Laporan Tugas Kecil 1
IF2211 - Strategi Algoritma
Semester II Tahun 2022/2023

“Penyelesaian Permainan Kartu 24 dengan Algoritma Brute Force”

Dipersiapkan oleh:
Eugene Yap Jin Quan - 13521074

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

I. Deskripsi Algoritma Brute Force

Algoritma Brute Force yang digunakan adalah mengeksplorasi seluruh permutasi dari keempat kartu {A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K}, seluruh permutasi susunan operator matematika {+, -, *, /}, dan seluruh permutasi susunan operasi (atau susunan tanda kurung). Banyak permutasi dari keempat kartu adalah maksimal sebanyak 24 susunan (diperoleh dari permutasi 4P4). Banyak permutasi dari susunan operator adalah 64 susunan (diperoleh dari $4 * 4 * 4$), sedangkan banyak susunan urutan operasi adalah 5 susunan. Kelima susunan urutan operasi dituliskan sebagai berikut.

$$(A[op1]B)[op2](C[op3]D)$$

$$\big((A[op1]B)[op2]C\big)[op3]D$$

$$\big(A[op1](B[op2]C)\big)[op3]D$$

$$A[op1]((B[op2]C)[op3]D)$$

$$A[op1](B[op2](C[op3]D))$$

Keterangan: A, B, C, D adalah nilai keempat kartu berdasarkan urutan permutasi, dan [op1], [op2], [op3] adalah operator matematika yang disusun berdasarkan urutan permutasi. Operator [op1], [op2], dan [op3] masing-masing dapat bernilai +, -, * atau /.

Pada program ini, level terluar dari pencarian solusi adalah perulangan permutasi susunan keempat kartu. Pada perulangan ini, program akan mengeksplorasi sebanyak 24 permutasi dari keempat kartu. Program ini juga mencatat susunan kartu yang telah diproses, sehingga dapat melewatkan pemrosesan sebuah permutasi yang sebelumnya telah tercatat.

Pada sebuah langkah pengecekan permutasi kartu terjadi perulangan permutasi susunan operator. Perulangan ini mengeksplorasi sebanyak 64 permutasi operator yang terurut sebagai [op1][op2][op3]. Urutan operator beserta urutan susunan kartu kemudian digunakan dalam pengecekan kelima susunan urutan operasi ekspresi. Apabila sebuah ekspresi dievaluasi dan diperoleh nilai 24, ekspresi tersebut akan disimpan sebagai solusi.

Berdasarkan jumlah permutasi pada ketiga objek, total kemungkinan maksimal yang dapat ditinjau oleh algoritma adalah sebanyak 7680 kemungkinan.

II. Source Code Program (C++)

File: expr.h

```
#ifndef EXPR
#define EXPR
#include <utility>
#include <string>
#include <iostream>
```

```
using namespace std;

typedef pair<double, string> expr;

/* return expr type from int type */
expr create_expr (int number);

/* display expr */
void display_expr (expr var);

/* returns expr result, result = e1 <op> e2 */
expr evaluate (expr e1, char op, expr e2);

#endif
```

File: expr.cpp

```
#include "expr.h"

/* return expr type from int type */
expr create_expr (int number) {
    expr result;
    result.first = number;
    result.second = to_string(number);
    return result;
}

/* display expr */
void display_expr (expr var) {
    cout << "Value = " << var.first << endl;
    cout << "Expression = " << var.second << endl;
}

/* returns expr result, result = e1 <op> e2 */
expr evaluate (expr e1, char op, expr e2) {
    expr result;
    if (e1.second == "~" || e2.second == "~") {
        result.first = 0;
        result.second = "~";
    } else {
        switch (op) {
            case '+': {
                result.first = e1.first + e2.first;
                break;
            }
            case '-': {
                result.first = e1.first - e2.first;
                break;
            }
            case '*': {
                result.first = e1.first * e2.first;
                break;
            }
            case '/': {
                if (e2.first == 0) {
                    result.first = 0;
                    result.second = "~";
                    return result;
                } else {
                    result.first = e1.first / e2.first;
                }
                break;
            }
        }
        result.second = "( " + e1.second + ' ' + op + ' ' + e2.second + " )";
    }
    return result;
}
```

File: solver.h

```
#ifndef SOLVER
#define SOLVER

#include "expr.h"
#include "utils.h"
#include <vector>
#include <fstream>
#include <ctime>
#include <string>
using namespace std;

typedef vector<expr> v_expr;

/* stores solutions of 24 card game */
extern v_expr solutions;

/* save var <expr> to solutions <v_expr> if var.first is 24 and not undefined */
void save_solut(expr var);

/* prints solutions */
void display_solut();

/* arguments: perm <v_expr>; accepted argument is an ordered permutation of 4 numbers */
/* utility: find solutions to 24 game for given 4 number permutation */
void solve(v_expr perm);

/* saves solutions of 24 card game to a .txt file */
void save_to_file(int * cards, double search_time);

#endif
```

File: solver.cpp

```
#include "solver.h"

/* stores solutions of 24 card game */
v_expr solutions = {};

/* save var <expr> to solutions <v_expr> if var.first is 24 and not undefined */
void save_solut(expr var) {
    if (var.first == 24 && var.second != "~") {
        solutions.push_back(var);
    }
}

/* prints solutions */
void display_solut() {
    for (int i = 0; i < solutions.size(); i++) {
        cout << solutions[i].second << endl;
    }
}

/* arguments: perm <v_expr>; accepted argument is an ordered permutation of 4 numbers */
/* utility: find solutions to 24 game for given 4 number permutation */
void solve(v_expr perm) {
    char ops[] = {'+', '-', '*', '/'};
    char op1, op2, op3;
    for (int i = 0; i < 4; i++) {
        op1 = ops[i];
        for (int j = 0; j < 4; j++) {
            op2 = ops[j];
            for (int k = 0; k < 4; k++) {
                op3 = ops[k];
                save_solut(evaluate(evaluate(perm[0], op1, perm[1]), op2, evaluate(perm[2], op3,
perm[3])));    /* (a o b) o (c o d) */
                save_solut(evaluate(evaluate(evaluate(perm[0], op1, perm[1]), op2, perm[2]), op3,
perm[3]));    /* ((a o b) o c) o d */
```

```
                save_solut(evaluate(evaluate(perm[0], op1, evaluate(perm[1], op2, perm[2])), op3, 
perm[3]));    /* (a o (b o c)) o d */
                save_solut(evaluate(perm[0], op1, evaluate(evaluate(perm[1], op2, perm[2]), op3, 
perm[3])));    /* a o ((b o c) o d) */
                save_solut(evaluate(perm[0], op1, evaluate(perm[1], op2, evaluate(perm[2], op3, 
perm[3]))));    /* a o (b o (c o d)) */
            }
        }
    }

}

/* saves solutions of 24 card game to a .txt file */
void save_to_file(int * cards, double search_time) {
    /* set filename with date--time as serial */
    time_t curr_time;
    tm * curr_tm;
    char filename[100];
    time(&curr_time);
    curr_tm = localtime(&curr_time);
    strftime(filename, 100, "24game_%Y-%m-%d--%H-%M-%S.txt", curr_tm);

    string path(filename);
    path.insert(0, "../test/");

    ofstream solver;
    solver.open(path);

    solver << "Kartu terpilih: ";
    for (int i = 0; i < 4; i++) {
        solver << value_to_card(cards[i]) << " ";
    }
    solver << endl << endl;

    solver << "Solusi: " << endl;
    if (solutions.size() == 0) {
        solver << "Tidak ada solusi" << endl;
    } else {
        solver << "Banyak solusi: " << solutions.size() << endl;
        for (int i = 0; i < solutions.size(); i++) {
            solver << solutions[i].second << endl;
        }
    }
    solver << "Waktu pencarian: " << search_time << " detik" << endl;
    solver.close();

    cout << "File telah disimpan pada folder test (" << filename << ")" << endl;

}
```

```
File: utils.h
```

```
#ifndef UTILS
#define UTILS

#include <random>
#include <time.h>
#include <iostream>
using namespace std;

extern clock_t checkpoint;
extern double recorded_time;

/* records current clock */
void start_timer();

/* saves time elapsed since last checkpoint */
void stop_timer();
```

```
/* get card value (1-10) of input string */
int card_value(string input);

/* input 4 cards by user */
void input_cards(int * cards);

/* randomize 4 cards */
void random_cards(int * cards);

/* get cards from value */
string value_to_card(int value);

#endif
```

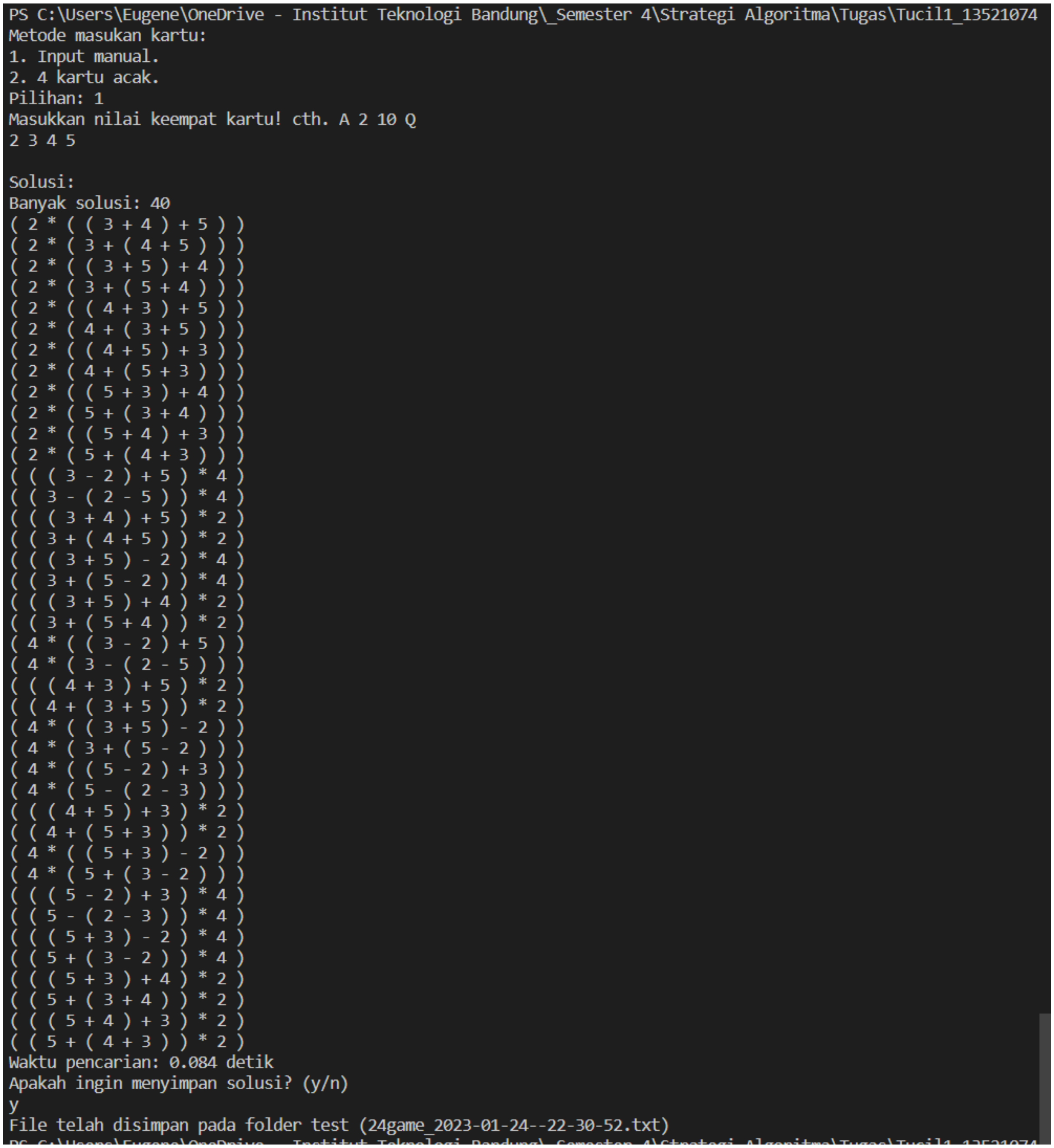
File: utils.cpp

```
#include "utils.h"

clock_t checkpoint;
double recorded_time;

/* records current clock */
void start_timer() {
    checkpoint = clock();
}

/* saves time elapsed since last checkpoint */
void stop_timer() {
    recorded_time = (double)(clock() - checkpoint) / CLOCKS_PER_SEC;
}

/* get card value (1-10) of input string */
int card_value(string input) {
    if (input == "A") {
        return 1;
    } else if (input == "10") {
        return 10;
    } else if (input == "J") {
        return 11;
    } else if (input == "Q") {
        return 12;
    } else if (input == "K") {
        return 13;
    } else if (input == "2") {
        return 2;
    } else if (input == "3") {
        return 3;
    } else if (input == "4") {
        return 4;
    } else if (input == "5") {
        return 5;
    } else if (input == "6") {
        return 6;
    } else if (input == "7") {
        return 7;
    } else if (input == "8") {
        return 8;
    } else if (input == "9") {
        return 9;
    } else {
        return -1;
    }
}

/* input 4 cards by user */
void input_cards(int * cards) {
    cout << "Masukkan nilai keempat kartu! cth. A 2 10 Q" << endl;
    string inputs[4];
    do {
        cin >> inputs[0] >> inputs[1] >> inputs[2] >> inputs[3];

        for (int i = 0; i < 4; i++) {
            cards[i] = card_value(inputs[i]);
        }

        if (cards[0] == -1 || cards[1] == -1 || cards[2] == -1 || cards[3] == -1) {
            cout << "Masukan tidak sesuai!" << endl;
        }
    } while (cards[0] == -1 || cards[1] == -1 || cards[2] == -1 || cards[3] == -1);
}

/* randomize 4 cards */
void random_cards(int * cards) {
    srand(time(NULL));
    int random;
    cout << "Kartu terpilih:" << endl;
    for (int i = 0; i < 4; i++) {
        random = rand() % 13 + 1;
        cards[i] = random;
        cout << value_to_card(random) + " ";
    }
    cout << endl;
}

/* get cards from value */
string value_to_card(int value) {
    string result;
    switch (value) {
        case 1: {
            result = "A";
            break;
        }
        case 11: {
            result = "J";
            break;
        }
        case 12: {
            result = "Q";
            break;
        }
        case 13: {
            result = "K";
            break;
        }
        default: {
            result = to_string(value);
            break;
        }
    }
    return result;
}
```

File: main.cpp

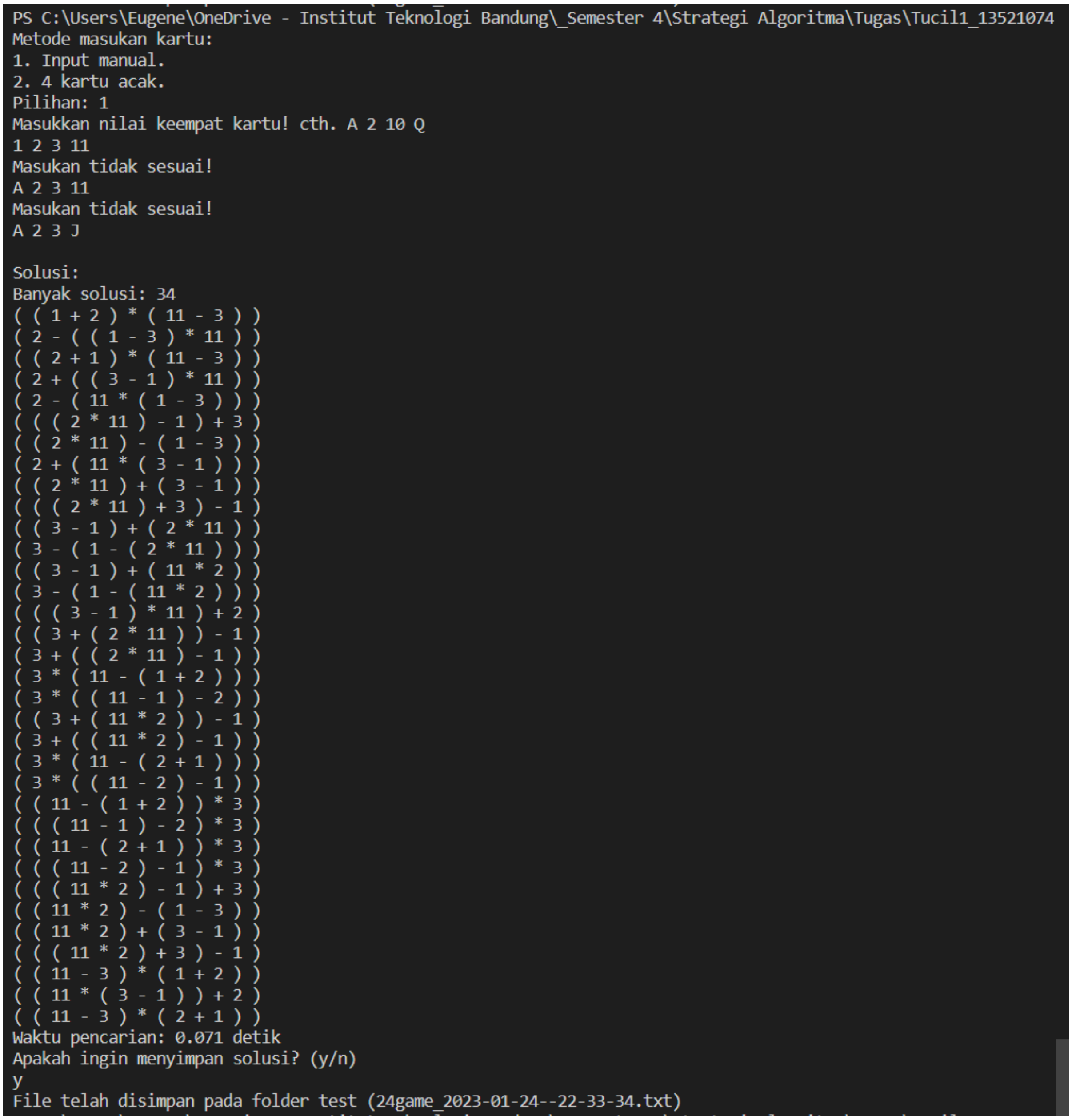
```
#include <iostream>
using namespace std;

#include "expr.h"
#include "solver.h"
#include "utils.h"

int main() {
    int card_inputs[4];

    /* input method selection & validation */
    cout << "Metode masukan kartu:\n" << "1. Input manual.\n" << "2. 4 kartu acak.\n";
    string choice;
    cout << "Pilihan: ";
    do {
        cin >> choice;
        if (choice != "1" && choice != "2") {
            cout << "Masukan tidak sesuai!" << endl;
        }
    } while (choice != "1" && choice != "2");

    /* input 4 cards */
    if (choice == "1") {
        input_cards(card_inputs);
    } else {
        random_cards(card_inputs);
    }

    /* chore: permute cards*/

    v_expr perms = {};   /* perm: abcd */

    string curr_perm;   /* current permutation */
    vector<string> processed = {};  /* processed perms */
    bool is_new_perm;

    /* solve & view solutions */
    cout << "\nSolusi:" << endl;

    start_timer();
    for (int i = 0; i < 4; i++) {       /* permutation loop */
        for (int j = 0; j < 4; j++) {
            for (int k = 0; k < 4; k++) {
                for (int l = 0; l < 4; l++) {
                    if (i != j && i != k && i != l && j != k && j != l && k != l) {     /* process if
i,j,k,l indices are unique */
                        curr_perm = value_to_card(card_inputs[i]) + value_to_card(card_inputs[j]) +
value_to_card(card_inputs[k]) + value_to_card(card_inputs[l]);

                        is_new_perm = true;         /* check if permutation is previously processed */
                        for (int a = 0; a < processed.size(); a++) {
                            if (curr_perm == processed[a]) {
                                is_new_perm = false;
                                break;
                            }
                        }

                        if (is_new_perm) {      /* process if permutation is new */
                            processed.push_back(curr_perm);

                            perms.clear();      /* create permutation v_expr */
                            perms.push_back(create_expr(card_inputs[i]));
                            perms.push_back(create_expr(card_inputs[j]));
                            perms.push_back(create_expr(card_inputs[k]));
                            perms.push_back(create_expr(card_inputs[l]));

                            solve(perms);
                        }
                    }
                }
            }
        }
    }
    stop_timer();

    int total_solut = solutions.size();

    if (total_solut > 0) {
        cout << "Banyak solusi: " << total_solut << endl;
        display_solut();

    } else {
        cout << "Tidak ada solusi" << endl;
    }

    cout << "Waktu pencarian: " << recorded_time << " detik" << endl;


    cout << "Apakah ingin menyimpan solusi? (y/n)" << endl;
    do
    {
        cin >> choice;
        if (choice != "y" && choice != "n") {
            cout << "Masukan salah!" << endl;
        }
    } while (choice != "y" && choice != "n");

    if (choice == "y") {
        save_to_file(card_inputs, recorded_time);
    }
}
```

III. Input dan Output Program



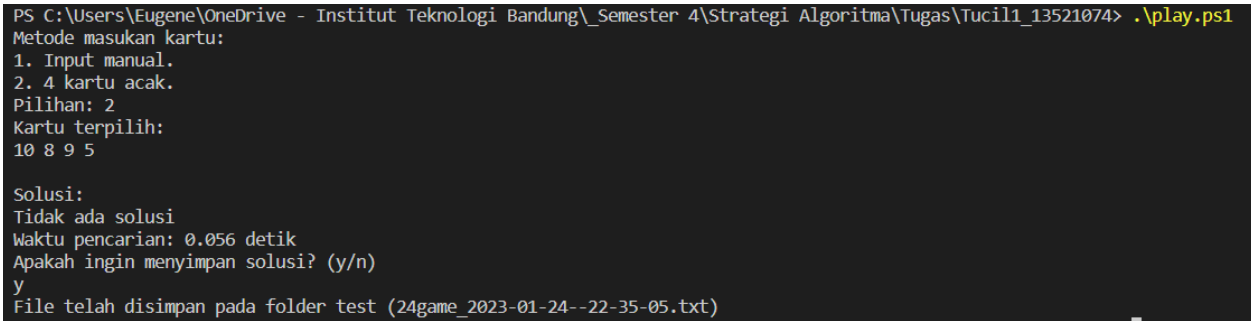
```
PS C:\Users\Eugene\OneDrive - Institut Teknologi Bandung\_Semester 4\Strategi Algoritma\Tugas\Tucil1_13521074
Metode masukan kartu:
1. Input manual.
2. 4 kartu acak.
Pilihan: 1
Masukkan nilai keempat kartu! cth. A 2 10 Q
2 3 4 5

Solusi:
Banyak solusi: 40
( 2 * ( ( 3 + 4 ) + 5 ) )
( 2 * ( 3 + ( 4 + 5 ) ) )
( 2 * ( ( 3 + 5 ) + 4 ) )
( 2 * ( 3 + ( 5 + 4 ) ) )
( 2 * ( ( 4 + 3 ) + 5 ) )
( 2 * ( 4 + ( 3 + 5 ) ) )
( 2 * ( ( 4 + 5 ) + 3 ) )
( 2 * ( 4 + ( 5 + 3 ) ) )
( 2 * ( ( 5 + 3 ) + 4 ) )
( 2 * ( 5 + ( 3 + 4 ) ) )
( 2 * ( ( 5 + 4 ) + 3 ) )
( 2 * ( 5 + ( 4 + 3 ) ) )
( ( ( 3 - 2 ) + 5 ) * 4 )
( ( 3 - ( 2 - 5 ) ) * 4 )
( ( ( 3 + 4 ) + 5 ) * 2 )
( ( 3 + ( 4 + 5 ) ) * 2 )
( ( ( 3 + 5 ) - 2 ) * 4 )
( ( 3 + ( 5 - 2 ) ) * 4 )
( ( ( 3 + 5 ) + 4 ) * 2 )
( ( 3 + ( 5 + 4 ) ) * 2 )
( 4 * ( ( 3 - 2 ) + 5 ) )
( 4 * ( 3 - ( 2 - 5 ) ) )
( ( ( 4 + 3 ) + 5 ) * 2 )
( ( 4 + ( 3 + 5 ) ) * 2 )
( 4 * ( ( 3 + 5 ) - 2 ) )
( 4 * ( 3 + ( 5 - 2 ) ) )
( 4 * ( ( 5 - 2 ) + 3 ) )
( 4 * ( 5 - ( 2 - 3 ) ) )
( ( ( 4 + 5 ) + 3 ) * 2 )
( ( 4 + ( 5 + 3 ) ) * 2 )
( 4 * ( ( 5 + 3 ) - 2 ) )
( 4 * ( 5 + ( 3 - 2 ) ) )
( ( ( 5 - 2 ) + 3 ) * 4 )
( ( 5 - ( 2 - 3 ) ) * 4 )
( ( ( 5 + 3 ) - 2 ) * 4 )
( ( 5 + ( 3 - 2 ) ) * 4 )
( ( ( 5 + 3 ) + 4 ) * 2 )
( ( 5 + ( 3 + 4 ) ) * 2 )
( ( ( 5 + 4 ) + 3 ) * 2 )
( ( 5 + ( 4 + 3 ) ) * 2 )
Waktu pencarian: 0.084 detik
Apakah ingin menyimpan solusi? (y/n)
y
File telah disimpan pada folder test (24game_2023-01-24--22-30-52.txt)
PS C:\Users\Eugene\OneDrive - Institut Teknologi Bandung\_Semester 4\Strategi Algoritma\Tugas\Tucil1_13521074
```

Gambar 1 (Input Manual: 2 3 4 5)



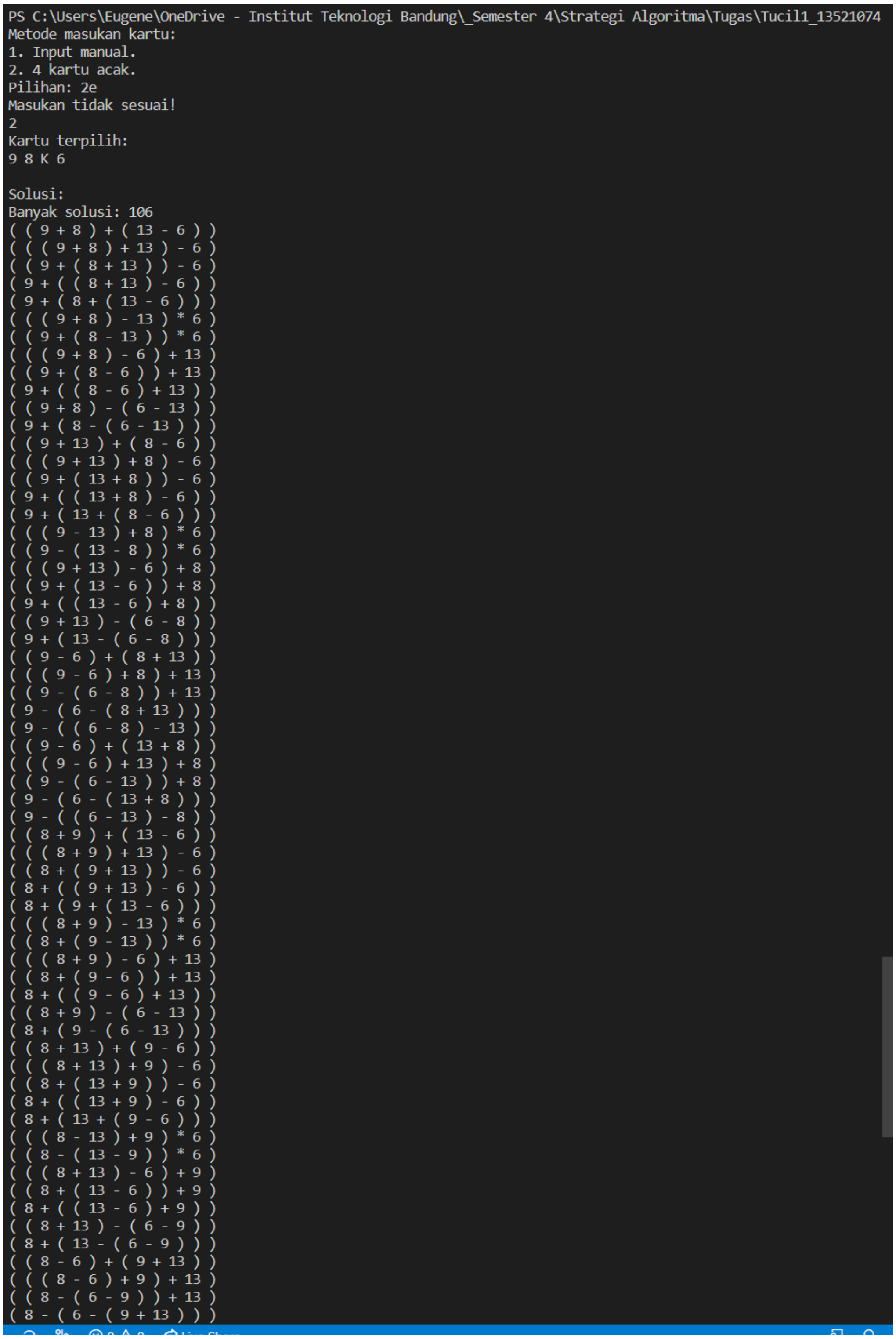
```
PS C:\Users\Eugene\OneDrive - Institut Teknologi Bandung\_Semester 4\Strategi Algoritma\Tugas\Tucil1_13521074
Metode masukan kartu:
1. Input manual.
2. 4 kartu acak.
Pilihan: 1
Masukkan nilai keempat kartu! cth. A 2 10 Q
1 2 3 11
Masukan tidak sesuai!
A 2 3 11
Masukan tidak sesuai!
A 2 3 J

Solusi:
Banyak solusi: 34
( ( 1 + 2 ) * ( 11 - 3 ) )
( 2 - ( ( 1 - 3 ) * 11 ) )
( ( 2 + 1 ) * ( 11 - 3 ) )
( 2 + ( ( 3 - 1 ) * 11 ) )
( 2 - ( 11 * ( 1 - 3 ) ) )
( ( ( 2 * 11 ) - 1 ) + 3 )
( ( 2 * 11 ) - ( 1 - 3 ) )
( 2 + ( 11 * ( 3 - 1 ) ) )
( ( 2 * 11 ) + ( 3 - 1 ) )
( ( ( 2 * 11 ) + 3 ) - 1 )
( ( 3 - 1 ) + ( 2 * 11 ) )
( 3 - ( 1 - ( 2 * 11 ) ) )
( ( 3 - 1 ) + ( 11 * 2 ) )
( 3 - ( 1 - ( 11 * 2 ) ) )
( ( ( 3 - 1 ) * 11 ) + 2 )
( ( 3 + ( 2 * 11 ) ) - 1 )
( 3 + ( ( 2 * 11 ) - 1 ) )
( 3 * ( 11 - ( 1 + 2 ) ) )
( 3 * ( ( 11 - 1 ) - 2 ) )
( ( 3 + ( 11 * 2 ) ) - 1 )
( 3 + ( ( 11 * 2 ) - 1 ) )
( 3 * ( 11 - ( 2 + 1 ) ) )
( 3 * ( ( 11 - 2 ) - 1 ) )
( ( 11 - ( 1 + 2 ) ) * 3 )
( ( ( 11 - 1 ) - 2 ) * 3 )
( ( 11 - ( 2 + 1 ) ) * 3 )
( ( ( 11 - 2 ) - 1 ) * 3 )
( ( ( 11 * 2 ) - 1 ) + 3 )
( ( 11 * 2 ) - ( 1 - 3 ) )
( ( 11 * 2 ) + ( 3 - 1 ) )
( ( ( 11 * 2 ) + 3 ) - 1 )
( ( 11 - 3 ) * ( 1 + 2 ) )
( ( 11 * ( 3 - 1 ) ) + 2 )
( ( 11 - 3 ) * ( 2 + 1 ) )
Waktu pencarian: 0.071 detik
Apakah ingin menyimpan solusi? (y/n)
y
File telah disimpan pada folder test (24game_2023-01-24--22-33-34.txt)
```

Gambar 2 (Input Manual: A 2 3 J)

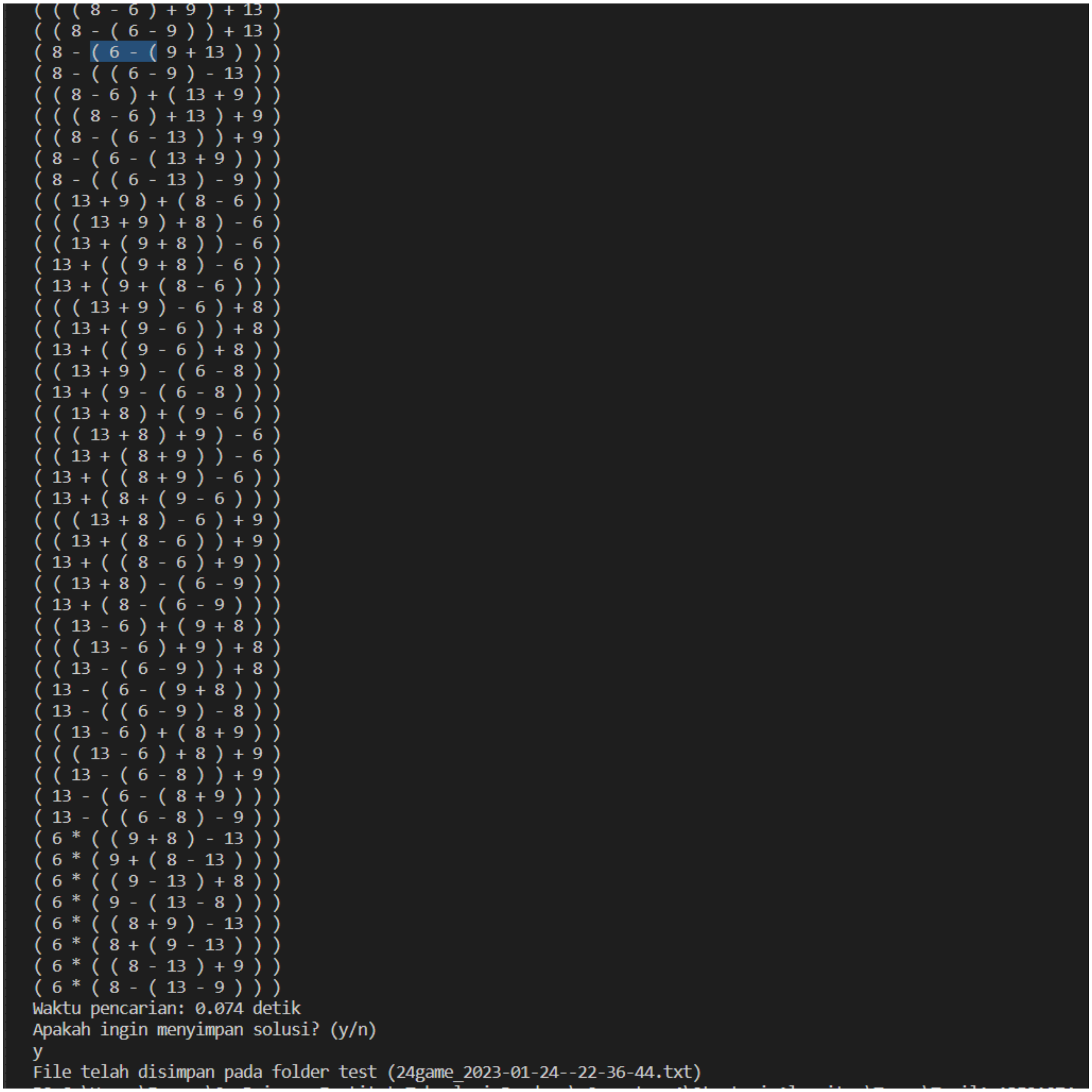


```
PS C:\Users\Eugene\OneDrive - Institut Teknologi Bandung\_Semester 4\Strategi Algoritma\Tugas\Tucil1_13521074> .\play.ps1
Metode masukan kartu:
1. Input manual.
2. 4 kartu acak.
Pilihan: 2
Kartu terpilih:
10 8 9 5

Solusi:
Tidak ada solusi
Waktu pencarian: 0.056 detik
Apakah ingin menyimpan solusi? (y/n)
y
File telah disimpan pada folder test (24game_2023-01-24--22-35-05.txt)
```

Gambar 3 (Input Random)

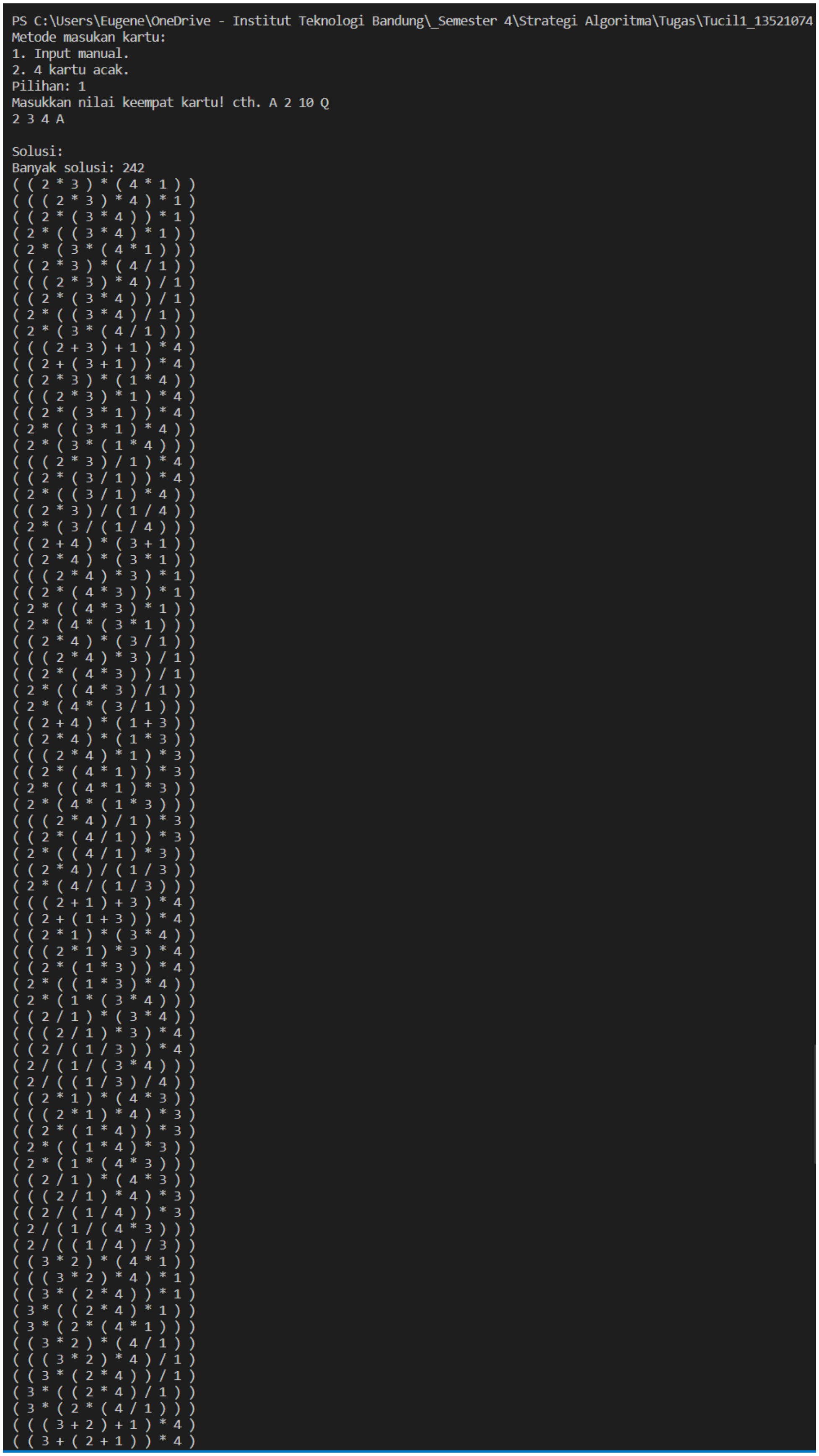


```
PS C:\Users\Eugene\OneDrive - Institut Teknologi Bandung\_Semester 4\Strategi Algoritma\Tugas\Tucil1_13521074
Metode masukan kartu:
1. Input manual.
2. 4 kartu acak.
Pilihan: 2e
Masukan tidak sesuai!
2
Kartu terpilih:
9 8 K 6

Solusi:
Banyak solusi: 106
( ( 9 + 8 ) + ( 13 - 6 ) )
( ( ( 9 + 8 ) + 13 ) - 6 )
( ( 9 + ( 8 + 13 ) ) - 6 )
( 9 + ( ( 8 + 13 ) - 6 ) )
( 9 + ( 8 + ( 13 - 6 ) ) )
( ( ( 9 + 8 ) - 13 ) * 6 )
( ( 9 + ( 8 - 13 ) ) * 6 )
( ( ( 9 + 8 ) - 6 ) + 13 )
( ( 9 + ( 8 - 6 ) ) + 13 )
( 9 + ( ( 8 - 6 ) + 13 ) )
( ( 9 + 8 ) - ( 6 - 13 ) )
( 9 + ( 8 - ( 6 - 13 ) ) )
( ( 9 + 13 ) + ( 8 - 6 ) )
( ( ( 9 + 13 ) + 8 ) - 6 )
( ( 9 + ( 13 + 8 ) ) - 6 )
( 9 + ( ( 13 + 8 ) - 6 ) )
( 9 + ( 13 + ( 8 - 6 ) ) )
( ( ( 9 - 13 ) + 8 ) * 6 )
( ( 9 - ( 13 - 8 ) ) * 6 )
( ( ( 9 + 13 ) - 6 ) + 8 )
( ( 9 + ( 13 - 6 ) ) + 8 )
( 9 + ( ( 13 - 6 ) + 8 ) )
( ( 9 + 13 ) - ( 6 - 8 ) )
( 9 + ( 13 - ( 6 - 8 ) ) )
( ( 9 - 6 ) + ( 8 + 13 ) )
( ( ( 9 - 6 ) + 8 ) + 13 )
( ( 9 - ( 6 - 8 ) ) + 13 )
( 9 - ( 6 - ( 8 + 13 ) ) )
( 9 - ( ( 6 - 8 ) - 13 ) )
( ( 9 - 6 ) + ( 13 + 8 ) )
( ( ( 9 - 6 ) + 13 ) + 8 )
( ( 9 - ( 6 - 13 ) ) + 8 )
( 9 - ( 6 - ( 13 + 8 ) ) )
( 9 - ( ( 6 - 13 ) - 8 ) )
( ( 8 + 9 ) + ( 13 - 6 ) )
( ( ( 8 + 9 ) + 13 ) - 6 )
( ( 8 + ( 9 + 13 ) ) - 6 )
( 8 + ( ( 9 + 13 ) - 6 ) )
( 8 + ( 9 + ( 13 - 6 ) ) )
( ( ( 8 + 9 ) - 13 ) * 6 )
( ( 8 + ( 9 - 13 ) ) * 6 )
( ( ( 8 + 9 ) - 6 ) + 13 )
( ( 8 + ( 9 - 6 ) ) + 13 )
( 8 + ( ( 9 - 6 ) + 13 ) )
( ( 8 + 9 ) - ( 6 - 13 ) )
( 8 + ( 9 - ( 6 - 13 ) ) )
( ( 8 + 13 ) + ( 9 - 6 ) )
( ( ( 8 + 13 ) + 9 ) - 6 )
( ( 8 + ( 13 + 9 ) ) - 6 )
( 8 + ( ( 13 + 9 ) - 6 ) )
( 8 + ( 13 + ( 9 - 6 ) ) )
( ( ( 8 - 13 ) + 9 ) * 6 )
( ( 8 - ( 13 - 9 ) ) * 6 )
( ( ( 8 + 13 ) - 6 ) + 9 )
( ( 8 + ( 13 - 6 ) ) + 9 )
( 8 + ( ( 13 - 6 ) + 9 ) )
( ( 8 + 13 ) - ( 6 - 9 ) )
( 8 + ( 13 - ( 6 - 9 ) ) )
( ( 8 - 6 ) + ( 9 + 13 ) )
( ( ( 8 - 6 ) + 9 ) + 13 )
( ( 8 - ( 6 - 9 ) ) + 13 )
( 8 - ( 6 - ( 9 + 13 ) ) )
```

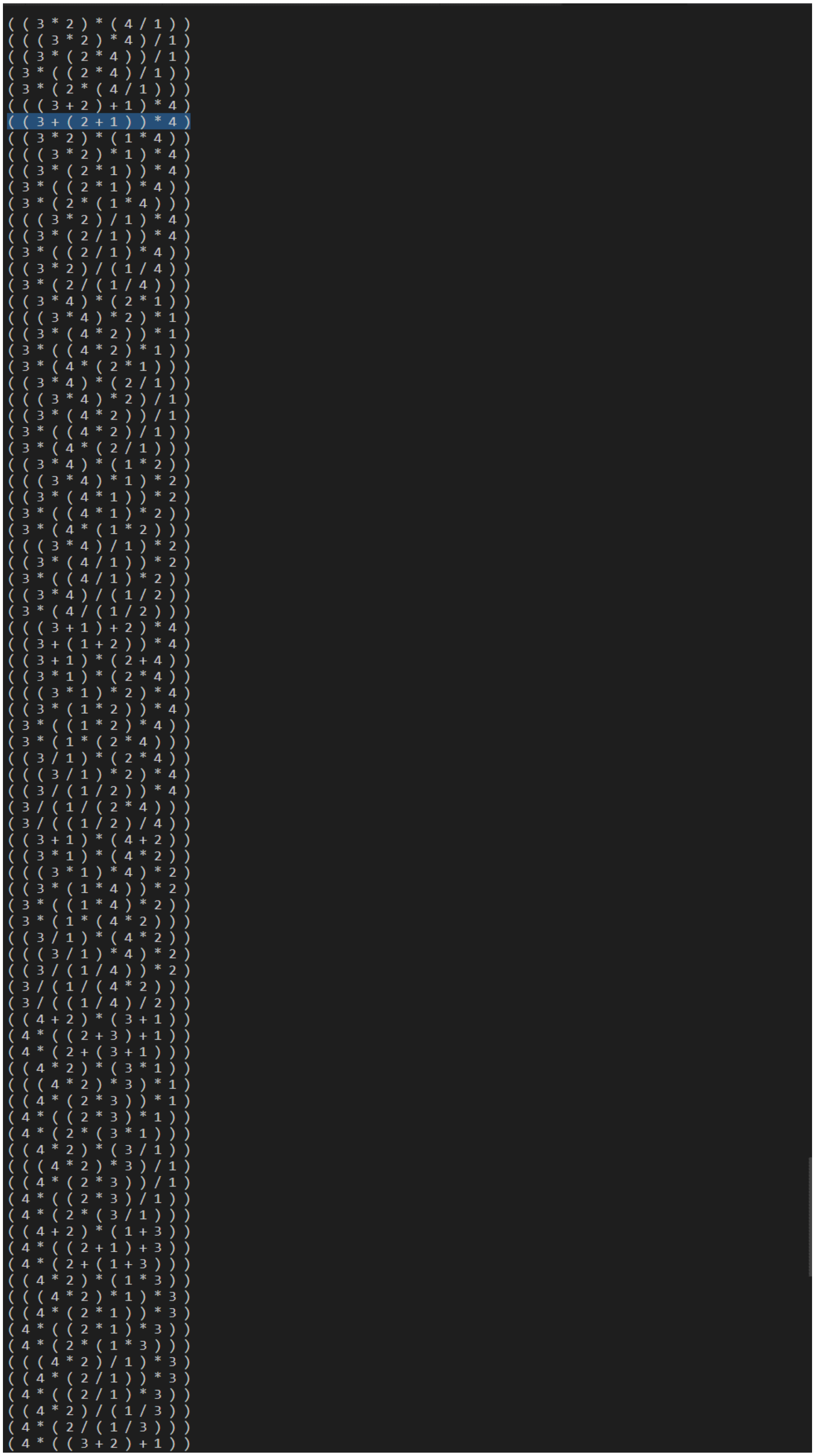


```
( ( ( 8 - 6 ) + 9 ) + 13 )
( ( 8 - ( 6 - 9 ) ) + 13 )
( 8 - ( 6 - ( 9 + 13 ) ) )
( 8 - ( ( 6 - 9 ) - 13 ) )
( ( 8 - 6 ) + ( 13 + 9 ) )
( ( ( 8 - 6 ) + 13 ) + 9 )
( ( 8 - ( 6 - 13 ) ) + 9 )
( 8 - ( 6 - ( 13 + 9 ) ) )
( 8 - ( ( 6 - 13 ) - 9 ) )
( ( 13 + 9 ) + ( 8 - 6 ) )
( ( ( 13 + 9 ) + 8 ) - 6 )
( ( 13 + ( 9 + 8 ) ) - 6 )
( 13 + ( ( 9 + 8 ) - 6 ) )
( 13 + ( 9 + ( 8 - 6 ) ) )
( ( ( 13 + 9 ) - 6 ) + 8 )
( ( 13 + ( 9 - 6 ) ) + 8 )
( 13 + ( ( 9 - 6 ) + 8 ) )
( ( 13 + 9 ) - ( 6 - 8 ) )
( 13 + ( 9 - ( 6 - 8 ) ) )
( ( 13 + 8 ) + ( 9 - 6 ) )
( ( ( 13 + 8 ) + 9 ) - 6 )
( ( 13 + ( 8 + 9 ) ) - 6 )
( 13 + ( ( 8 + 9 ) - 6 ) )
( 13 + ( 8 + ( 9 - 6 ) ) )
( ( ( 13 + 8 ) - 6 ) + 9 )
( ( 13 + ( 8 - 6 ) ) + 9 )
( 13 + ( ( 8 - 6 ) + 9 ) )
( ( 13 + 8 ) - ( 6 - 9 ) )
( 13 + ( 8 - ( 6 - 9 ) ) )
( ( 13 - 6 ) + ( 9 + 8 ) )
( ( ( 13 - 6 ) + 9 ) + 8 )
( ( 13 - ( 6 - 9 ) ) + 8 )
( 13 - ( 6 - ( 9 + 8 ) ) )
( 13 - ( ( 6 - 9 ) - 8 ) )
( ( 13 - 6 ) + ( 8 + 9 ) )
( ( ( 13 - 6 ) + 8 ) + 9 )
( ( 13 - ( 6 - 8 ) ) + 9 )
( 13 - ( 6 - ( 8 + 9 ) ) )
( 13 - ( ( 6 - 8 ) - 9 ) )
( 6 * ( ( 9 + 8 ) - 13 ) )
( 6 * ( 9 + ( 8 - 13 ) ) )
( 6 * ( ( 9 - 13 ) + 8 ) )
( 6 * ( 9 - ( 13 - 8 ) ) )
( 6 * ( ( 8 + 9 ) - 13 ) )
( 6 * ( 8 + ( 9 - 13 ) ) )
( 6 * ( ( 8 - 13 ) + 9 ) )
( 6 * ( 8 - ( 13 - 9 ) ) )
Waktu pencarian: 0.074 detik
Apakah ingin menyimpan solusi? (y/n)
y
File telah disimpan pada folder test (24game_2023-01-24--22-36-44.txt)
```

Gambar 4 (Input Random)

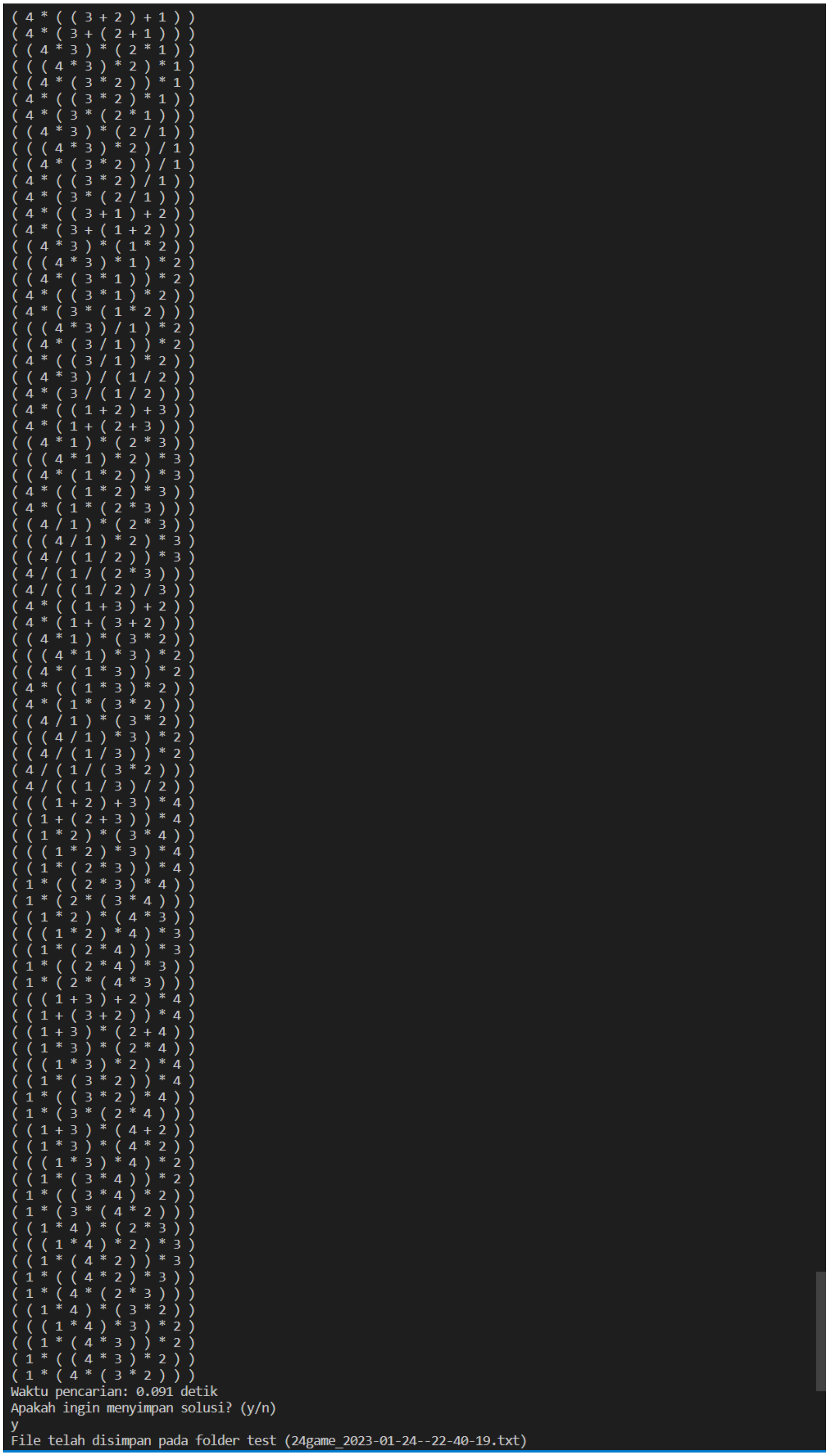


```
PS C:\Users\Eugene\OneDrive - Institut Teknologi Bandung\_Semester 4\Strategi Algoritma\Tugas\Tucil1_13521074
Metode masukan kartu:
1. Input manual.
2. 4 kartu acak.
Pilihan: 1
Masukkan nilai keempat kartu! cth. A 2 10 Q
2 3 4 A

Solusi:
Banyak solusi: 242
( ( 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 * ( 3 * 4 ) ) / 1 )
( 2 * ( ( 3 * 4 ) / 1 ) )
( 2 * ( 3 * ( 4 / 1 ) ) )
( ( ( 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 + 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 ) * ( 1 + 3 ) )
( ( 2 * 4 ) * ( 1 * 3 ) )
( ( ( 2 * 4 ) * 1 ) * 3 )
( ( 2 * ( 4 * 1 ) ) * 3 )
( 2 * ( ( 4 * 1 ) * 3 ) )
( 2 * ( 4 * ( 1 * 3 ) ) )
( ( ( 2 * 4 ) / 1 ) * 3 )
( ( 2 * ( 4 / 1 ) ) * 3 )
( 2 * ( ( 4 / 1 ) * 3 ) )
( ( 2 * 4 ) / ( 1 / 3 ) )
( 2 * ( 4 / ( 1 / 3 ) ) )
( ( ( 2 + 1 ) + 3 ) * 4 )
( ( 2 + ( 1 + 3 ) ) * 4 )
( ( 2 * 1 ) * ( 3 * 4 ) )
( ( ( 2 * 1 ) * 3 ) * 4 )
( ( 2 * ( 1 * 3 ) ) * 4 )
( 2 * ( ( 1 * 3 ) * 4 ) )
( 2 * ( 1 * ( 3 * 4 ) ) )
( ( 2 / 1 ) * ( 3 * 4 ) )
( ( ( 2 / 1 ) * 3 ) * 4 )
( ( 2 / ( 1 / 3 ) ) * 4 )
( 2 / ( 1 / ( 3 * 4 ) ) )
( 2 / ( ( 1 / 3 ) / 4 ) )
( ( 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 ) )
( ( 3 * 2 ) * ( 4 * 1 ) )
( ( ( 3 * 2 ) * 4 ) * 1 )
( ( 3 * ( 2 * 4 ) ) * 1 )
( 3 * ( ( 2 * 4 ) * 1 ) )
( 3 * ( 2 * ( 4 * 1 ) ) )
( ( 3 * 2 ) * ( 4 / 1 ) )
( ( ( 3 * 2 ) * 4 ) / 1 )
( ( 3 * ( 2 * 4 ) ) / 1 )
( 3 * ( ( 2 * 4 ) / 1 ) )
( 3 * ( 2 * ( 4 / 1 ) ) )
( ( ( 3 + 2 ) + 1 ) * 4 )
( ( 3 + ( 2 + 1 ) ) * 4 )
```



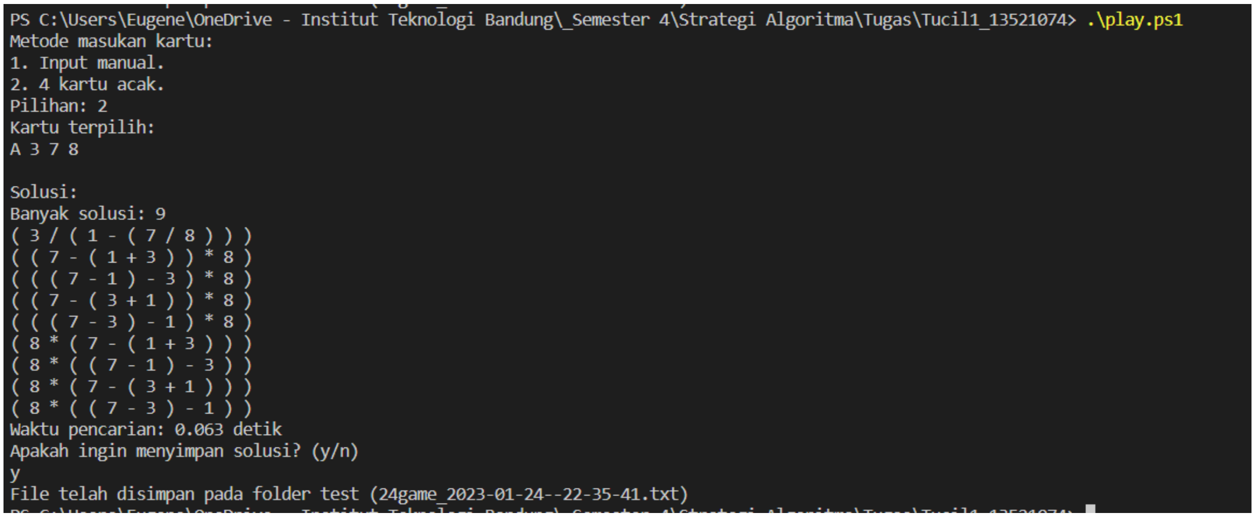
```
( ( 3 * 2 ) * ( 4 / 1 ) )
( ( ( 3 * 2 ) * 4 ) / 1 )
( ( 3 * ( 2 * 4 ) ) / 1 )
( 3 * ( ( 2 * 4 ) / 1 ) )
( 3 * ( 2 * ( 4 / 1 ) ) )
( ( ( 3 + 2 ) + 1 ) * 4 )
( ( 3 + ( 2 + 1 ) ) * 4 )
( ( 3 * 2 ) * ( 1 * 4 ) )
( ( ( 3 * 2 ) * 1 ) * 4 )
( ( 3 * ( 2 * 1 ) ) * 4 )
( 3 * ( ( 2 * 1 ) * 4 ) )
( 3 * ( 2 *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

Gambar 5 (Input Manual: 2 3 4 A)



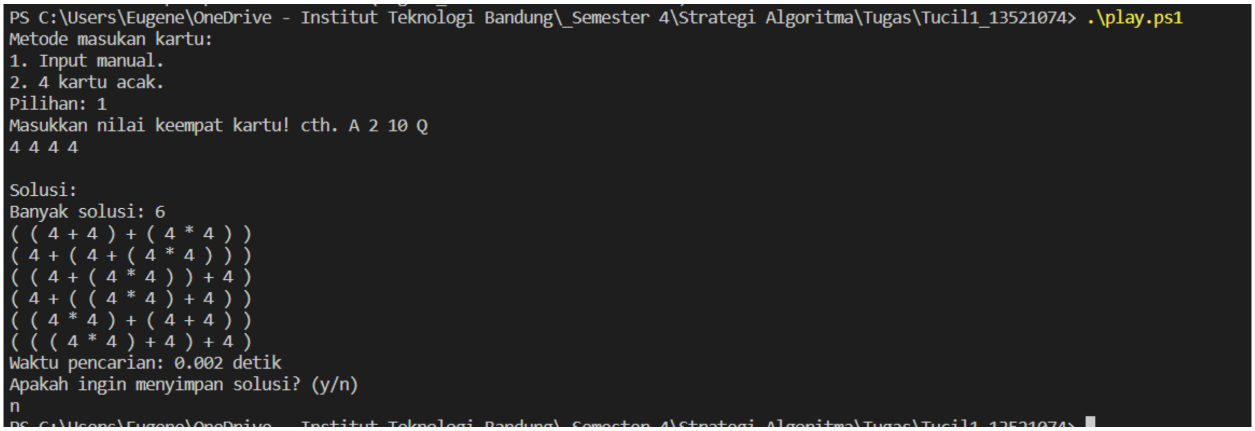
```
( 4 * ( ( 3 + 2 ) + 1 ) )
( 4 * ( 3 + ( 2 + 1 ) ) )
( ( 4 * 3 ) * ( 2 * 1 ) )
( ( ( 4 * 3 ) * 2 ) * 1 )
( ( 4 * ( 3 * 2 ) ) * 1 )
( 4 * ( ( 3 * 2 ) * 1 ) )
( 4 * ( 3 * ( 2 * 1 ) ) )
( ( 4 * 3 ) * ( 2 / 1 ) )
( ( ( 4 * 3 ) * 2 ) / 1 )
( ( 4 * ( 3 * 2 ) ) / 1 )
( 4 * ( ( 3 * 2 ) / 1 ) )
( 4 * ( 3 * ( 2 / 1 ) ) )
( 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 * ( ( 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 * 3 ) )
( ( ( 4 / 1 ) * 2 ) * 3 )
( ( 4 / ( 1 / 2 ) ) * 3 )
( 4 / ( 1 / ( 2 * 3 ) ) )
( 4 / ( ( 1 / 2 ) / 3 ) )
( 4 * ( ( 1 + 3 ) + 2 ) )
( 4 * ( 1 + ( 3 + 2 ) ) )
( ( 4 * 1 ) * ( 3 * 2 ) )
( ( ( 4 * 1 ) * 3 ) * 2 )
( ( 4 * ( 1 * 3 ) ) * 2 )
( 4 * ( ( 1 * 3 ) * 2 ) )
( 4 * ( 1 * ( 3 * 2 ) ) )
( ( 4 / 1 ) * ( 3 * 2 ) )
( ( ( 4 / 1 ) * 3 ) * 2 )
( ( 4 / ( 1 / 3 ) ) * 2 )
( 4 / ( 1 / ( 3 * 2 ) ) )
( 4 / ( ( 1 / 3 ) / 2 ) )
( ( ( 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 ) )
( ( ( 1 * 2 ) * 4 ) * 3 )
( ( 1 * ( 2 * 4 ) ) * 3 )
( 1 * ( ( 2 * 4 ) * 3 ) )
( 1 * ( 2 * ( 4 * 3 ) ) )
( ( ( 1 + 3 ) + 2 ) * 4 )
( ( 1 + ( 3 + 2 ) ) * 4 )
( ( 1 + 3 ) * ( 2 + 4 ) )
( ( 1 * 3 ) * ( 2 * 4 ) )
( ( ( 1 * 3 ) * 2 ) * 4 )
( ( 1 * ( 3 * 2 ) ) * 4 )
( 1 * ( ( 3 * 2 ) * 4 ) )
( 1 * ( 3 * ( 2 * 4 ) ) )
( ( 1 + 3 ) * ( 4 + 2 ) )
( ( 1 * 3 ) * ( 4 * 2 ) )
( ( ( 1 * 3 ) * 4 ) * 2 )
( ( 1 * ( 3 * 4 ) ) * 2 )
( 1 * ( ( 3 * 4 ) * 2 ) )
( 1 * ( 3 * ( 4 * 2 ) ) )
( ( 1 * 4 ) * ( 2 * 3 ) )
( ( ( 1 * 4 ) * 2 ) * 3 )
( ( 1 * ( 4 * 2 ) ) * 3 )
( 1 * ( ( 4 * 2 ) * 3 ) )
( 1 * ( 4 * ( 2 * 3 ) ) )
( ( 1 * 4 ) * ( 3 * 2 ) )
( ( ( 1 * 4 ) * 3 ) * 2 )
( ( 1 * ( 4 * 3 ) ) * 2 )
( 1 * ( ( 4 * 3 ) * 2 ) )
( 1 * ( 4 * ( 3 * 2 ) ) )
Waktu pencarian: 0.091 detik
Apakah ingin menyimpan solusi? (y/n)
y
File telah disimpan pada folder test (24game_2023-01-24--22-40-19.txt)
```

Gambar 5 (Input Manual: 2 3 4 A)



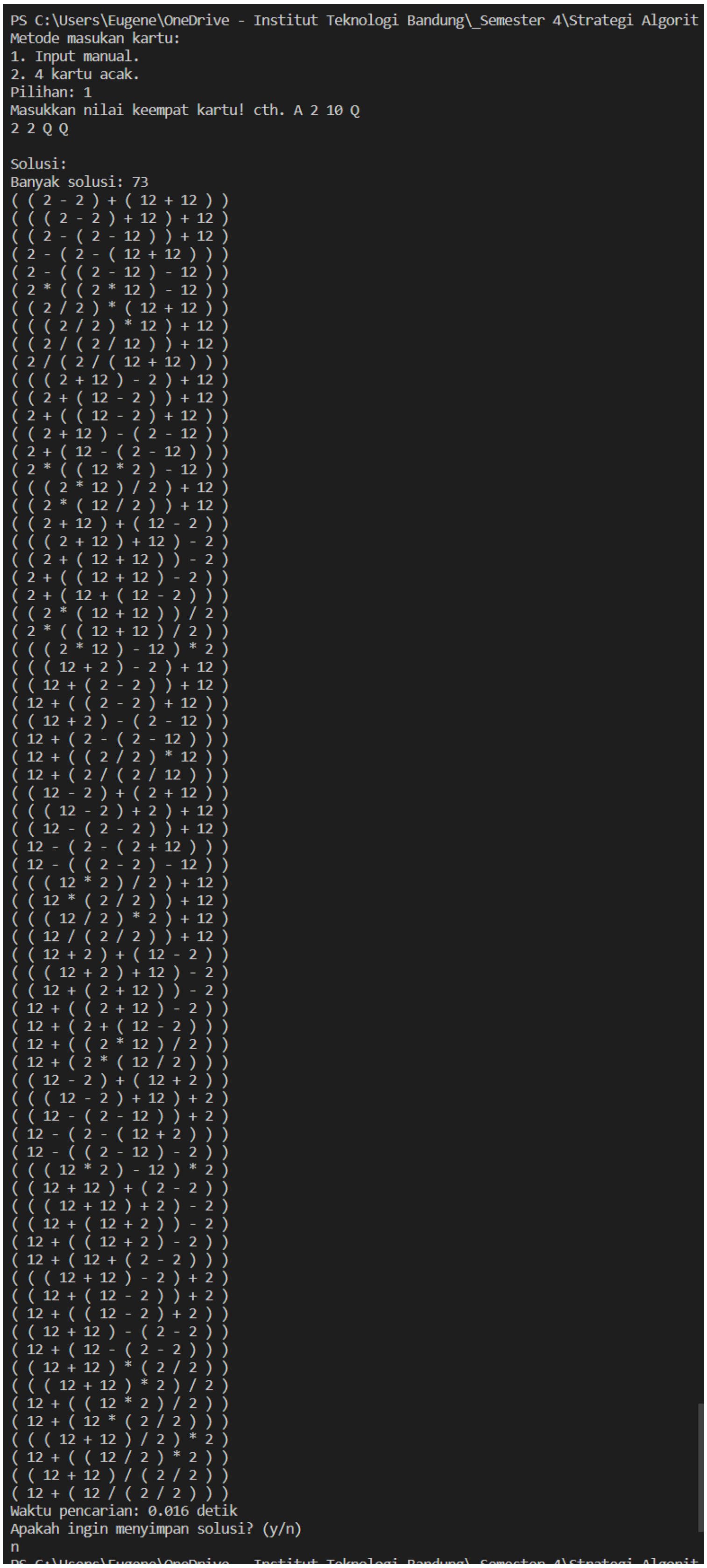
```
PS C:\Users\Eugene\OneDrive - Institut Teknologi Bandung\_Semester 4\Strategi Algoritma\Tugas\Tucil1_13521074> .\play.ps1
Metode masukan kartu:
1. Input manual.
2. 4 kartu acak.
Pilihan: 2
Kartu terpilih:
A 3 7 8

Solusi:
Banyak solusi: 9
( 3 / ( 1 - ( 7 / 8 ) ) )
( ( 7 - ( 1 + 3 ) ) * 8 )
( ( ( 7 - 1 ) - 3 ) * 8 )
( ( 7 - ( 3 + 1 ) ) * 8 )
( ( ( 7 - 3 ) - 1 ) * 8 )
( 8 * ( 7 - ( 1 + 3 ) ) )
( 8 * ( ( 7 - 1 ) - 3 ) )
( 8 * ( 7 - ( 3 + 1 ) ) )
( 8 * ( ( 7 - 3 ) - 1 ) )
Waktu pencarian: 0.063 detik
Apakah ingin menyimpan solusi? (y/n)
y
File telah disimpan pada folder test (24game_2023-01-24--22-35-41.txt)
PS C:\Users\Eugene\OneDrive - Institut Teknologi Bandung\_Semester 4\Strategi Algoritma\Tugas\Tucil1_13521074>
```

Gambar 6 (Input Random)



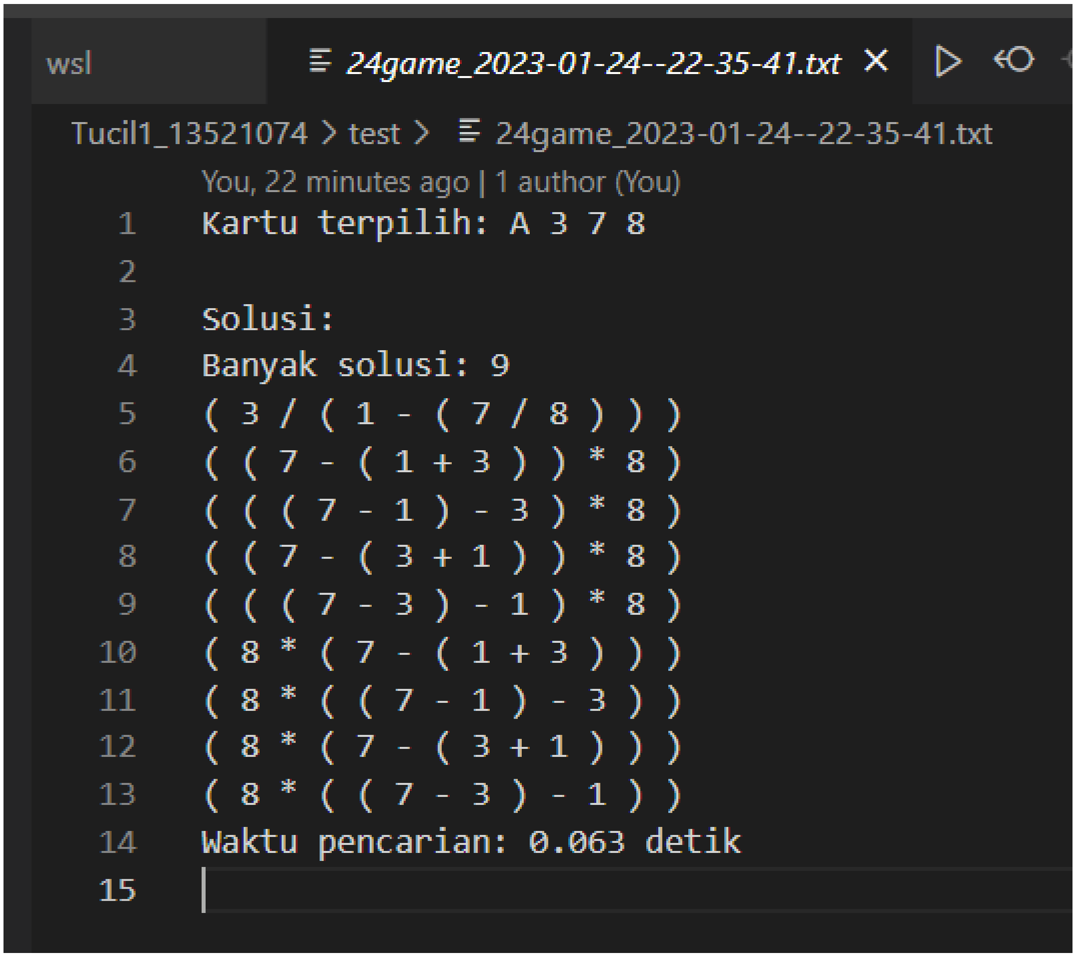
```
PS C:\Users\Eugene\OneDrive - Institut Teknologi Bandung\_Semester 4\Strategi Algoritma\Tugas\Tucil1_13521074> .\play.ps1
Metode masukan kartu:
1. Input manual.
2. 4 kartu acak.
Pilihan: 1
Masukkan nilai keempat kartu! cth. A 2 10 Q
4 4 4 4

Solusi:
Banyak solusi: 6
( ( 4 + 4 ) + ( 4 * 4 ) )
( 4 + ( 4 + ( 4 * 4 ) ) )
( ( 4 + ( 4 * 4 ) ) + 4 )
( 4 + ( ( 4 * 4 ) + 4 ) )
( ( 4 * 4 ) + ( 4 + 4 ) )
( ( ( 4 * 4 ) + 4 ) + 4 )
Waktu pencarian: 0.002 detik
Apakah ingin menyimpan solusi? (y/n)
n
PS C:\Users\Eugene\OneDrive - Institut Teknologi Bandung\_Semester 4\Strategi Algoritma\Tugas\Tucil1_13521074>
```

Gambar 7 (Input Manual: 4 4 4 4; Tidak Output Teks)



```
PS C:\Users\Eugene\OneDrive - Institut Teknologi Bandung\_Semester 4\Strategi Algorit
Metode masukan kartu:
1. Input manual.
2. 4 kartu acak.
Pilihan: 1
Masukkan nilai keempat kartu! cth. A 2 10 Q
2 2 Q Q

Solusi:
Banyak solusi: 73
( ( 2 - 2 ) + ( 12 + 12 ) )
( ( ( 2 - 2 ) + 12 ) + 12 )
( ( 2 - ( 2 - 12 ) ) + 12 )
( 2 - ( 2 - ( 12 + 12 ) ) )
( 2 - ( ( 2 - 12 ) - 12 ) )
( 2 * ( ( 2 * 12 ) - 12 ) )
( ( 2 / 2 ) * ( 12 + 12 ) )
( ( ( 2 / 2 ) * 12 ) + 12 )
( ( 2 / ( 2 / 12 ) ) + 12 )
( 2 / ( 2 / ( 12 + 12 ) ) )
( ( ( 2 + 12 ) - 2 ) + 12 )
( ( 2 + ( 12 - 2 ) ) + 12 )
( 2 + ( ( 12 - 2 ) + 12 ) )
( ( 2 + 12 ) - ( 2 - 12 ) )
( 2 + ( 12 - ( 2 - 12 ) ) )
( 2 * ( ( 12 * 2 ) - 12 ) )
( ( ( 2 * 12 ) / 2 ) + 12 )
( ( 2 * ( 12 / 2 ) ) + 12 )
( ( 2 + 12 ) + ( 12 - 2 ) )
( ( ( 2 + 12 ) + 12 ) - 2 )
( ( 2 + ( 12 + 12 ) ) - 2 )
( 2 + ( ( 12 + 12 ) - 2 ) )
( 2 + ( 12 + ( 12 - 2 ) ) )
( ( 2 * ( 12 + 12 ) ) / 2 )
( 2 * ( ( 12 + 12 ) / 2 ) )
( ( ( 2 * 12 ) - 12 ) * 2 )
( ( ( 12 + 2 ) - 2 ) + 12 )
( ( 12 + ( 2 - 2 ) ) + 12 )
( 12 + ( ( 2 - 2 ) + 12 ) )
( ( 12 + 2 ) - ( 2 - 12 ) )
( 12 + ( 2 - ( 2 - 12 ) ) )
( 12 + ( ( 2 / 2 ) * 12 ) )
( 12 + ( 2 / ( 2 / 12 ) ) )
( ( 12 - 2 ) + ( 2 + 12 ) )
( ( ( 12 - 2 ) + 2 ) + 12 )
( ( 12 - ( 2 - 2 ) ) + 12 )
( 12 - ( 2 - ( 2 + 12 ) ) )
( 12 - ( ( 2 - 2 ) - 12 ) )
( ( ( 12 * 2 ) / 2 ) + 12 )
( ( 12 * ( 2 / 2 ) ) + 12 )
( ( ( 12 / 2 ) * 2 ) + 12 )
( ( 12 / ( 2 / 2 ) ) + 12 )
( ( 12 + 2 ) + ( 12 - 2 ) )
( ( ( 12 + 2 ) + 12 ) - 2 )
( ( 12 + ( 2 + 12 ) ) - 2 )
( 12 + ( ( 2 + 12 ) - 2 ) )
( 12 + ( 2 + ( 12 - 2 ) ) )
( 12 + ( ( 2 * 12 ) / 2 ) )
( 12 + ( 2 * ( 12 / 2 ) ) )
( ( 12 - 2 ) + ( 12 + 2 ) )
( ( ( 12 - 2 ) + 12 ) + 2 )
( ( 12 - ( 2 - 12 ) ) + 2 )
( 12 - ( 2 - ( 12 + 2 ) ) )
( 12 - ( ( 2 - 12 ) - 2 ) )
( ( ( 12 * 2 ) - 12 ) * 2 )
( ( 12 + 12 ) + ( 2 - 2 ) )
( ( ( 12 + 12 ) + 2 ) - 2 )
( ( 12 + ( 12 + 2 ) ) - 2 )
( 12 + ( ( 12 + 2 ) - 2 ) )
( 12 + ( 12 + ( 2 - 2 ) ) )
( ( ( 12 + 12 ) - 2 ) + 2 )
( ( 12 + ( 12 - 2 ) ) + 2 )
( 12 + ( ( 12 - 2 ) + 2 ) )
( ( 12 + 12 ) - ( 2 - 2 ) )
( 12 + ( 12 - ( 2 - 2 ) ) )
( ( 12 + 12 ) * ( 2 / 2 ) )
( ( ( 12 + 12 ) * 2 ) / 2 )
( 12 + ( ( 12 * 2 ) / 2 ) )
( 12 + ( 12 * ( 2 / 2 ) ) )
( ( ( 12 + 12 ) / 2 ) * 2 )
( 12 + ( ( 12 / 2 ) * 2 ) )
( ( 12 + 12 ) / ( 2 / 2 ) )
( 12 + ( 12 / ( 2 / 2 ) ) )
Waktu pencarian: 0.016 detik
Apakah ingin menyimpan solusi? (y/n)
n
PS C:\Users\Eugene\OneDrive - Institut Teknologi Bandung\_Semester 4\Strategi Algorit
```

Gambar 8 (Input Manual: 2 2 Q Q; Tidak Output Teks)


wsl
24game_2023-01-24--22-35-41.txt
Tucil1_13521074 > test > 24game_2023-01-24--22-35-41.txt
You, 22 minutes ago | 1 author (You)

```
Kartu terpilih: A 3 7 8

Solusi:
Banyak solusi: 9
( 3 / ( 1 - ( 7 / 8 ) ) )
( ( 7 - ( 1 + 3 ) ) * 8 )
( ( ( 7 - 1 ) - 3 ) * 8 )
( ( 7 - ( 3 + 1 ) ) * 8 )
( ( ( 7 - 3 ) - 1 ) * 8 )
( 8 * ( 7 - ( 1 + 3 ) ) )
( 8 * ( ( 7 - 1 ) - 3 ) )
( 8 * ( 7 - ( 3 + 1 ) ) )
( 8 * ( ( 7 - 3 ) - 1 ) )
Waktu pencarian: 0.063 detik

```

Gambar 9 (Contoh Tampilan File Teks dari Pengujian pada Gambar 6)

IV. Checklist

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

V. Repository

Link repository GitHub: https://github.com/Eugene316/Tucil1_13521074