

**Laporan Tugas Kecil
Tugas Besar IF2211 Strategi Algoritma
Penyelesaian Permainan Kartu 24 dengan Algoritma Brute Force**

Oleh
Raynard Tanadi / 13521143



**Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
INSTITUT TEKNOLOGI BANDUNG
2023**

DAFTAR ISI

DAFTAR ISI	2
BAB I	3
A. Pengertian	3
B. Alur Algoritma Brute Force	3
BAB II	8
A. Tucil.cpp (semua kode berada di sini, dibagi menjadi berbagai fungsi)	8
BAB III	16
A. Contoh 1 (User Input, Tidak ada solusi, File txt)	16
B. Contoh 2 (User Input, Solusi sedikit, File txt)	18
C. Contoh 3 (User Input, Solusi banyak, File txt)	20
D. Contoh 4 (Random input, File txt)	23
E. Contoh 5 (Random input, File txt)	26
F. Contoh 6 (Random input, File txt)	28
BAB IV	30
A. Link Repository GitHub	30
B. Tabel Cek List	30

BAB I

ALGORITMA BRUTE FORCE

A. Pengertian

Algoritma Brute Force adalah algoritma yang menggunakan pendekatan yang lempang atau *straightforward* untuk memecahkan suatu persoalan. Algoritma ini juga memiliki beberapa karakteristik, seperti sederhana, langsung, dan jelas caranya.

B. Alur Algoritma Brute Force

Pada tugas kecil 1 kali ini, mahasiswa IF2211 ditugaskan untuk mencari solusi dari permainan kartu 24 dengan menggunakan algoritma brute force. Untuk alur dari algoritma brute force pada program ini antara lain sebagai berikut :

1. Melakukan permutasi dari keempat input yang didapatkan dengan menggunakan nested loop dan beberapa pengkondisian.
2. Melakukan perhitungan dengan mencoba segala kemungkinan dari kurung dan operator yang ada dengan membaginya menjadi lima kasus kurung dan melakukan nested loop untuk operator di setiap kasusnya.
3. Terakhir, menghitung total dari solusi yang ada dengan menghitung isi dari vector yang berupa kumpulan dari solusi yang ada yang dimasukkan setiap kali ditemukan solusi.

Dengan alur seperti di atas, berikut adalah kodenya yang dituliskan dalam bahasa C++ :

```

void permutasiAngka(double a[], string kartu[]){
    clock_t start,end;
    start = clock();
    double temptemp[96];
    vector<string> v;
    bool beda;
    int jumlah = 0;
    for (int j = 0;j<4;j++){
        for (int k = 0;k<4;k++){
            for (int l = 0;l<4;l++){
                for (int m = 0;m<4;m++){
                    beda = true;
                    if ((j!=k)&& (j!=l)&&(j!=m)&&(k!=l)&&(k!=m)&&(l!=m)){
                        if (jumlah>0){
                            int n = 0;
                            while (n<jumlah){
                                if ((temptemp[n]==a[j])&&(temptemp[n+1]==a[k])&&(temptemp[n+2]==a[l])&&(temptemp[n+3]==a[m])){
                                    beda = false;
                                    n=jumlah;
                                }
                                n +=4;
                            }
                        }
                        if(beda==true){
                            v = algo(a[j],a[k],a[l],a[m],v);
                            temptemp[jumlah]=a[j];
                            jumlah+=1;
                            temptemp[jumlah] = a[k];
                            jumlah+=1;
                            temptemp[jumlah]=a[l];
                            jumlah+=1;
                            temptemp[jumlah] = a[m];
                            jumlah+=1;
                        }
                    }
                }
            }
        }
    }
}

```

Kode diatas merupakan realisasi dari langkah pertama. Pada algoritma tersebut, dilakukan permutasi dengan menggunakan nested loop sebanyak empat kali. Masing - masing dari huruf j, k, l, dan m merepresentasikan keempat kartu pada permainan 24 sehingga pada setiap loop keempat huruf tersebut tidak boleh sama, maksudnya huruf l tidak boleh bernilai 1 jika k sudah bernilai 1.

Lalu, program akan mengecek apakah urutan yang keempat kartu sekarang sudah pernah disimpan atau belum. Oleh karena itu, dibutuhkan array/list bernama temptemp untuk menyimpan dan mengeceknya.

Pada setiap urutan yang ada akan dilakukan percobaan semua kemungkinan operator dan kurung yang ada dengan memanggil fungsi algo.

```

vector<string> algo(double pertama, double kedua, double ketiga, double keempat, vector<string> arr) {

    // Output solusi
    int n = 0;

    string satu = to_string(int(pertama));
    string dua = to_string(int(kedua));
    string tiga = to_string(int(ketiga));
    string empat = to_string(int(keempat));
    //((a b)c)d
    for (int i = 0; i < 4; i++) {
        for (int j = 0; j < 4; j++) {
            for (int l = 0; l < 4; l++) {
                double hasil = pertama;
                hasil = operasi(hasil, kedua, i);
                hasil = operasi(hasil, ketiga, j);
                hasil = operasi(hasil, keempat, l);

                if (hasil == 24) {
                    string solusi = "(" + satu + simbolOperasi(i) + dua + ")" + simbolOperasi(j) + tiga + ")" + simbolOperasi(l) + empat;
                    arr.push_back(solusi);
                }
            }
        }
    }

    // (a (b c))d
    for (int i = 0; i < 4; i++) {
        for (int j = 0; j < 4; j++) {
            for (int l = 0; l < 4; l++) {
                double hasil = kedua;
                hasil = operasi(hasil, ketiga, j);
                hasil = operasi(pertama, hasil, i);
                hasil = operasi(hasil, keempat, l);
            }
        }
    }
}

```

```

        if (hasil == 24) {
            string solusi = "(" + satu + simbolOperasi(i) + "(" + dua + simbolOperasi(j) + tiga + ")" + simbolOperasi(l) + empat;
            arr.push_back(solusi);
        }
    }
}

// a ((b c) d)
for (int i = 0; i < 4; i++) {
    for (int j = 0; j < 4; j++) {
        for (int l = 0; l < 4; l++) {
            double hasil = kedua;
            hasil = operasi(hasil, ketiga, j);
            hasil = operasi(hasil, keempat, l);
            hasil = operasi(pertama, hasil, i);

            if (hasil == 24) {
                string solusi = satu + simbolOperasi(i) + "(" + dua + simbolOperasi(j) + tiga + ")" + simbolOperasi(l) + empat + ")";
                arr.push_back(solusi);
            }
        }
    }
}

// a (b (c d))
for (int i = 0; i < 4; i++) {
    for (int j = 0; j < 4; j++) {
        for (int l = 0; l < 4; l++) {
            double hasil = ketiga;
            hasil = operasi(hasil, keempat, l);
            hasil = operasi(kedua, hasil, j);
            hasil = operasi(pertama, hasil, i);

            if (hasil == 24) {
                string solusi = satu + simbolOperasi(i) + "(" + dua + simbolOperasi(j) + "(" + tiga + simbolOperasi(l) + empat + ")";
                arr.push_back(solusi);
            }
        }
    }
}
}

```

```

    }
}

}

// (a b) (c d)
for (int i = 0; i < 4; i++) {
    for (int j = 0; j < 4; j++) {
        for (int l = 0; l < 4; l++) {
            double hasil = pertama;
            double hasil2 = ketiga;
            hasil = operasi(hasil, kedua, l);
            hasil2 = operasi(hasil2, keempat, l);
            hasil = operasi(hasil, hasil2, j);

            if (hasil == 24) {
                string solusi = "(" + satu + simbolOperasi(i) + dua + ")" + simbolOperasi(j) + "(" + tiga + simbolOperasi(l) + empat + ")";
                arr.push_back(solusi);
            }
        }
    }
}

return arr;
}

```

Kode diatas merupakan realisasi dari langkah kedua. Pada algoritma tersebut, terdapat pembagian kasus menjadi lima kasus kurung, yaitu (a b)c)d, (a (b c))d, a((b c)d), a(b (c d)), (a b) (c d). Lalu, pada setiap kasus dilakukan percobaan semua kemungkinan operator dengan menggunakan nested loop. Jika hasil dari perhitungan bernilai 24, maka akan dianggap menjadi solusi dan disimpan ke dalam vector yang ada.

Cara untuk menggunakan kurung pada perhitungan adalah dengan melakukan operasi pada kedua bilangan yang berada di dalam kurung terlebih dahulu sehingga diperlukan pembagian kasus di awal.

Untuk memudahkan perhitungan, dibuat fungsi bernama operasi sebagai berikut :

```
double operasi(double angka1, double angka2, int pilihan){
    double hasil;
    if(pilihan==0){
        hasil = angka1+angka2;
    }
    else if (pilihan ==1){
        hasil = angka1-angka2;
    }
    else if(pilihan==2){
        hasil= angka1*angka2;
    }
    else if(pilihan==3){
        if (angka2 != 0){
            hasil = angka1/angka2;
        }
    }
    return hasil;
}
```

Fungsi di atas berfungsi untuk melakukan perhitungan sederhana seperti pertambahan, pengurangan, perkalian, dan pembagian. Lalu, fungsi tersebut akan memberikan hasil dari perhitungan tersebut.

```
int total = size(v);
cout<<"Total solusi ada sebanyak "<<total<<endl;
cout<<"===== "<<endl;
```

Terakhir, kode diatas merupakan realisasi dari pencarian total solusi yang ada dengan menghitung jumlah elemen pada vector.

BAB II

SOURCE PROGRAM

A. Tucil.cpp (semua kode berada di sini, dibagi menjadi berbagai fungsi)

```
src > C++ Tucil.cpp > operasi(double, double, int)
1  #include <iostream>
2  #include <vector>
3  #include <chrono>
4  #include <string.h>
5  #include <fstream>
6
7  using namespace std;
8
9  string simbolOperasi (int pilihan){
10     string operasi;
11     if (pilihan == 0){
12         operasi = "+";
13     }
14     else if (pilihan == 1){
15         operasi = "-";
16     }
17     else if (pilihan == 2){
18         operasi = "*";
19     }
20     else if (pilihan == 3){
21         operasi = "/";
22     }
23     return operasi;
24 }
25
26 double operasi(double angka1, double angka2, int pilihan){
27     double hasil;
28     if(pilihan==0){
29         hasil = angka1+angka2;
30     }
31     else if (pilihan ==1){
32         hasil = angka1-angka2;
33     }
34     else if(pilihan==2){
35         hasil= angka1*angka2;
36     }
37     else if(pilihan==3){
38         if (angka2 != 0){
39             hasil = angka1/angka2;
```



```

40     }
41 }
42 return hasil;
43 }
44
45 double stringtoint(string kata){
46     if (kata == "A"){
47         return 1;
48     }
49     else if (kata == "J"){
50         return 11;
51     }
52     else if (kata == "Q"){
53         return 12;
54     }
55     else if (kata == "K"){
56         return 13;
57     }
58     else {
59         return stoi(kata);
60     }
61 }
62
63 vector<string> algo(double pertama, double kedua, double ketiga, double keempat, vector<string> arr) {
64
65     // Output solusi
66     int n = 0;
67
68     string satu = to_string(int(pertama));
69     string dua = to_string(int(kedua));
70     string tiga = to_string(int(ketiga));
71     string empat = to_string(int(keempat));
72     //((a b)c)d
73     for (int i =0;i<4;i++){
74         for (int j=0;j<4;j++){
75             for(int l = 0;l<4;l++){
76                 double hasil = pertama;
77                 hasil = operasi(hasil,kedua,i);

```

```

78         hasil = operasi(hasil,ketiga,j);
79         hasil = operasi(hasil,keempat,1);
80
81         if (hasil == 24){
82             string solusi = "(" + satu + simbolOperasi(i) + dua + ")" + simbolOperasi(j) + tiga + ")" + simbolOperasi(1) + empat;
83             arr.push_back(solusi);
84         }
85     }
86 }
87
88
89 //a (b c)d
90 for (int i =0;i<4;i++){
91     for (int j=0;j<4;j++){
92         for(int l = 0;l<4;l++){
93             double hasil = kedua;
94             hasil = operasi(hasil,ketiga,j);
95             hasil = operasi(pertama,hasil,i);
96             hasil = operasi(hasil,keempat,l);
97
98             if (hasil == 24){
99                 string solusi = "(" + satu + simbolOperasi(i) + "(" + dua + simbolOperasi(j) + tiga + ")" + simbolOperasi(1) + empat;
100                 arr.push_back(solusi);
101             }
102         }
103     }
104 }
105
106 //a ((b c)d)
107 for (int i =0;i<4;i++){
108     for (int j=0;j<4;j++){
109         for(int l = 0;l<4;l++){
110             double hasil = kedua;
111             hasil = operasi(hasil,ketiga,j);
112             hasil = operasi(hasil,keempat,l);
113             hasil = operasi(pertama,hasil,i);
114

```

```

114         if (hasil == 24){
115             string solusi = satu + simbolOperasi(i) + "(" + dua + simbolOperasi(j) + tiga + ")" + simbolOperasi(1) + empat + ")";
116             arr.push_back(solusi);
117         }
118     }
119 }
120
121
122
123 //a (b (c d))
124 for (int i =0;i<4;i++){
125     for (int j=0;j<4;j++){
126         for(int l = 0;l<4;l++){
127             double hasil = ketiga;
128             hasil = operasi(hasil,keempat,l);
129             hasil = operasi(kedua,hasil,j);
130             hasil = operasi(pertama,hasil,i);
131
132
133             if (hasil == 24){
134                 string solusi = satu + simbolOperasi(i) + "(" + dua + simbolOperasi(j) + "(" + tiga + simbolOperasi(1) + empat + ")";
135                 arr.push_back(solusi);
136             }
137         }
138     }
139 }
140
141 //a (b) (c d)
142 for (int i =0;i<4;i++){
143     for (int j=0;j<4;j++){
144         for(int l = 0;l<4;l++){
145             double hasil = pertama;
146             double hasil2 = ketiga;
147             hasil = operasi(hasil,kedua,i);
148             hasil2 = operasi(hasil2,keempat,l);
149             hasil = operasi(hasil,hasil2,j);
150

```

```

150
151         if (hasil == 24){
152             string solusi = "(" + satu + simbolOperasi(i) + dua + ")" + simbolOperasi(j) + "(" + tiga + simbolOperasi(1) + empat + ")";
153             arr.push_back(solusi);
154         }
155     }
156 }
157 }
158 return arr;
159 }
160
161
162 void savedata( string kartu[], int solusi, vector<string> semuaSolusi){
163     ofstream outdata;
164     bool checker = false;
165     string pilihansave;
166     while (!checker){
167         cout<<"Apakah anda ingin menyimpan solusi sebagai file? (y/n) : ";
168         cin >> pilihansave;
169         if (pilihansave == "y"){
170             checker = true;
171             string namafile;
172             cout<< "Masukan nama file yang diinginkan (tanpa .txt): ";
173             cin >> namafile;
174             outdata.open("test/"+namafile+".txt");
175             outdata<< "Kartu Anda adalah : ";
176             for (int i = 0;i<4;i++){
177                 outdata<< kartu[i]<< " ";
178             }
179             outdata<<endl;
180             outdata<<"===== "<<endl;
181             if (solusi != 0){
182                 outdata<< "Terdapat solusi sebanyak : "<<solusi<<endl;
183                 outdata<<"Solusi: "<<endl;
184                 for (auto elem : semuaSolusi){
185                     outdata<< elem << endl;
186                 }
187             }

```

```

188         else {
189             outdata<<"Tidak ada solusi"<<endl;
190         }
191         outdata<<"===== "<<endl;
192         outdata.close();
193         cout<<"File berhasil disimpan"<<endl;
194     }
195     else if (pilihansave=="n"){
196         checker = true;
197     }
198     else {
199         cout<<"Masukan salah, ulangi masukan"<<endl;
200         checker = false;
201     }
202 }
203
204 }
205
206 void permutasiAngka(double a[], string kartu[]){
207     clock_t start,end;
208     start = clock();
209     double temptemp[96];
210     vector<string> v;
211     bool beda;
212     int jumlah = 0;
213     for (int j = 0;j<4;j++){
214         for (int k = 0;k<4;k++){
215             for (int l = 0;l<4;l++){
216                 for (int m = 0;m<4;m++){
217                     beda = true;
218                     if ((j!=k)&& (j!=l)&&(j!=m)&&(k!=l)&&(k!=m)&&(l!=m)){
219                         if (jumlah>0){
220                             int n = 0;
221                             while (n<jumlah){
222                                 if ((temptemp[n]==a[j])&&(temptemp[n+1]==a[k])&&(temptemp[n+2]==a[l])&&(temptemp[n+3]==a[m])){
223                                     beda = false;
224                                     n=jumlah;
225                                 }
226                                 n +=4;

```

```

227     }
228 }
229 if(beda==true){
230
231     v = algo(a[j],a[k],a[l],a[m],v);
232     temptemp[jumlah]=a[j];
233     jumlah+=1;
234     temptemp[jumlah] = a[k];
235     jumlah+=1;
236     temptemp[jumlah]=a[l];
237     jumlah+=1;
238     temptemp[jumlah] = a[m];
239     jumlah+=1;
240 }
241 }
242 }
243 }
244 }
245 }
246 int total = size(v);
247 cout<<"Total solusi ada sebanyak "<<total<<endl;
248 cout<<"===== "<<endl;
249 cout<<"Berikut adalah solusinya : "<<endl;
250 for (auto elem : v){
251     cout<< elem << endl;
252 }
253 end = clock();
254 double totalwaktu = double(end-start)/double(CLOCKS_PER_SEC);
255 cout<< "Total waktu adalah : "<< totalwaktu << " detik "<<endl;
256 cout<<"===== "<<endl;
257 savedata(kartu,total,v);
258 }
259
260

```

```

261 int main (){
262     // ASCII
263     string ascii;
264     ifstream bacafile ("src/Ascii.txt");
265     while (getline(bacafile, ascii)){
266         cout<<ascii<<endl;
267     }
268     bacafile.close();
269
270
271     cout<<"
272     cout<<"
273     cout<<"
274     cout<<"
275     cout<<"
276     cout<<"
277     cout<<"
278     cout<<"
279     cout<<"
280     cout<<endl;
281
282     // input
283     string Kartu[4];
284     double Kartuint[4];
285     int salah;
286     bool masukan = false;
287     int input;
288     bool loop = true;
289     string loopconfirm;
290
291     while (loop){
292         bool cobalagi = true;
293         cout<<"===== "<<endl;
294         cout<<"Menu : "<<endl;
295         cout<<"1. Input sendiri kartu yang diinginkan"<<endl;
296         cout<<"2. Me-generate kartu secara otomatis"<<endl;
297         cout<<"===== "<<endl;
298         cout<<"Pilih pilihanmu (1/2) : ";
299         cin>>input;

```

```

300     cout<<"===== "<<endl;
301
302
303     if (input==1){
304         cout<<"Masukkan kartu berupa (A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K)" << endl;
305         while (masukan == false) {
306             string str;
307             int j = 0;
308             int k = 0;
309             int totalspasi = 0;
310             salah = 0;
311             cout << "Masukkan kartu : ";
312             getline(cin >> ws, str);
313             char temp = '0';
314             for (char spasi : str){
315                 if (spasi == ' ' && temp != ' '){
316                     totalspasi++;
317                     temp = spasi;
318                 }
319                 else {
320                     temp = spasi;
321                 }
322             }
323             if ( totalspasi != 3){
324                 masukan = false;
325             }
326             else {
327                 while (j <4){
328                     Kartu[j] = str[k];
329                     j++;
330                     k++;
331                     while (str[k]==' '){
332                         k++;
333                     }
334                 }
335             }

```

```

336     cout<<"===== "<<endl;
337     for (int i = 0; i<4;i++){
338         if ((Kartu[i] == "A") || (Kartu[i] == "2") || (Kartu[i] == "3") || (Kartu[i] == "4") || (Kartu[i] == "5") || (Kartu[i] ==
339         {
340             masukan = true;
341         }
342         else {
343             masukan = false;
344             salah +=1;
345         }
346     }
347     if (salah>0){
348         cout<<"Masukan tidak sesuai"<< endl;
349         masukan = false;
350     }
351 }
352 Kartuint[0] = stringtoint(Kartu[0]);
353 Kartuint[1] = stringtoint(Kartu[1]);
354 Kartuint[2] = stringtoint(Kartu[2]);
355 Kartuint[3] = stringtoint(Kartu[3]);
356 }
357 else {
358     srand(time(0));
359     for (int c = 0; c<4; c++){
360         Kartuint[c] = (rand() % 13)+1;
361     }
362     for (int d = 0; d<4; d++){
363         if ((Kartuint[d]!=1)&&(Kartuint[d]<11)){
364             Kartu[d] = to_string(int(Kartuint[d]));
365         }
366         else if (Kartuint[d]==1){
367             Kartu[d] = "A";
368         }
369         else if (Kartuint[d]==11){
370             Kartu[d] = "J";
371         }
372         else if (Kartuint[d]==12){
373             Kartu[d] = "Q";

```

```

336
337
338 || (Kartu[i] == "6") || (Kartu[i] == "7") || (Kartu[i] == "8") || (Kartu[i] == "9") || (Kartu[i] == "10") || (Kartu[i] == "J") || (Kartu[i] == "Q") || (Kartu[i] == "K"))
339
340

```

(Perpanjangan dari line 338)

```

374     }
375     else if (Kartuint[d]==13){
376         Kartu[d] = "K";
377     }
378 }
379 }
380 cout << "Kartu anda adalah : "<<Kartu[0]<<" "<<Kartu[1]<<" "<<Kartu[2]<<" "<<Kartu[3]<<endl;
381 cout<<"===== "<<endl;
382
383
384 permutasiAngka(Kartuint, Kartu);
385
386 cout<<"===== "<<endl;
387
388 while (cobalagi){
389     cobalagi = false;
390     cout<< "Apakah anda ingin mencoba angka lain?(y/n) : ";
391     cin >> loopconfirm;
392     if (loopconfirm == "y"){
393         loop = true;
394         cobalagi = false;
395     }
396     else if (loopconfirm == "n"){
397         loop = false;
398         cobalagi = false;
399         cout<<"Terima kasih sudah menggunakan program ini :) "<<endl;
400     }
401     else {
402         cout<<"Masukan salah, ulangi masukkan "<<endl;
403         cobalagi = true;
404     }
405 }
406 }
407
408
409 return 0;
410 }

```

BAB III

INPUT & OUTPUT PROGRAM

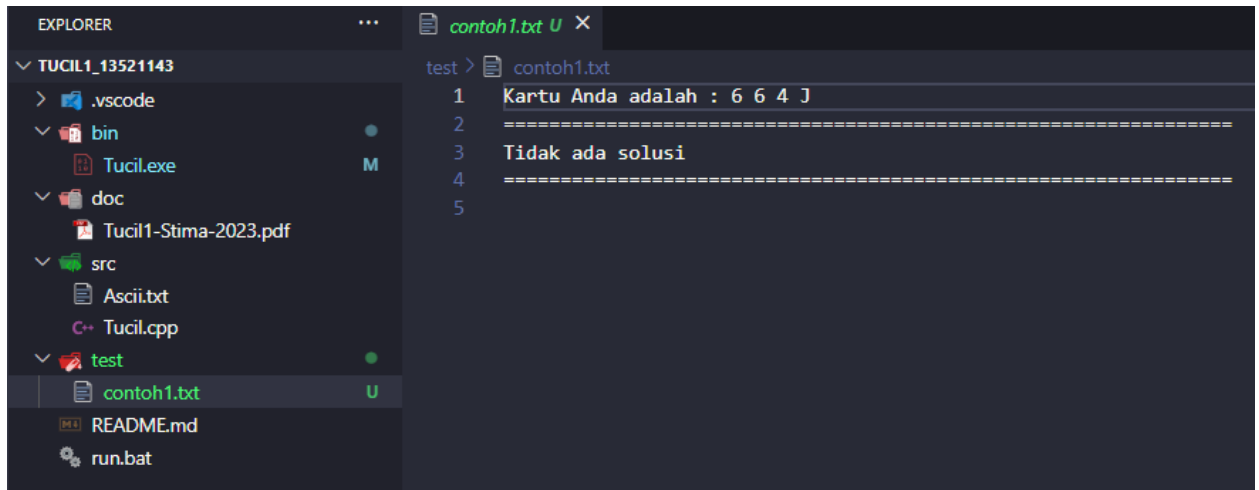
A. Contoh 1 (User Input, Tidak ada solusi, File txt)

```

WELCOME TO
24 GAME

10 ^ | J ^ | Q ^ | K ^ |
^ ^ ^ | ^ ^ ^ | ^ ^ ^ | ^ ^ ^ |
^ ^ ^ | ^ ^ ^ | ^ ^ ^ | ^ ^ ^ |
^ ^ ^ | ^ ^ ^ | ^ ^ ^ | ^ ^ ^ |
_10 | _J | _Q | _K |
_10 | _J | _Q | _K |

=====
Menu :
1. Input sendiri kartu yang diinginkan
2. Me-generate kartu secara otomatis
=====
Pilih pilihanmu (1/2) : 1
=====
Masukkan kartu berupa (A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K)
Masukkan kartu : 6 6 4 J
=====
Kartu anda adalah : 6 6 4 J
=====
Total solusi ada sebanyak 0
=====
Berikut adalah solusinya :
Total waktu adalah : 0.001 detik
=====
Apakah anda ingin menyimpan solusi sebagai file? (y/n) : y
Masukan nama file yang diinginkan (tanpa .txt): contoh1
File berhasil disimpan
=====
Apakah anda ingin mencoba angka lain?(y/n) : 
```

B. Contoh 2 (User Input, Solusi sedikit, File txt)

```

=====
WELCOME TO
SUDOKU GAME
=====
      K ^
      ^ ^ ^
      ^ ^ ^
      ^ ^ ^
      ^ ^ ^
      K
      Q ^
      ^ ^ ^
      ^ ^ ^
      ^ ^ ^
      Q
      J ^
      ^ ^ ^
      ^ ^ ^
      ^ ^ ^
      J
      10 ^
      ^ ^ ^
      ^ ^ ^
      ^ ^ ^
      10
      A ^
      ^ ^ ^
      ^ ^ ^
      ^ ^ ^
      A
=====
Menu :
1. Input sendiri kartu yang diinginkan
2. Me-generate kartu secara otomatis
=====
Pilih pilihanmu (1/2) : 1
=====
Masukkan kartu berupa (A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K)
Masukkan kartu : 5 7 7 J
=====
Kartu anda adalah : 5 7 7 J
=====
Total solusi ada sebanyak 2
=====
Berikut adalah solusinya :
(5-(11/7))*7
7*(5-(11/7))
Total waktu adalah : 0.001 detik
=====
Apakah anda ingin menyimpan solusi sebagai file? (y/n) : y
Masukan nama file yang diinginkan (tanpa .txt): contoh2
File berhasil disimpan
=====
Apakah anda ingin mencoba angka lain?(y/n) : 

```



C. Contoh 3 (User Input, Solusi banyak, File txt)

```

WELCOME TO
24 GAME

10 ^ | J ^ | Q ^ | K ^ | A ^ |
^ ^ ^ | ^ ^ ^ | ^ ^ ^ | ^ ^ ^ | ^ ^ ^ |
^ ^ ^ | ^ ^ ^ | ^ ^ ^ | ^ ^ ^ | ^ ^ ^ |
^ ^ ^ | ^ ^ ^ | ^ ^ ^ | ^ ^ ^ | ^ ^ ^ |
_ 10 | _ J | _ Q | _ K | _ A |

=====
Menu :
1. Input sendiri kartu yang diinginkan
2. Me-generate kartu secara otomatis
=====
Pilih pilihanmu (1/2) : 1
=====
Masukkan kartu berupa (A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K)
Masukkan kartu : A 2 3 4
=====
Kartu anda adalah : A 2 3 4
=====
Total solusi ada sebanyak 242
=====
Berikut adalah solusinya :
((1+2)+3)*4
((1*2)*3)*4
(1+(2+3))*4
(1*(2*3))*4
1*((2*3)*4)
1*(2*(3*4))
(1*2)*(3*4)
((1*2)*4)*3

```

$(4*2)*(1*3)$
 $(4*2)/(1/3)$
 $((4*2)*3)*1$
 $((4*2)*3)/1$
 $(4*(2*3))*1$
 $(4*(2*3))/1$
 $4*((2+3)+1)$
 $4*((2*3)*1)$
 $4*((2*3)/1)$
 $4*(2+(3+1))$
 $4*(2*(3*1))$
 $4*(2*(3/1))$
 $(4+2)*(3+1)$
 $(4*2)*(3*1)$
 $(4*2)*(3/1)$
 $((4*3)*1)*2$
 $((4*3)/1)*2$
 $(4*(3*1))*2$
 $(4*(3/1))*2$
 $4*((3+1)+2)$
 $4*((3*1)*2)$
 $4*((3/1)*2)$
 $4*(3+(1+2))$
 $4*(3*(1*2))$
 $4*(3/(1/2))$
 $(4*3)*(1*2)$
 $(4*3)/(1/2)$
 $((4*3)*2)*1$
 $((4*3)*2)/1$
 $(4*(3*2))*1$
 $(4*(3*2))/1$
 $4*((3+2)+1)$
 $4*((3*2)*1)$
 $4*((3*2)/1)$
 $4*(3+(2+1))$
 $4*(3*(2*1))$
 $4*(3*(2/1))$
 $(4*3)*(2*1)$
 $(4*3)*(2/1)$

Total waktu adalah : 0.083 detik

=====

Apakah anda ingin menyimpan solusi sebagai file? (y/n) : y

Masukan nama file yang diinginkan (tanpa .txt): contoh3

File berhasil disimpan

=====

Apakah anda ingin mencoba angka lain?(y/n) :

EXPLORER

TUCIL1_13521143

>

.vscode

>

bin

Tucil.exe

M

>

doc

Tucil1-Stima-2023.pdf

>

src

Ascii.txt

Tucil.cpp

>

test

contoh1.txt

contoh2.txt

contoh3.txt

U

README.md

run.bat

test > contoh3.txt

1 Kartu Anda adalah : A 2 3 4

2 =====

3 Terdapat solusi sebanyak : 242

4 Solusi:

5 $((1+2)+3)*4$

6 $((1*2)*3)*4$

7 $(1+(2+3))*4$

8 $(1*(2*3))*4$

9 $1*((2*3)*4)$

10 $1*(2*(3*4))$

11 $(1*2)*(3*4)$

12 $((1*2)*4)*3$

13 $(1*(2*4))*3$

14 $1*((2*4)*3)$

15 $1*(2*(4*3))$

16 $(1*2)*(4*3)$

17 $((1+3)+2)*4$

18 $((1*3)*2)*4$

19 $(1+(3+2))*4$

20 $(1*(3*2))*4$

21 $1*((3*2)*4)$

EXPLORER

TUCIL1_13521143

>

.vscode

>

bin

Tucil.exe

M

>

doc

Tucil1-Stima-2023.pdf

>

src

Ascii.txt

Tucil.cpp

>

test

contoh1.txt

contoh2.txt

contoh3.txt

U

README.md

run.bat

test > contoh3.txt

229 $4*((3/1)*2)$

230 $4*(3+(1+2))$

231 $4*(3*(1*2))$

232 $4*(3/(1/2))$

233 $(4*3)*(1*2)$

234 $(4*3)/(1/2)$

235 $((4*3)*2)*1$

236 $((4*3)*2)/1$

237 $(4*(3*2))*1$

238 $(4*(3*2))/1$

239 $4*((3+2)+1)$

240 $4*((3*2)*1)$

241 $4*((3*2)/1)$

242 $4*(3+(2+1))$

243 $4*(3*(2*1))$

244 $4*(3*(2/1))$

245 $(4*3)*(2*1)$

246 $(4*3)*(2/1)$

247 =====

248

D. Contoh 4 (Random input, File txt)

```

WELCOME TO
34 GAME

10 ^ | J ^ | Q ^ | K ^ | A ^ |
^ ^ ^ | ^ ^ ^ | ^ ^ ^ | ^ ^ ^ | ^ ^ ^ |
^ ^ ^ | ^ ^ ^ | ^ ^ ^ | ^ ^ ^ | ^ ^ ^ |
^ ^ ^ | ^ ^ ^ | ^ ^ ^ | ^ ^ ^ | ^ ^ ^ |
_ 10 | _ J | _ Q | _ K | _ A |

=====
Menu :
1. Input sendiri kartu yang diinginkan
2. Me-generate kartu secara otomatis
=====
Pilih pilihanmu (1/2) : 2
=====
Kartu anda adalah : Q 3 2 4
=====
Total solusi ada sebanyak 64
=====
Berikut adalah solusinya :
((12/3)+2)*4
(12-(3*2))*4
12*((3*2)-4)
12/(3/(2+4))
(12/3)*(2+4)
12/(3/(4+2))
(12/3)*(4+2)
(12-(2*3))*4

```

```
(3*(12+4))/2
3*((12+4)/2)
3/(2/(12+4))
(3/2)*(12+4)
((3*2)-4)*12
3/(2/(4+12))
(3/2)*(4+12)
(3*(4+12))/2
3*((4+12)/2)
(2+(12/3))*4
2*(12*(4-3))
2*(12/(4-3))
(2*12)*(4-3)
(2*12)/(4-3)
((2*3)-4)*12
((2+4)*12)/3
(2+4)*(12/3)
((2+4)/3)*12
(2*(4-3))*12
(2/(4-3))*12
2*((4-3)*12)
2/((4-3)/12)
(2+4)/(3/12)
((4+12)*3)/2
4*((12/3)+2)
4*(12-(3*2))
(4+12)*(3/2)
((4+12)/2)*3
4*(12-(2*3))
(4+12)/(2/3)
((4-3)*12)*2
(4-3)*(12*2)
((4-3)*2)*12
(4-3)*(2*12)
((4+2)*12)/3
4*(2+(12/3))
(4+2)*(12/3)
((4+2)/3)*12
(4+2)/(3/12)
```

Total waktu adalah : 0.021 detik

=====
Apakah anda ingin menyimpan solusi sebagai file? (y/n) : y

Masukan nama file yang diinginkan (tanpa .txt): contoh4

File berhasil disimpan

=====
Apakah anda ingin mencoba angka lain?(y/n) : ☐

EXPLORER

TUCIL1_13521143

.vscode

bin

doc

src

test

README.md

run.bat

Tucil.exe

Tucil1-Stima-2023.pdf

Ascii.txt

Tucil.cpp

contoh1.txt

contoh2.txt

contoh3.txt

contoh4.txt

test > contoh4.txt

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

Kartu Anda adalah : Q 3 2 4

=====

Terdapat solusi sebanyak : 64

Solusi:

$((12/3)+2)*4$

$(12-(3*2))*4$

$12*((3*2)-4)$

$12/(3/(2+4))$

$(12/3)*(2+4)$

$12/(3/(4+2))$

$(12/3)*(4+2)$

$(12-(2*3))*4$

$12*((2*3)-4)$

$(12*(2+4))/3$

$12*((2+4)/3)$

$12*(2*(4-3))$

$12*(2/(4-3))$

$(12*2)*(4-3)$

$(12*2)/(4-3)$

$((12+4)*3)/2$

$(12*(4-3))*2$

$(12/(4-3))*2$

$12*((4-3)*2)$

EXPLORER

TUCIL1_13521143

.vscode

bin

doc

src

test

README.md

run.bat

Tucil.exe

Tucil1-Stima-2023.pdf

Ascii.txt

Tucil.cpp

contoh1.txt

contoh2.txt

contoh3.txt

contoh4.txt

test > contoh4.txt

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

$4*((12/3)+2)$

$4*(12-(3*2))$

$(4+12)*(3/2)$

$((4+12)/2)*3$

$4*(12-(2*3))$

$(4+12)/(2/3)$

$((4-3)*12)*2$

$(4-3)*(12*2)$

$((4-3)*2)*12$

$(4-3)*(2*12)$

$((4+2)*12)/3$

$4*(2+(12/3))$

$(4+2)*(12/3)$

$((4+2)/3)*12$

$(4+2)/(3/12)$

=====

E. Contoh 5 (Random input, File txt)

```

WELCOME TO

24 GAME

10  ^  J  ^  Q  ^  K  ^
^ ^ ^  ^ ^ ^  ^ ^ ^  ^ ^ ^  ^ ^ ^
^ ^ ^  ^ ^ ^  ^ ^ ^  ^ ^ ^  ^ ^ ^
^ ^ ^  ^ ^ ^  ^ ^ ^  ^ ^ ^  ^ ^ ^
_ 10  _  _  _  _  _  _  _

=====
Menu :
1. Input sendiri kartu yang diinginkan
2. Me-generate kartu secara otomatis
=====
Pilih pilihanmu (1/2) : 2
=====
Kartu anda adalah : 2 K J 10
=====
Total solusi ada sebanyak 20
=====
Berikut adalah solusinya :
2*((13-11)+10)
2*(13-(11-10))
2*((13+10)-11)
2*(13+(10-11))
2*((10+13)-11)
2*(10+(13-11))
(2+10)*(13-11)
2*((10-11)+13)
2*(10-(11-13))
(13-11)*(2+10)
((13-11)+10)*2
(13-(11-10))*2

```

```

2*(10-(11-13))
(13-11)*(2+10)
((13-11)+10)*2
(13-(11-10))*2
(13-11)*(10+2)
((13+10)-11)*2
(13+(10-11))*2
(10+2)*(13-11)
((10+13)-11)*2
(10+(13-11))*2
((10-11)+13)*2
(10-(11-13))*2

```

Total waktu adalah : 0.007 detik

=====
Apakah anda ingin menyimpan solusi sebagai file? (y/n) : y

Masukan nama file yang diinginkan (tanpa .txt): contoh5

File berhasil disimpan

=====
Apakah anda ingin mencoba angka lain?(y/n) : ☐

```

EXPLORER
TUCIL1_13521143
  .vscode
  bin
  Tucil.exe
  doc
  Tucil1-Stima-2023.pdf
  src
  Ascii.txt
  Tucil.cpp
  test
    contoh1.txt
    contoh2.txt
    contoh3.txt
    contoh4.txt
    contoh5.txt
  README.md
  run.bat

test > contoh5.txt
1  Kartu Anda adalah : 2 K J 10
2  =====
3  Terdapat solusi sebanyak : 20
4  Solusi:
5  2*((13-11)+10)
6  2*(13-(11-10))
7  2*((13+10)-11)
8  2*(13+(10-11))
9  2*((10+13)-11)
10 2*(10+(13-11))
11 (2+10)*(13-11)
12 2*((10-11)+13)
13 2*(10-(11-13))
14 (13-11)*(2+10)
15 ((13-11)+10)*2
16 (13-(11-10))*2
17 (13-11)*(10+2)
18 ((13+10)-11)*2
19 (13+(10-11))*2
20 (10+2)*(13-11)
21 ((10+13)-11)*2
22 (10+(13-11))*2
23 ((10-11)+13)*2
24 (10-(11-13))*2
25 =====
26

```

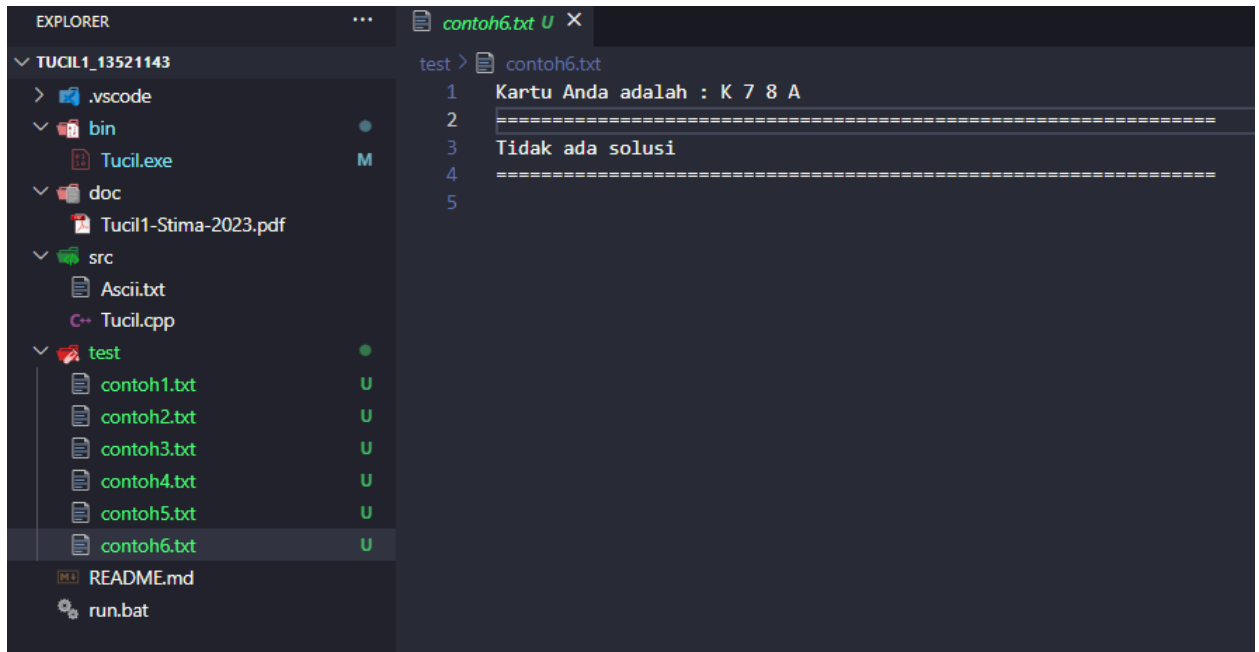
F. Contoh 6 (Random input, File txt)

```

WELCOME TO
24 GAME

10 ^ | J ^ | Q ^ | K ^ |
^ ^ ^ | ^ ^ ^ | ^ ^ ^ | ^ ^ ^ |
^ ^ ^ | ^ ^ ^ | ^ ^ ^ | ^ ^ ^ |
^ ^ ^ | ^ ^ ^ | ^ ^ ^ | ^ ^ ^ |
_10 | _J | _Q | _K |
_10 | _J | _Q | _K |

=====
Menu :
1. Input sendiri kartu yang diinginkan
2. Me-generate kartu secara otomatis
=====
Pilih pilihanmu (1/2) : 2
=====
Kartu anda adalah : K 7 8 A
=====
Total solusi ada sebanyak 0
=====
Berikut adalah solusinya :
Total waktu adalah : 0.001 detik
=====
Apakah anda ingin menyimpan solusi sebagai file? (y/n) : y
Masukan nama file yang diinginkan (tanpa .txt): contoh6
File berhasil disimpan
=====
Apakah anda ingin mencoba angka lain?(y/n) : 
```



BAB IV
LINK REPOSITORY & TABEL CEK LIST

A. Link Repository GitHub

- https://github.com/Raylouiss/Tucil1_13521143

B. Tabel Cek List

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