
204 else{
205     return 0;
206 }
207 }
208
209 long BruteForce(string x, vector<string>& solutions, int& sol){
210     // KAMUS LOKAL
211     string inpt, solution, op[4] = {" + ", " - ", " / ", " * "};
212     int input[4];
213     bool already, valid = false, save = false;
214     double hasil;
215     char ans;
216     int a, b, c, d, e, lngth = 7, space_count;
217
218     // ALGORITMA
219     if (x == "1"){
220         while (!valid){
221             string sinput[4];
222             bool enough_space = false;
223             space_count = 0;
224             cout << "Nilai-nilai kartu: A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K" << endl;
225             cout << "Masukkan 4 nilai kartu (setiap nilai dipisahkan dengan spasi):" << endl;
226             getline(cin >> ws, inpt);
227             for(char space : inpt){
228                 if (space == ' '){
229                     space_count++;
230                 }
231             }
232             if (space_count == 3){
233                 enough_space = true;
234                 int i = 0, j = 0;
235                 while ((i < inpt.length()) && (j <= 3)){
236                     if (inpt[i] != ' '){
237                         sinput[j] += inpt[i];
238                         i++;
239                     }
240                     else{
241                         i++;
242                         j++;
243                     }
244                 }
245             }
246             if (enough_space){
247                 for (int x = 0; x < 4; x++){
248                     valid = true;
249                     e = strToInt(sinput[x]);
250                     if (e > 0 && e < 14){
251                         input[x] = e;
252                     }
253                     else{

```

```

253         valid = false;
254         break;
255     }
256 }
257 }
258 }
259 }
260 if (x == "2"){
261     valid = true;
262     string sinput[4];
263     srand(time(nullptr));
264     for (int i = 0; i < 4; i++){
265         input[i] = (rand() % 13) + 1;
266         sinput[i] = intToStr(input[i]);
267     }
268     cout << "Kartu yang didapatkan: ";
269     for (int i = 0; i < 4; i++){
270         cout << intToStr(input[i]) << " ";
271     }
272     cout << endl;
273 }
274 //cout << "The four numbers are: " << input[0] << ", " << input[1] << ", " << input[2] << ", " << input[3] << "." << endl;
275 // Algoritma
276 // Case 1
277 int mulai = clock();
278 if (valid){
279     for (string i : op){
280         for (string j : op){
281             for (string k : op){
282                 // looping operasi
283                 for (int idx1 = 0; idx1 < 4; idx1++){
284                     a = input[idx1];
285                     for (int idx2 = 0; idx2 < 4; idx2++){
286                         b = input[idx2];
287                         for (int idx3 = 0; idx3 < 4; idx3++){
288                             c = input[idx3];
289                             for (int idx4 = 0; idx4 < 4; idx4++){
290                                 d = input[idx4];
291                                 // looping (permutasi) input
292                                 if (idx1 != idx2 && idx1 != idx3 && idx1 != idx4 && idx2 != idx3 && idx2 != idx4 && idx3 != idx4){
293                                     // varian 1
294                                     hasil = hitung(a, b, c, d, i, j, k, 1);
295                                     if (hasil == 24){
296                                         solution = "(" + intToStr(a) + i + intToStr(b) + ")" + j + intToStr(c) + ")" + k + intToStr(d);
297                                         already = false;
298                                         for (string check : solutions){
299                                             if (solution == check){
300                                                 already = true;
301                                                 break;
302                                             }
303                                         }
304                                         if (!already){
305                                             solutions.push_back(solution);
306                                             sol++;
307                                         }
308                                     }
309                                     // varian 2
310                                     hasil = hitung(a, b, c, d, i, j, k, 2);
311                                     if (hasil == 24){
312                                         solution = "(" + intToStr(a) + i + "(" + intToStr(b) + j + intToStr(c) + ")" + k + intToStr(d);
313                                         already = false;
314                                         for (string check : solutions){
315                                             if (solution == check){
316                                                 already = true;
317                                                 break;
318                                             }
319                                         }
320                                         if (!already){
321                                             solutions.push_back(solution);
322                                             sol++;
323                                         }
324                                     }
325                                     hasil = hitung(a, b, c, d, i, j, k, 3);

```

```

326         if (hasil == 24){
327             solution = "(" + intToStr(a) + i + intToStr(b) + ")" + j + "(" + intToStr(c) + k + intToStr(d) + ")";
328             already = false;
329             for (string check : solutions){
330                 if (solution == check){
331                     already = true;
332                     break;
333                 }
334             }
335             if (!already){
336                 solutions.push_back(solution);
337                 sol++;
338             }
339         }
340         hasil = hitung(a, b, c, d, i, j, k, 4);
341         if (hasil == 24){
342             solution = intToStr(a) + i + "(" + intToStr(b) + j + intToStr(c) + ")" + k + intToStr(d) + ")";
343             already = false;
344             for (string check : solutions){
345                 if (solution == check){
346                     already = true;
347                     break;
348                 }
349             }
350             if (!already){
351                 solutions.push_back(solution);
352                 sol++;
353             }
354         }
355         // a op (b op (c op d)) --> c ama d, b ama hasil, a ama hasil.
356         hasil = hitung(a, b, c, d, i, j, k, 5);
357         if (hasil == 24){
358             solution = intToStr(a) + i + "(" + intToStr(b) + j + "(" + intToStr(c) + k + intToStr(d) + ")";
359             already = false;
360             for (string check : solutions){
361                 if (solution == check){
362                     already = true;
363                     break;
364                 }
365             }
366             if (!already){
367                 solutions.push_back(solution);
368                 sol++;
369             }
370         }
371     }
372 }
373 }
374 }
375 }
376 }
377 }
378 }
379 }
380 int berhenti = clock();
381 return (berhenti-mulai);
382 }
383
384 void saveFile(vector<string> answers){
385     string ans, filename;
386     bool save;
387     while (!save){
388         if ((ans != "y") || (ans != "Y") || (ans != "n") || (ans != "N")){
389             cout << "Apakah anda ingin menyimpan luaran solusi tersebut? (Y/N)" << endl;
390             getline(cin, ans);
391             save = true;
392         }
393     }
394     if (ans == "y" || ans == "Y"){
395         cout << "Nama file yang diinginkan: (tanpa format)" << endl;
396         getline(cin, filename);
397         //string full = "../bin/" + filename + ".txt";
398         string file = "../bin/" + filename + ".txt";

```



```

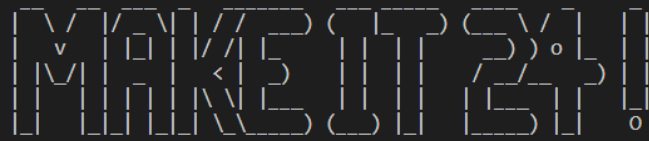
399     int all = 1;
400     ofstream MyFile(file);
401     for (string soll: answers){
402         if (all % 4 == 0){
403             MyFile << soll << endl;
404         }
405         else{
406             MyFile << soll << "    ";
407         }
408         all++;
409     }
410     //MyFile.open(full);
411     cout << "File tersimpan." << endl;
412     MyFile.close();
413 }
414 else{
415     cout << "File tidak tersimpan." << endl;
416 }
417 }
418 int main(){
419     // KAMUS
420     vector<string> answers;
421     string filename = " ";
422     int ex, sol = 0;
423     bool pick = false, save = false;
424     string mode, ans;
425     // ALGORITMA UTAMA
426     cout << "Selamat datang di permainan kartu 24!" << endl;
427     cout << "
428     cout << "|  \ \ /  /|_ \ \ | / / _ ) ( _ _ ) ( _ \ \ /  | _ | |" << endl;
429     cout << "|   v   | | | | / / | _   | | | |   _ ) ) o _ | |" << endl;
430     cout << "|  \ \ /  | _   | < | _ )   | | | |   / _ _ ) | |" << endl;
431     cout << "| | | | | | | | \ \ \ | _   | | | |   | _   | |   | _|" << endl;
432     cout << "| | | | | | | | \ \ \ _ ) ( _ ) | |   | _   | |   o" << "\n" << endl;
433     while (!pick){
434         cout << "Pilihlah (urutan angka) mode permainan yang ingin dimainkan." << endl;
435         cout << "1. Input pilihan pengguna" << endl;
436         cout << "2. Pilihan angka secara acak." << endl;
437         getline(cin, mode);
438         if (mode == "1"){
439             ex = BruteForce(mode, answers, sol);
440             pick = true;
441         }
442         else if (mode == "2"){
443             ex = BruteForce(mode, answers, sol);
444             pick = true;
445         }
446     }
447     // Output
448     int pause = 1;
449     for (string soll: answers){
450         cout << soll << endl;
451     }
452     cout << "Jumlah solusi : " << sol << "." << endl;
453     cout << "Waktu Eksekusi : " << ex << " milisekon." << endl;
454     saveFile(answers);
455 }

```

3. Screenshot dari *Input* dan *Output* dari program

a. Testcase dengan *Input user*

1. Input pertama



Pilihlah (urutan angka) mode permainan yang ingin dimainkan.

1. Input pilihan pengguna

2. Pilihan angka secara acak.

1

Nilai-nilai kartu: A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K

Masukkan 4 nilai kartu (setiap nilai dipisahkan dengan spasi):

A 8 9 Q

$((1 + 9) - 8) * 12$

$(1 + (9 - 8)) * 12$

$((9 + 1) - 8) * 12$

$(9 + (1 - 8)) * 12$

$((1 - 8) + 9) * 12$

$((9 - 8) + 1) * 12$

$(1 - (8 - 9)) * 12$

$(9 - (8 - 1)) * 12$

$(12 - 9) / (1 / 8)$

$((12 - 9) / 1) * 8$

$(12 - (9 / 1)) * 8$

$((12 - 9) * 8) / 1$

$(12 - 9) * (8 / 1)$

$(12 - (1 * 9)) * 8$

$((12 - 9) * 1) * 8$

$(12 - (9 * 1)) * 8$

$(12 - 9) * (1 * 8)$

$((12 - 9) * 8) * 1$

$(12 - 9) * (8 * 1)$

$((12 / 1) - 9) * 8$

$8 / (1 / (12 - 9))$

$(8 / 1) * (12 - 9)$

$12 * ((1 + 9) - 8)$

$12 * (1 + (9 - 8))$

$12 * ((9 + 1) - 8)$

$12 * (9 + (1 - 8))$

$12 * ((1 - 8) + 9)$

$12 * ((9 - 8) + 1)$

$12 * (1 - (8 - 9))$

$12 * (9 - (8 - 1))$

$(8 * (12 - 9)) / 1$

$(8 * (12 - 9)) / 1$

$8 * ((12 - 9) / 1)$

$8 * (12 - (9 / 1))$

$((1 * 12) - 9) * 8$

$(1 * (12 - 9)) * 8$

$1 * ((12 - 9) * 8)$

$8 * (12 - (1 * 9))$

$(8 * (12 - 9)) * 1$

$8 * ((12 - 9) * 1)$

$8 * (12 - (9 * 1))$

$((12 * 1) - 9) * 8$

$8 * ((12 / 1) - 9)$

$(1 * 8) * (12 - 9)$

$1 * (8 * (12 - 9))$

$(8 * 1) * (12 - 9)$

$8 * ((1 * 12) - 9)$

$8 * (1 * (12 - 9))$

$8 * ((12 * 1) - 9)$

Jumlah solusi : 48.

Waktu Eksekusi : 9 milisekon.

Apakah anda ingin menyimpan luaran solusi tersebut? (Y/N)

☐

```
Apakah anda ingin menyimpan luaran solusi tersebut? (Y/N)
y
Nama file yang diinginkan: (tanpa format)
tc1
File tersimpan.
```

```
≡ tc1.txt U X
bin > ≡ tc1.txt
1 ((1 + 9) - 8) * 12 (1 + (9 - 8)) * 12 ((9 + 1) - 8) * 12 (9 + (1 - 8)) * 12
2 ((1 - 8) + 9) * 12 ((9 - 8) + 1) * 12 (1 - (8 - 9)) * 12 (9 - (8 - 1)) * 12
3 (12 - 9) / (1 / 8) ((12 - 9) / 1) * 8 (12 - (9 / 1)) * 8 ((12 - 9) * 8) / 1
4 (12 - 9) * (8 / 1) (12 - (1 * 9)) * 8 ((12 - 9) * 1) * 8 (12 - (9 * 1)) * 8
5 (12 - 9) * (1 * 8) ((12 - 9) * 8) * 1 (12 - 9) * (8 * 1) ((12 / 1) - 9) * 8
6 8 / (1 / (12 - 9)) (8 / 1) * (12 - 9) 12 * ((1 + 9) - 8) 12 * (1 + (9 - 8))
7 12 * ((9 + 1) - 8) 12 * (9 + (1 - 8)) 12 * ((1 - 8) + 9) 12 * ((9 - 8) + 1)
8 12 * (1 - (8 - 9)) 12 * (9 - (8 - 1)) (8 * (12 - 9)) / 1 8 * ((12 - 9) / 1)
9 8 * (12 - (9 / 1)) ((1 * 12) - 9) * 8 (1 * (12 - 9)) * 8 1 * ((12 - 9) * 8)
10 8 * (12 - (1 * 9)) (8 * (12 - 9)) * 1 8 * ((12 - 9) * 1) 8 * (12 - (9 * 1))
11 ((12 * 1) - 9) * 8 8 * ((12 / 1) - 9) (1 * 8) * (12 - 9) 1 * (8 * (12 - 9))
12 (8 * 1) * (12 - 9) 8 * ((1 * 12) - 9) 8 * (1 * (12 - 9)) 8 * ((12 * 1) - 9)
13
```

2. Input kedua (tidak memiliki solusi)

```
Pilihlah (urutan angka) mode permainan yang ingin dimainkan.
1. Input pilihan pengguna
2. Pilihan angka secara acak.
1
Nilai-nilai kartu: A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K
Masukkan 4 nilai kartu (setiap nilai dipisahkan dengan spasi):
7 7 7 7
Jumlah solusi : 0.
Waktu Eksekusi : 8 milisekon.
```

3. Input ketiga (apabila tidak tepat)

```
Pilihlah (urutan angka) mode permainan yang ingin dimainkan.
1. Input pilihan pengguna
2. Pilihan angka secara acak.
1
Nilai-nilai kartu: A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K
Masukkan 4 nilai kartu (setiap nilai dipisahkan dengan spasi):
1 2 A 4
Nilai-nilai kartu: A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K
Masukkan 4 nilai kartu (setiap nilai dipisahkan dengan spasi):
11 12 13 14
Nilai-nilai kartu: A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K
Masukkan 4 nilai kartu (setiap nilai dipisahkan dengan spasi):
A 2 3 4 5
Nilai-nilai kartu: A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K
Masukkan 4 nilai kartu (setiap nilai dipisahkan dengan spasi):
5 4 3 2 1
Nilai-nilai kartu: A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K
Masukkan 4 nilai kartu (setiap nilai dipisahkan dengan spasi):
A A A B
Nilai-nilai kartu: A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K
Masukkan 4 nilai kartu (setiap nilai dipisahkan dengan spasi):
A A A A A
Nilai-nilai kartu: A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K
Masukkan 4 nilai kartu (setiap nilai dipisahkan dengan spasi):
A A A A
Jumlah solusi : 0.
Waktu Eksekusi : 10 milisekon.
```

b. Testcase dengan nilai *random generator*

1. Testcase 1

NAME IT

Pilihlah (urutan angka) mode permainan yang ingin dimainkan.

1. Input pilihan pengguna
2. Pilihan angka secara acak.

2

Kartu yang didapatkan: 5 10 7 12

$$((10 + 7) + 12) - 5$$
$$(10 + (7 + 12)) - 5$$
$$(10 + 7) + (12 - 5)$$
$$10 + ((7 + 12) - 5)$$
$$10 + (7 + (12 - 5))$$
$$((10 + 12) + 7) - 5$$
$$(10 + (12 + 7)) = 5$$
$$(10 + 12) + (7 - 5)$$
$$10 + ((12 + 7) - 5)$$
$$10 + (12 + (7 - 5))$$
$$((7 + 10) + 12) - 5$$
$$(7 + (10 + 12)) = 5$$
$$(7 \pm 10) \pm (12 - 5)$$
$$7 + ((10 + 12) - 5)$$
$$7 \pm (10 \pm (12 - 5))$$
$$((7 + 12) + 10) - 5$$
$$(7 + (12 + 10)) = 5$$
$$(7 + (12 + 10)) = 5$$

$$(7 + 12) + (10 - 5)$$
$$7 + ((12 + 10) - 5)$$
$$7 + ((12 + 10) - 5)$$
$$7 + (12 + (10 - 5))$$

$$((12 + 10) + 7) = 5$$
$$((12 + 10) + 7) - 5$$
$$(12 + (10 + 7)) - 5$$
$$(12 + 10) + (7 - 5)$$
$$12 + ((10 + 7) - 5)$$
$$12 + (10 + (7 - 5))$$

```

((12 + 7) + 10) - 5
(12 + (7 + 10)) - 5
(12 + 7) + (10 - 5)
12 + ((7 + 10) - 5)
12 + (7 + (10 - 5))
((10 + 7) - 5) + 12
(10 + (7 - 5)) + 12
10 + ((7 - 5) + 12)
((10 + 12) - 5) + 7
(10 + (12 - 5)) + 7
10 + ((12 - 5) + 7)
((7 + 10) - 5) + 12
(7 + (10 - 5)) + 12
7 + ((10 - 5) + 12)
((7 + 12) - 5) + 10
(7 + (12 - 5)) + 10
7 + ((12 - 5) + 10)
((12 + 10) - 5) + 7
(12 + (10 - 5)) + 7
12 + ((10 - 5) + 7)
((12 + 7) - 5) + 10
(12 + (7 - 5)) + 10
12 + ((7 - 5) + 10)
(10 + 7) - (5 - 12)
10 + (7 - (5 - 12))
(10 + 12) - (5 - 7)
10 + (12 - (5 - 7))
(7 + 10) - (5 - 12)
7 + (10 - (5 - 12))
(7 + 12) - (5 - 10)
7 + (12 - (5 - 10))
(12 + 10) - (5 - 7)
12 + (10 - (5 - 7))
(12 + 7) - (5 - 10)
12 + (7 - (5 - 10))
((5 + 7) - 10) * 12
(5 + (7 - 10)) * 12
((7 + 5) - 10) * 12
(7 + (5 - 10)) * 12

```

```

((7 - 5) + 12) + 10
(7 - 5) + (12 + 10)
((12 - 5) + 10) + 7
(12 - 5) + (10 + 7)
((12 - 5) + 7) + 10
(12 - 5) + (7 + 10)
((5 - 10) + 7) * 12
(7 - (10 - 5)) * 12
(7 - (10 - 5)) * 12
(12 - 10) * (5 + 7)
12 * (7 + (5 - 10))
12 * ((5 - 10) + 7)
12 * ((7 - 10) + 5)
12 * (5 - (10 - 7))
12 * (7 - (10 - 5))
Jumlah solusi : 110.
Waktu Eksekusi : 13 milisekon.

```

2. Testcase 2

```
MAKEITSAI
Pilihlah (urutan angka) mode permainan yang ingin dimainkan.
1. Input pilihan pengguna
2. Pilihan angka secara acak.
2
Kartu yang didapatkan: 9 9 10 8
Jumlah solusi : 0.
Waktu Eksekusi : 9 milisekon.
Apakah anda ingin menyimpan luaran solusi tersebut? (Y/N)
```

4. Daftar Pustaka

Repository github: https://github.com/kenndave/Tucil1_13521145

5. Checklist

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