TUGAS KECIL 1 IF2211 STRATEGI ALGORITMA

SEMESTER II TAHUN 2022/2023

PROGRAM 24 SOLVER DENGAN ALGORITMA BRUTE FORCE

Disusun oleh:

Fahrian Afdholi 13521031

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

1. Algoritma *Brute Force*

Strategi brute force adalah suatu strategi algoritma yang memecahkan masalah dengan cara mencoba seluruh kemungkinan yang terjadi sehingga banyak sekali hal yang bisa dipecahkan oleh strategi algoritma ini dan mungkin bisa menyelesaikan seluruh masalah algoritma yang ada di dunia ini dan ada pula permasalahan yang hanya bisa diselesaikan oleh algoritma brute force saja.

Walaupun terlihat sangat bagus karena strategi ini sangat memakan banyak waktu karena cara memecahkan masalahnya yang menggunakan seluruh kemungkinan yang ada seperti contohnya jika kita ingin memecahkan password dengan 64 text dengan menggunakan algoritma brute force jika diasumsikan password tersebut dari seluruh angka dan huruf maka ada 4,011991914547630480065053387702 4e+99 kemungkinan yang harus kita cari dan sangat amat membutuhkan waktu yang sangat lama.

Program 24 solver bisa dipecahkan dengan algoritma bruteforce walaupun dengan waktu yang lumayan lama. Cara strategi ini memecahkannya adalah dengan cara mengambil keempat angka terlebih dahulu lalu strategi ini mengecek satu satu symbol dari (+,-,/,*) terhadap keempat angka tersebut satu satu, jika ada dari kemungkinan simbol yang sudah dicek adalah 24 maka simbol-simbol tersebut bersama empat angka akan disimpan ke dalam array satu satu. Setelah terismpan seluruhnya hasil akan ditampilkan pada program ada berapa banyak solusi yang dihasilkan.

2. Source Code dengan Bahasa Java

a. Tools.java

```java
import java.io.FileWriter;
import java.io.IOException;
import java.util.Random;
import java.util.Scanner;

8 usages  fchrgrib
public class Tools {
    9 usages
    private static final Scanner scan = new Scanner(System.in);
    3 usages  fchrgrib
    public static String[] ifStarOrSlash(String[] nums){
        for(int i = 0;i<nums.length;i++){
            if(nums[i].equals("*")){
                nums[i+1]=String.valueOf( d: Double.parseDouble(nums[i-1])*Double.parseDouble(nums[i+1]));
                nums[i-1]="0";
                nums[i]="+";
            }else if(nums[i].equals("/")){
                nums[i+1]=String.valueOf( d: Double.parseDouble(nums[i-1])/Double.parseDouble(nums[i+1]));
                nums[i-1]="0";
                nums[i]="+";
            }
        }
        return nums;
    }
    3 usages  fchrgrib
    public static double sumAll(String[] numsum){
        double sum = Double.parseDouble(numsum[0]);
        for(int i=0;i<numsum.length;i++){
            if(numsum[i].equals("+")){
                sum+=Double.parseDouble(numsum[i+1]);
            }else if(numsum[i].equals("-")){
                sum-=Double.parseDouble(numsum[i+1]);
            }
```

```java
1 usage  fchrgrib
public static boolean validationSave(){
    System.out.println("Apakah anda ingin menyimpan data tersebut?\n1.Ya\n2.Tidak");
    System.out.print("Pilih:");
    int pilih = scan.nextInt();
    while (pilih<1&&pilih>2){
        System.out.print("masukkan input yang valid: ");
        pilih= scan.nextInt();
    }
    return pilih==1;
}

4 usages  fchrgrib
private static boolean validationNumber(int num1,int num2,int num3,int num4){

    int[] num = {num1,num2,num3,num4};
    for (int i=0;i<4;i++){
        if(num[i]>13||num[i]<1){
            return false;
        }
    }
    for(int i=0;i<4;i++){
        for (int j=0;j<4;j++){
            if(i!=j){
                if((num[i]==num[j])){
                    return false;
                }
            }
        }
    }

    return true;
```

```
1 usage  fchrgrib
public static boolean isKeyboard(){
    System.out.println("Pilih Cara Bermain:\n1. Input dengan keyboard\n2. Dipilih angka random");
    System.out.print("Pilih: ");
    int pilih = scan.nextInt();
    while (pilih<1&&pilih>2){
        System.out.println("masukkan input key yang valid:");
        pilih = scan.nextInt();
    }
    return pilih==1;
}
1 usage  fchrgrib
public static boolean validationExit(){
    System.out.println("Apakah anda ingin melanjutkan permainan?\n1.Ya\n2.Tidak");
    System.out.print("Pilih: ");
    int pilih = scan.nextInt();
    while (pilih<1&&pilih>2){
        System.out.println("masukkan input yang valid:");
        pilih = scan.nextInt();
    }
    return pilih==2;
}
```

```
1 usage  fchrgrib
public static void writeResults(String res, String card) throws IOException {
    System.out.println("masukkan lokasi penyimpanan dan format nama file:\ncontoh cara menyimpan file : ./test/test.txt");
    String path = scan.next();
    FileWriter writeResult = new FileWriter(path);
    writeResult.write( str: "Kartu yang dipilih:\n"+card+"\n\n");
    if(res!="") {
        String[] result = res.split( regex: "\n");
        writeResult.write( str: result.length+" Solutions found\n\n");
        if (result.length != 0) {
            for (int i = 0; i < result.length; i++) {
                writeResult.write( str: result[i] + "\n");
            }
        }
        System.out.println("Data Berhasail Disimpan\n\n");
    }else {
        writeResult.write( str: "Tidak ada solusi\n\n");
        System.out.println("Data Berhasail Disimpan\n\n");
    }
    writeResult.close();
}
```

```java
private static int cardToNum(String card){
    if(card.equals("A")){
        return 1;
    }
    if(card.equals("J")){
        return 11;
    }
    if(card.equals("Q")){
        return 12;
    }
    if(card.equals("K")){
        return 13;
    }
    return Integer.valueOf(card);
}
```

2 usages fchrgrib

```java
private static String calculateResult(String result, int[] num) {

    for(int i=0;i<4;i++){
        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&&l!=k){
                        if(Results.results(num[i],num[j],num[k],num[l])!=""){
                            result+=Results.results(num[i],num[j],num[k],num[l]);
                        }

                    }
                }
            }
        }
    }
```

5 usages fchrgrib

```java
private static String getRandom(String[] array) {
    int rnd = new Random().nextInt(array.length);
    return array[rnd];
}
```

```java
public static String[] ifInputKeyboardOrRandom(String result,boolean isKeyboard){
    String[] resultss;
    if(isKeyboard){

        String[] card = new String[4];
        int[] num = new int[4];

        tutorCard();
        System.out.println("Masukkan kartu: ");
        for(int i=0;i<4;i++){
            card[i]=scan.next();
        }


        for(int i=0;i<4;i++){
            num[i] = cardToNum(card[i]);
        }

        boolean isNumValid = validationNumber(num[0],num[1],num[2],num[3]);

        while (isNumValid==false){
            System.out.println("kartu sama atau melebihi batas masukkan kartu yang sesuai format: ");
            for(int i=0;i<4;i++){
                card[i]=scan.next();
            }
            for(int i=0;i<4;i++){
                num[i] = cardToNum(card[i]);
            }
            isNumValid = validationNumber(num[0],num[1],num[2],num[3]);
        }
        System.out.println("\n\nKartu yang dipilih:");
        for(int i=0;i<4;i++){
```

```
        resultss = new String[]{(card[0]+" "+card[1]+" "+card[2]+" "+card[3]+" "), calculateResult(result, num)};
        return resultss;
    }else {

        String[] cards = {"A","2","3","4","5","6","7","8","9","10","J","Q","K"};
        int[] num = new int[4];


        String[] card = {getRandom(cards),getRandom(cards),getRandom(cards),getRandom(cards)};

        for(int i=0;i<4;i++){
            num[i] = cardToNum(card[i]);
        }
        boolean isNumValid = validationNumber(num[0],num[1],num[2],num[3]);
        while (!isNumValid){
            for(int i=0;i<4;i++){
                card[i]=getRandom(cards);
            }
            for(int i=0;i<4;i++){
                num[i] = cardToNum(card[i]);
            }
            isNumValid = validationNumber(num[0],num[1],num[2],num[3]);
        }

        System.out.println("\n\nKartu yang dipilih:");
        for(int i=0;i<4;i++){
            System.out.print(card[i]+" ");
        }
        System.out.println("\n");
        resultss = new String[]{(card[0]+" "+card[1]+" "+card[2]+" "+card[3]+" "), calculateResult(result, num)};
        return resultss;
    }
}
```

```
1 usage   fchrgrib
private static void tutorCard(){
    System.out.println("Contoh cara memasukkan kartu:\n A J 7 8\natau\nA\nJ\n7\n8\n");
}
```

b. SplashScreen.java

c. Priority.java

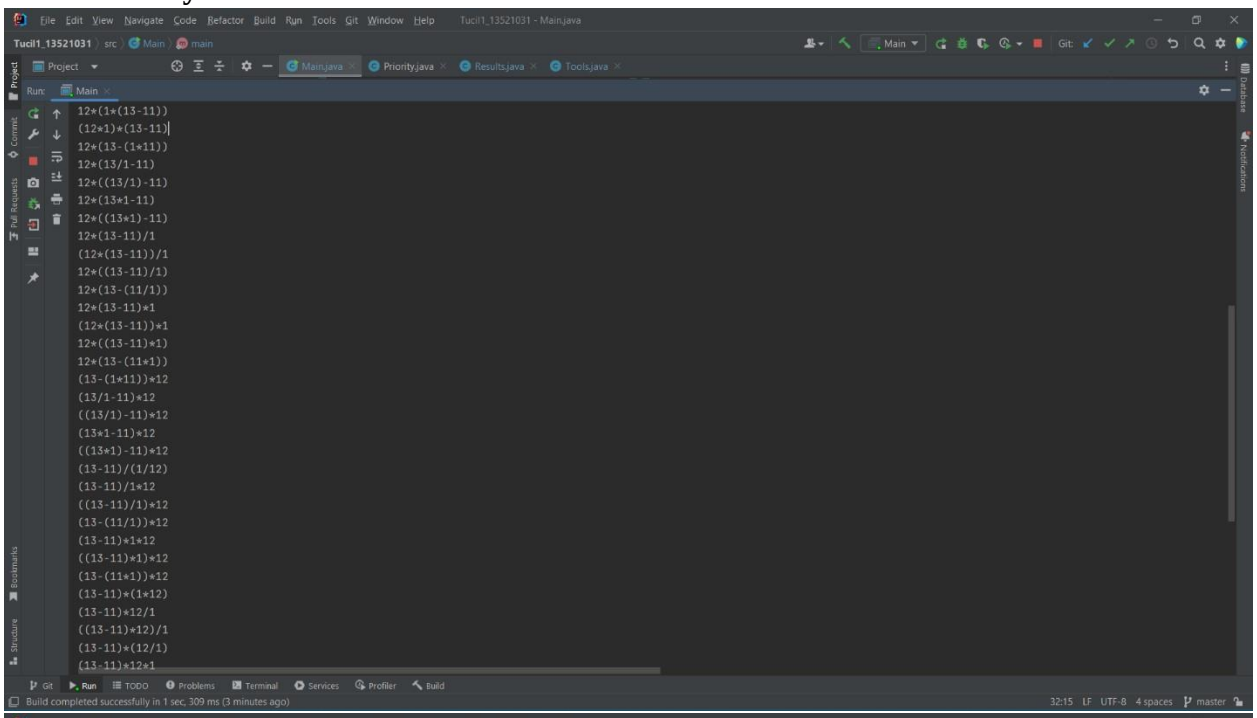
```java
3 usages  fchrgrib
public class Priority {
    6 usages
    private static final Tools tools = new Tools();

    20 usages  fchrgrib
    public static double one(String num1,String sym,String num2){
        String[] merge = {num1,sym,num2};
        return tools.sumAll(tools.ifStarOrSlash(merge));
    }
    5 usages  fchrgrib
    public static double two(String num1,String sym1,String num2,String sym2,String num3){
        String[] merge = {num1,sym1,num2,sym2,num3};
        return tools.sumAll(tools.ifStarOrSlash(merge));
    }
    1 usage  fchrgrib
    public static double three(String num1,String sym1,String num2,String sym2,String num3,String sym3,String num4){
        String[] merge = {num1,sym1,num2,sym2,num3,sym3,num4};
        return tools.sumAll(tools.ifStarOrSlash(merge));
    }

}
```

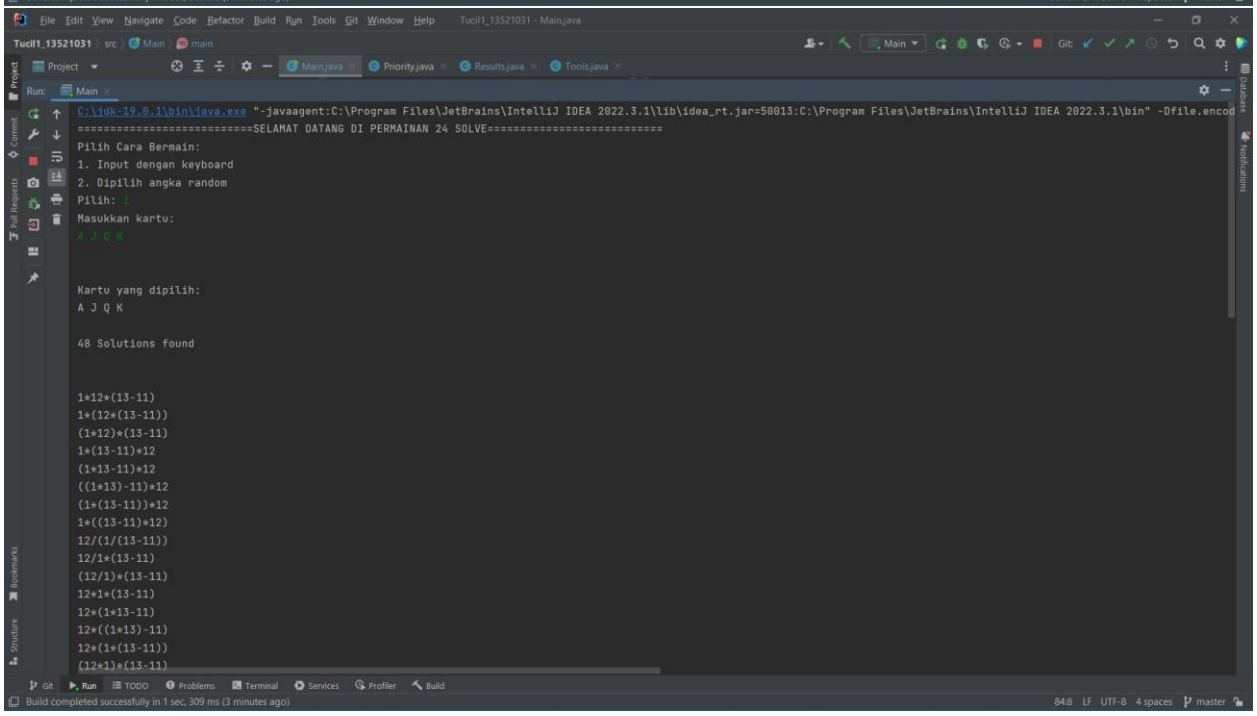
d. Main.java

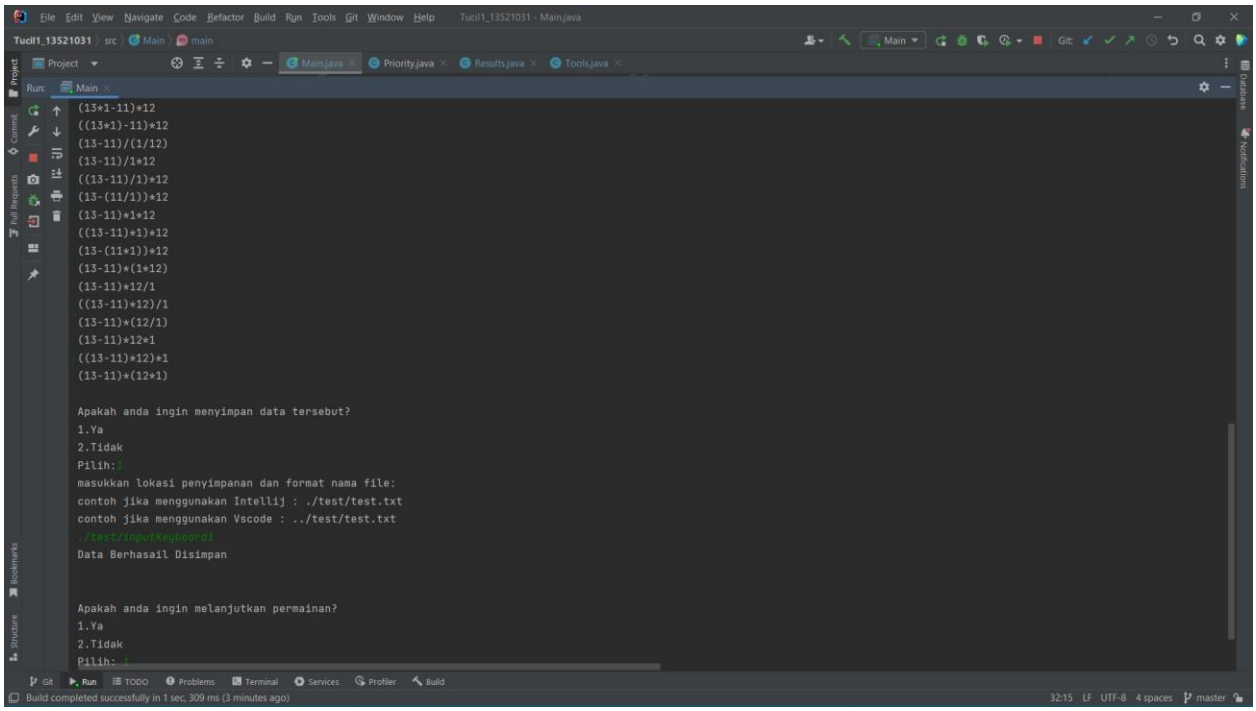
```java
import java.io.IOException;
import java.util.Scanner;

3 usages  fchrgrib
public class Main {
    no usages
    private static final Scanner scan = new Scanner(System.in);
    no usages  fchrgrib
    public static void main(String[] args) throws IOException {



        Splashscreen.splashscreen();
        boolean isEnd = false;
        while (!isEnd){
            String res = "";
            boolean isBoard = Tools.isKeyboard();
            String[] result = Tools.ifInputKeyboardOrRandom(res,isBoard);
            if(result[1]==""){
                System.out.println("Tidak ada solusi\n\n");
            }else {
                String [] cobs = result[1].split( regex: "\n");
                System.out.println(cobs.length+" Solutions found\n\n");
            }
            System.out.println(result[1]);
            boolean save= Tools.validationSave();
            if(save) Tools.writeResults(result[1],result[0]);
            isEnd = Tools.validationExit();
        }
    }
}
```

3. Screenshot I/O Program

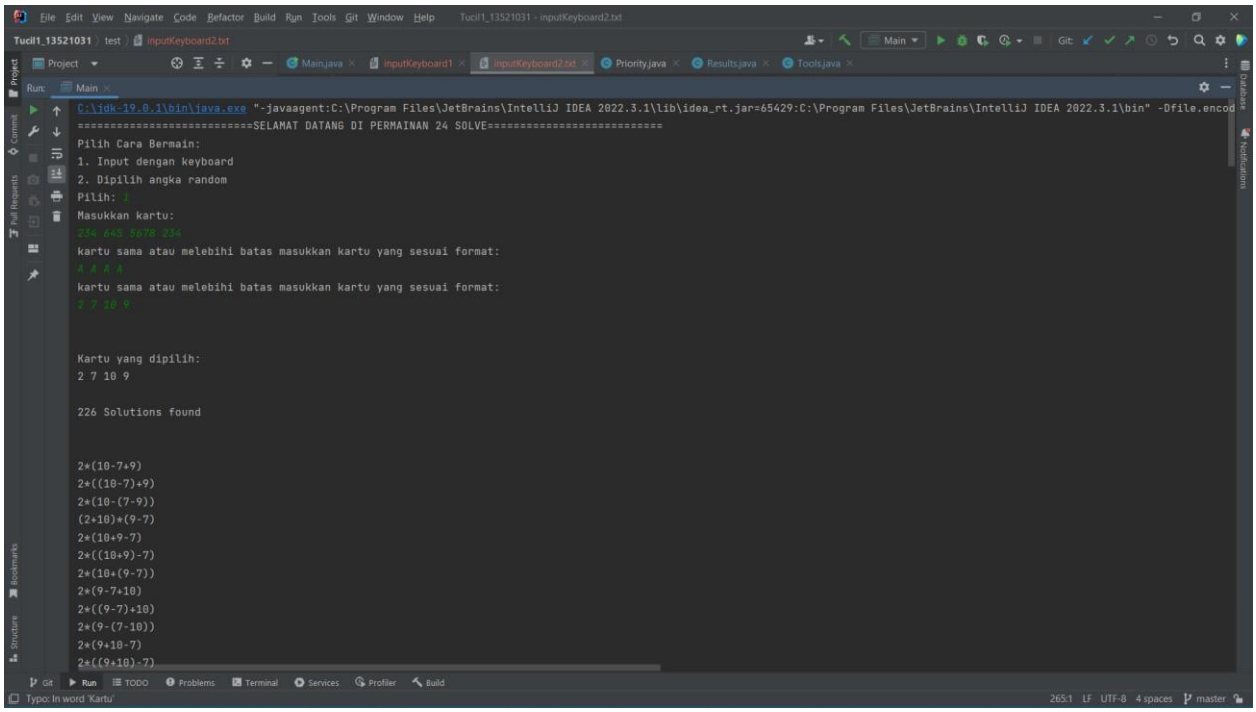
a. I/O with keyboard1

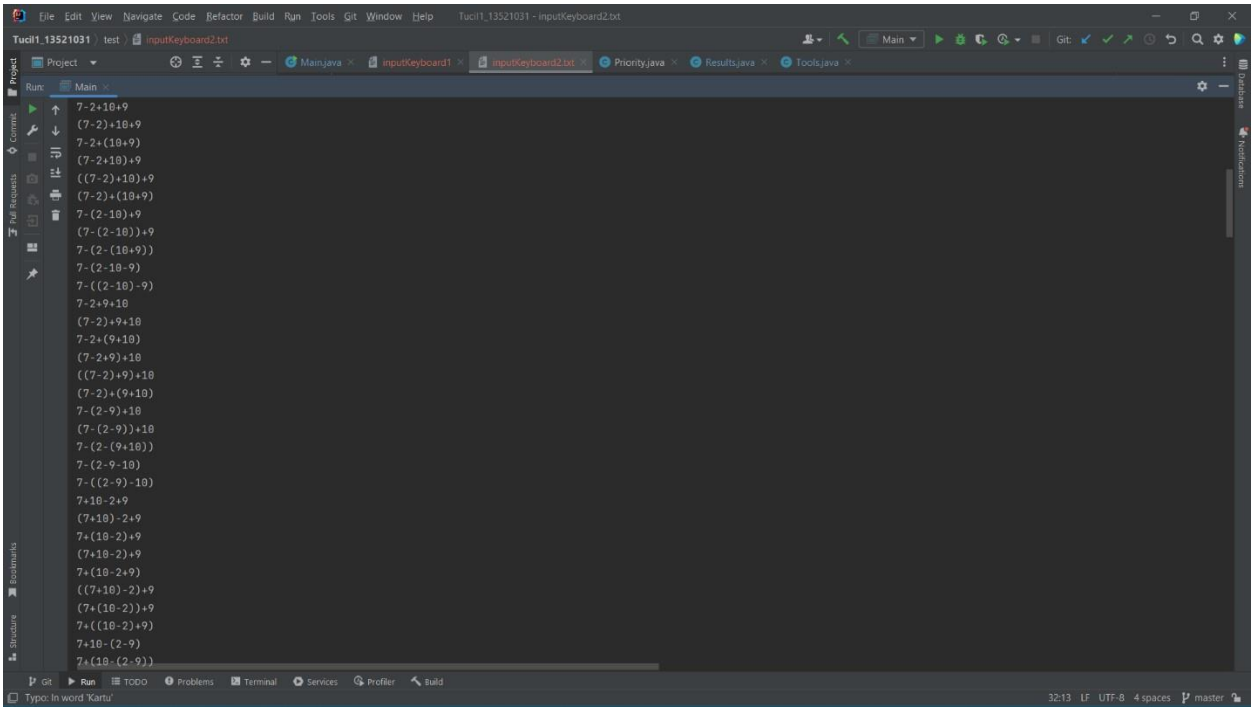


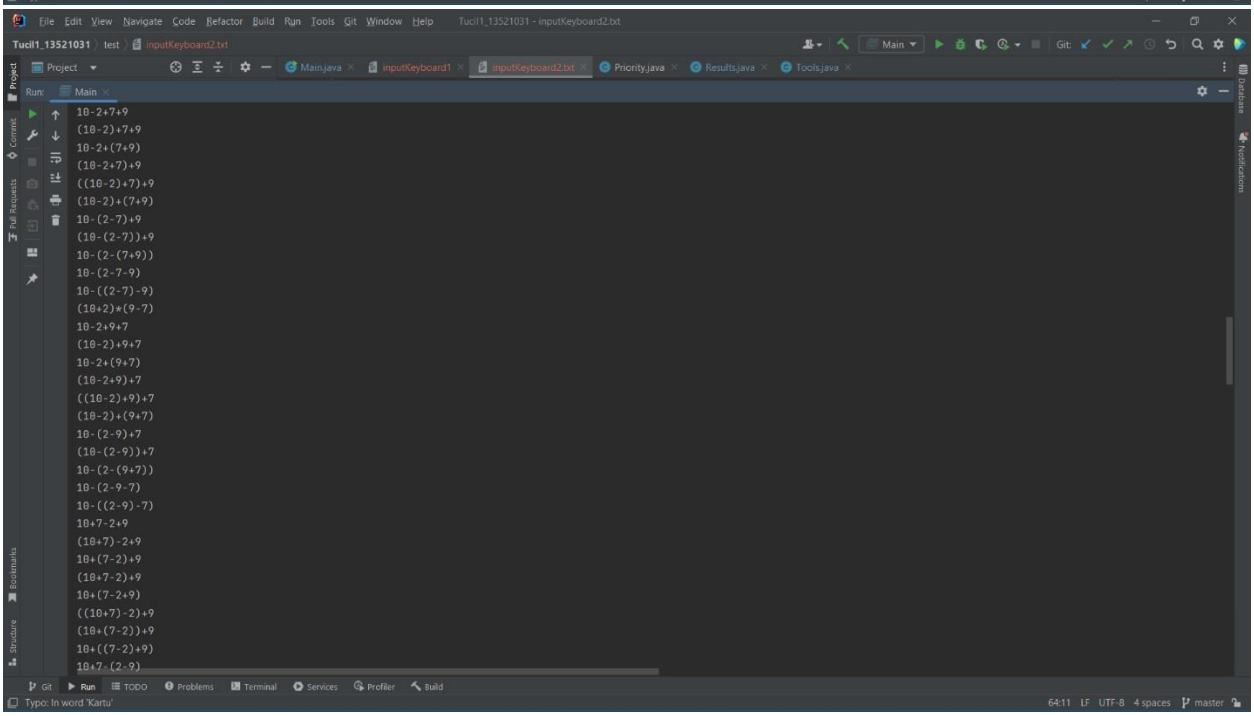


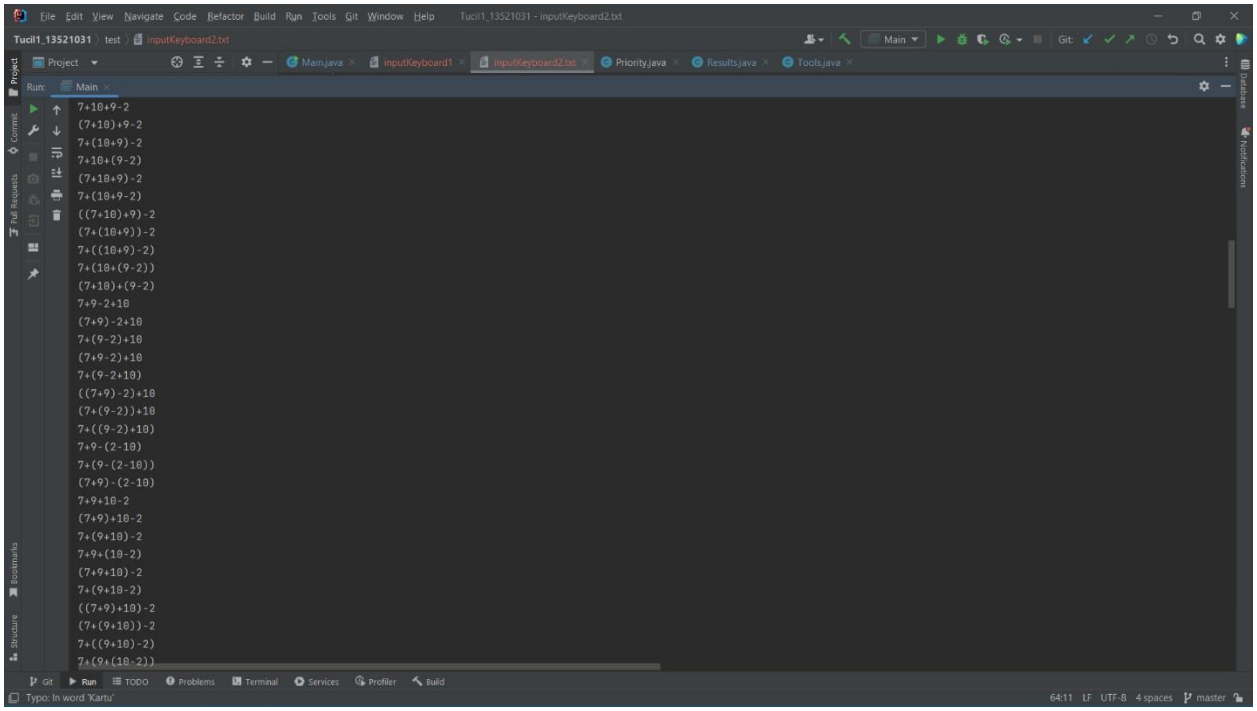


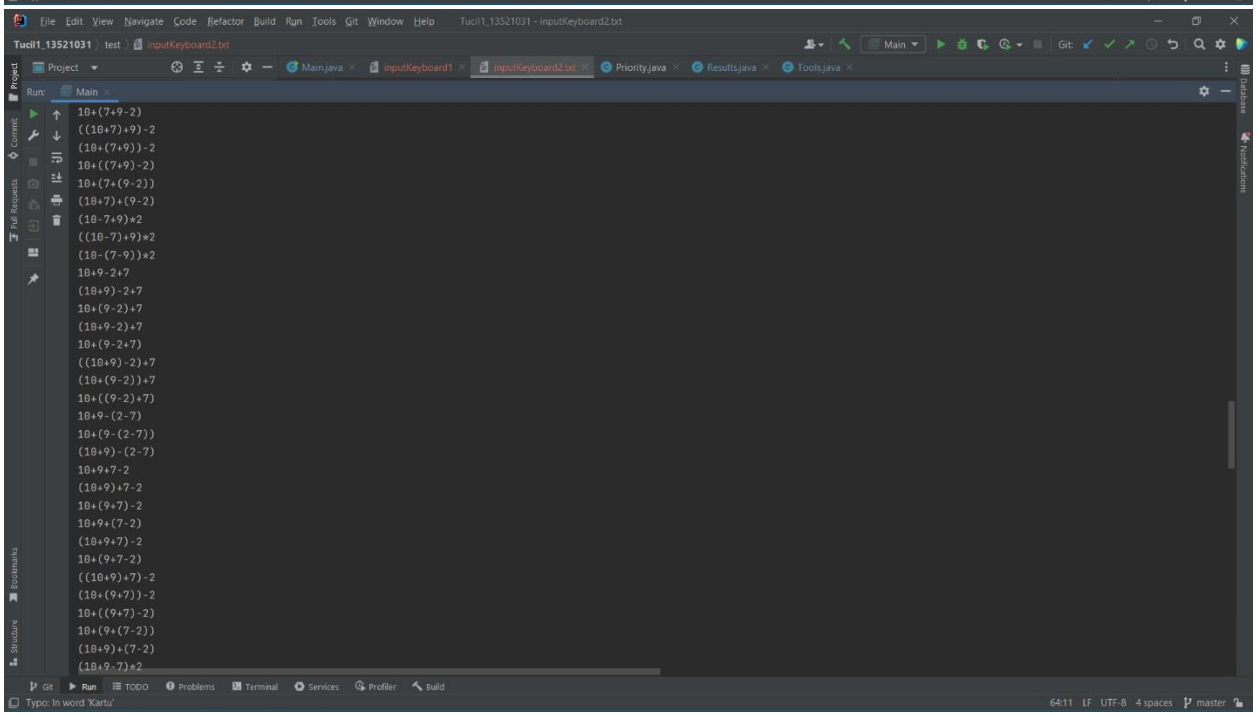
b. I/O with keyboard2

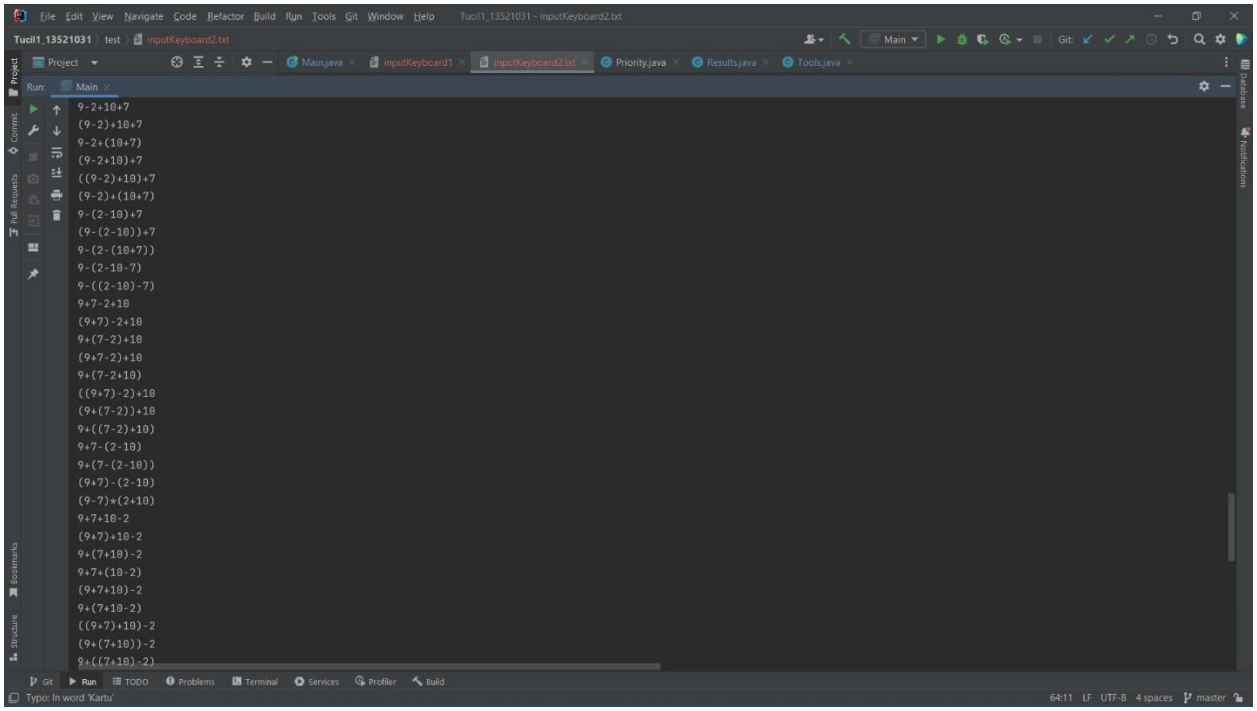


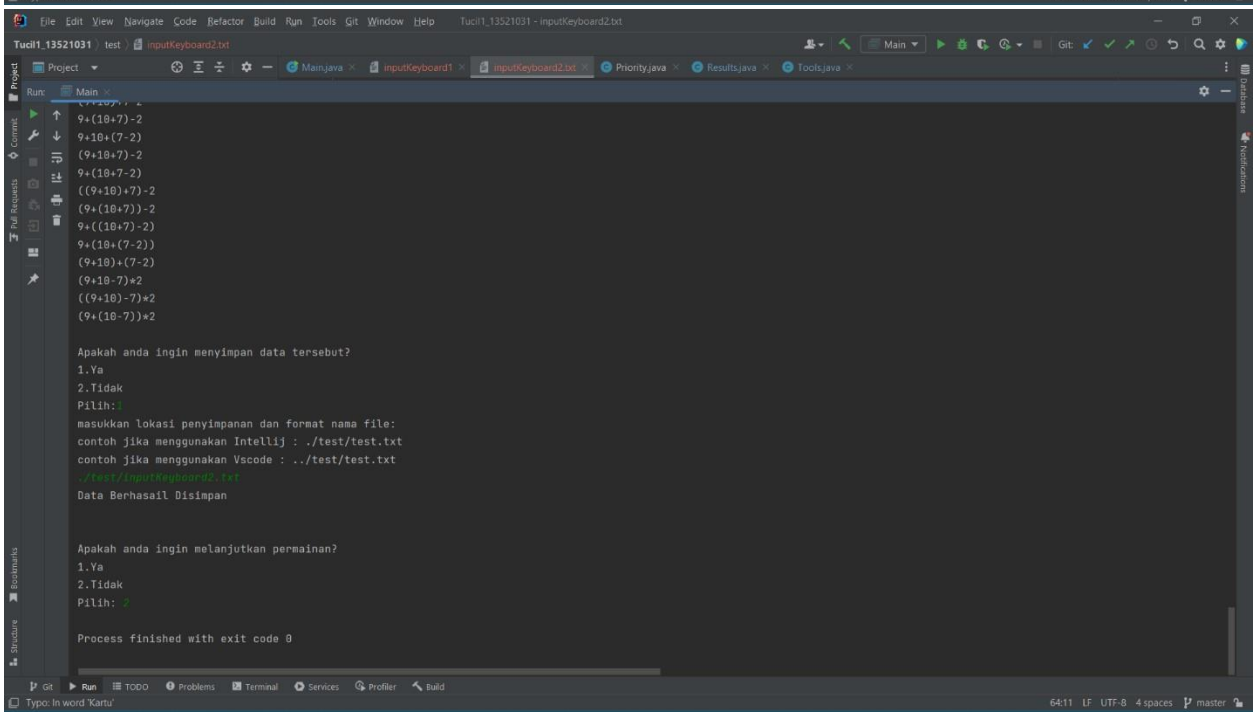

File Edit View Navigate Code Refactor Build Run Tools Git Window Help
Tucil1_13521031 - inputKeyboard2.txt
Tucil1_13521031 〉 test 〉 inputKeyboard2.txt
Main
Git:
Project
Main.java
inputKeyboard1
inputKeyboard2.txt
Priority.java
Results.java
Tools.java
Run:
Main
7-2+10+9
(7-2)+10+9
7-2+(10+9)
(7-2+10)+9
((7-2)+10)+9
(7-2)+(10+9)
7-(2-10)+9
(7-(2-10))+9
7-(2-(10+9))
7-(2-10-9)
7-((2-10)-9)
7-2+9+10
(7-2)+9+10
7-2+(9+10)
(7-2+9)+10
((7-2)+9)+10
(7-2)+(9+10)
7-(2-9)+10
(7-(2-9))+10
7-(2-(9+10))
7-(2-9-10)
7-((2-9)-10)
7+10-2+9
(7+10)-2+9
7+(10-2)+9
(7+10-2)+9
7+(10-2+9)
((7+10)-2)+9
(7+(10-2))+9
7+((10-2)+9)
7+10-(2-9)
7+(10-(2-9))
Git
Run
TODO
Problems
Terminal
Services
Profiler
Build
Typo: In word 'Kartu'
32:13 LF UTF-8 4 spaces master


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
Tucil1_13521031 - inputKeyboard2.txt
Tucil1_13521031 〉 test 〉 inputKeyboard2.txt
Main
Git:
Project
Main.java
inputKeyboard1
inputKeyboard2.txt
Priority.java
Results.java
Tools.java
Run:
Main
10-2+7+9
(10-2)+7+9
10-2+(7+9)
(10-2+7)+9
((10-2)+7)+9
(10-2)+(7+9)
10-(2-7)+9
(10-(2-7))+9
10-(2-(7+9))
10-(2-7-9)
10-((2-7)-9)
(10+2)*(9-7)
10-2+9+7
(10-2)+9+7
10-2+(9+7)
(10-2+9)+7
((10-2)+9)+7
(10-2)+(9+7)
10-(2-9)+7
(10-(2-9))+7
10-(2-(9+7))
10-(2-9-7)
10-((2-9)-7)
10+7-2+9
(10+7)-2+9
10+(7-2)+9
(10+7-2)+9
10+(7-2+9)
((10+7)-2)+9
(10+(7-2))+9
10+((7-2)+9)
10+7-(2-9)
Git
Run
TODO
Problems
Terminal
Services
Profiler
Build
Typo: In word 'Kartu'
64:11 LF UTF-8 4 spaces master

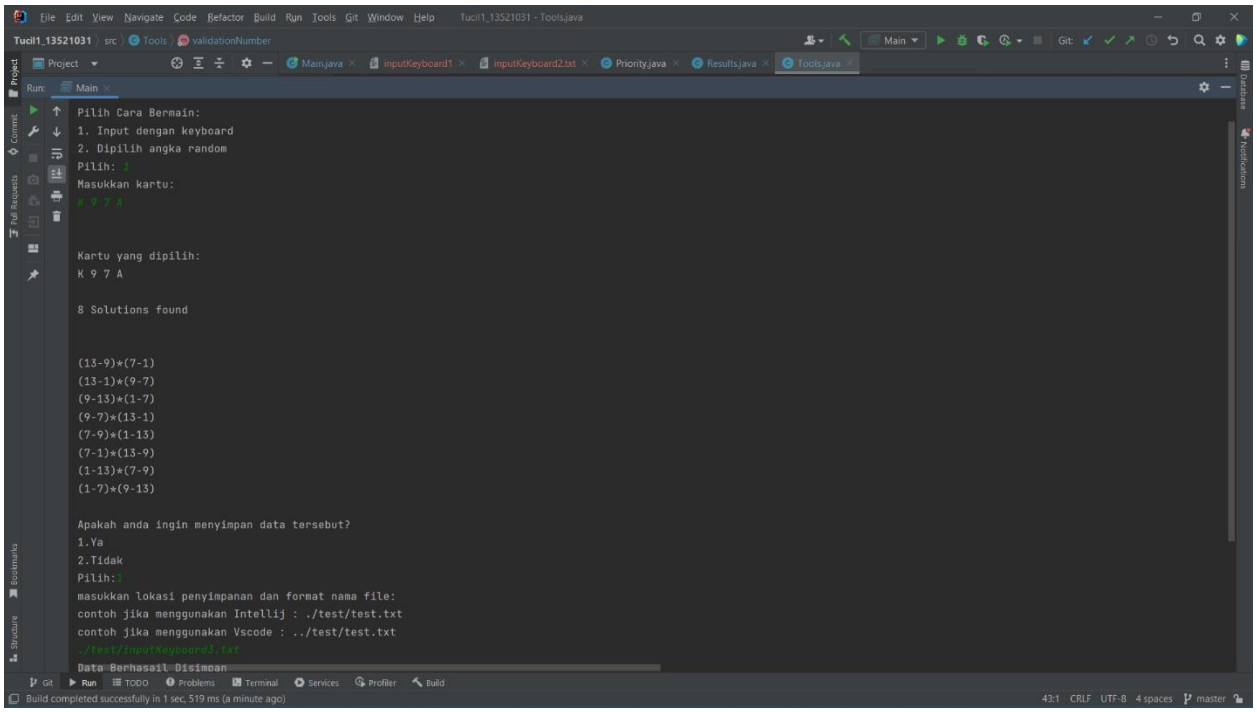

File Edit View Navigate Code Refactor Build Run Tools Git Window Help
Tucil1_13521031 - inputKeyboard2.txt
Tucil1_13521031 › test › inputKeyboard2.txt
Main
Git:
Project
Main.java
inputKeyboard1
inputKeyboard2.txt
Priority.java
Results.java
Tools.java
Run: Main
7+10+9-2
(7+10)+9-2
7+(10+9)-2
7+10+(9-2)
(7+10+9)-2
7+(10+9-2)
((7+10)+9)-2
(7+(10+9))-2
7+((10+9)-2)
7+(10+(9-2))
(7+10)+(9-2)
7+9-2+10
(7+9)-2+10
7+(9-2)+10
(7+9-2)+10
7+(9-2+10)
((7+9)-2)+10
(7+(9-2))+10
7+((9-2)+10)
7+9-(2-10)
7+(9-(2-10))
(7+9)-(2-10)
7+9+10-2
(7+9)+10-2
7+(9+10)-2
7+9+(10-2)
(7+9+10)-2
7+(9+10-2)
((7+9)+10)-2
(7+(9+10))-2
7+((9+10)-2)
7+(9+(10-2))
Git Run TODO Problems Terminal Services Profiler Build
Typo: In word 'Kartu'
64:11 LF UTF-8 4 spaces master

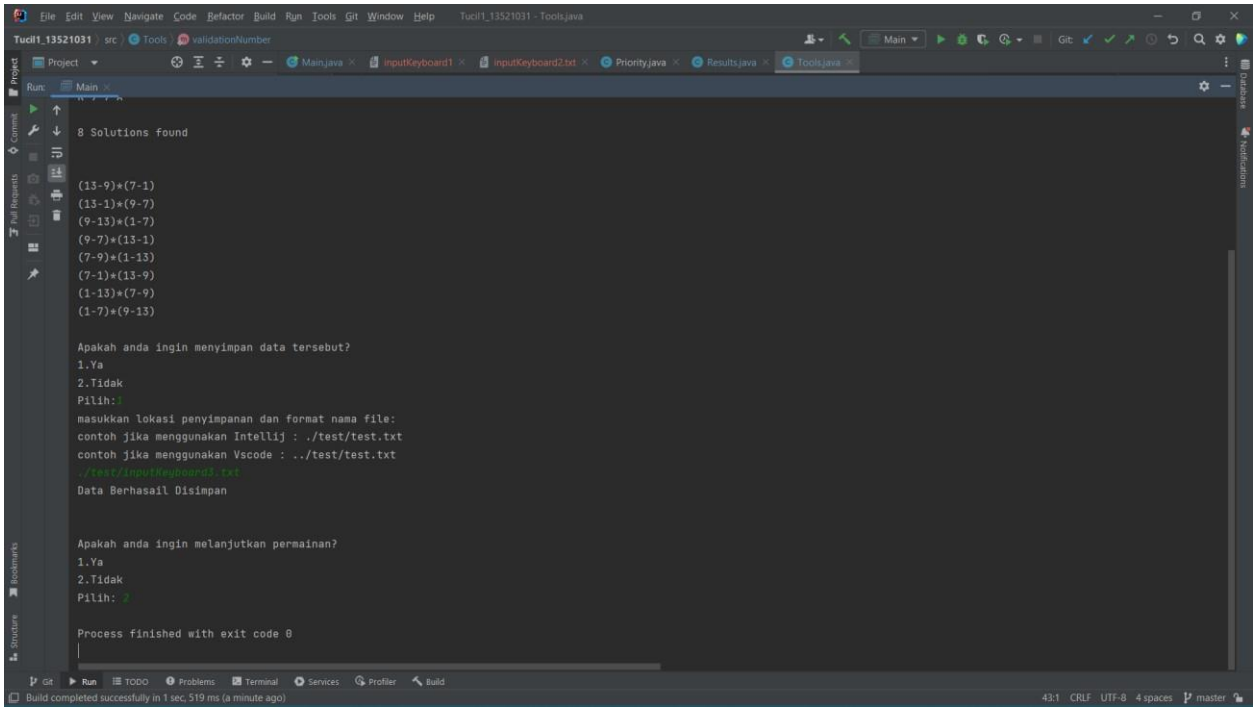

File Edit View Navigate Code Refactor Build Run Tools Git Window Help
Tucil1_13521031 - inputKeyboard2.txt
Tucil1_13521031 › test › inputKeyboard2.txt
Main
Git:
Project
Main.java
inputKeyboard1
inputKeyboard2.txt
Priority.java
Results.java
Tools.java
Run: Main
10+(7+9-2)
((10+7)+9)-2
(10+(7+9))-2
10+((7+9)-2)
10+(7+(9-2))
(10+7)+(9-2)
(10-7+9)*2
((10-7)+9)*2
(10-(7-9))*2
10+9-2+7
(10+9)-2+7
10+(9-2)+7
(10+9-2)+7
10+(9-2+7)
((10+9)-2)+7
(10+(9-2))+7
10+((9-2)+7)
10+9-(2-7)
10+(9-(2-7))
(10+9)-(2-7)
10+9+7-2
(10+9)+7-2
10+(9+7)-2
10+9+(7-2)
(10+9+7)-2
10+(9+7-2)
((10+9)+7)-2
(10+(9+7))-2
10+((9+7)-2)
10+(9+(7-2))
(10+9)+(7-2)
(10+9-7)*2
Git Run TODO Problems Terminal Services Profiler Build
Typo: In word 'Kartu'
64:11 LF UTF-8 4 spaces master


9-2+10+7
(9-2)+10+7
9-2+(10+7)
(9-2+10)+7
((9-2)+10)+7
(9-2)+(10+7)
9-(2-10)+7
(9-(2-10))+7
9-(2-(10+7))
9-(2-10-7)
9-((2-10)-7)
9+7-2+10
(9+7)-2+10
9+(7-2)+10
(9+7-2)+10
9+(7-2+10)
((9+7)-2)+10
(9+(7-2))+10
9+((7-2)+10)
9+7-(2-10)
9+(7-(2-10))
(9+7)-(2-10)
(9-7)*(2+10)
9+7+10-2
(9+7)+10-2
9+(7+10)-2
9+7+(10-2)
(9+7+10)-2
9+(7+10-2)
((9+7)+10)-2
(9+(7+10))-2
9+((7+10)-2)

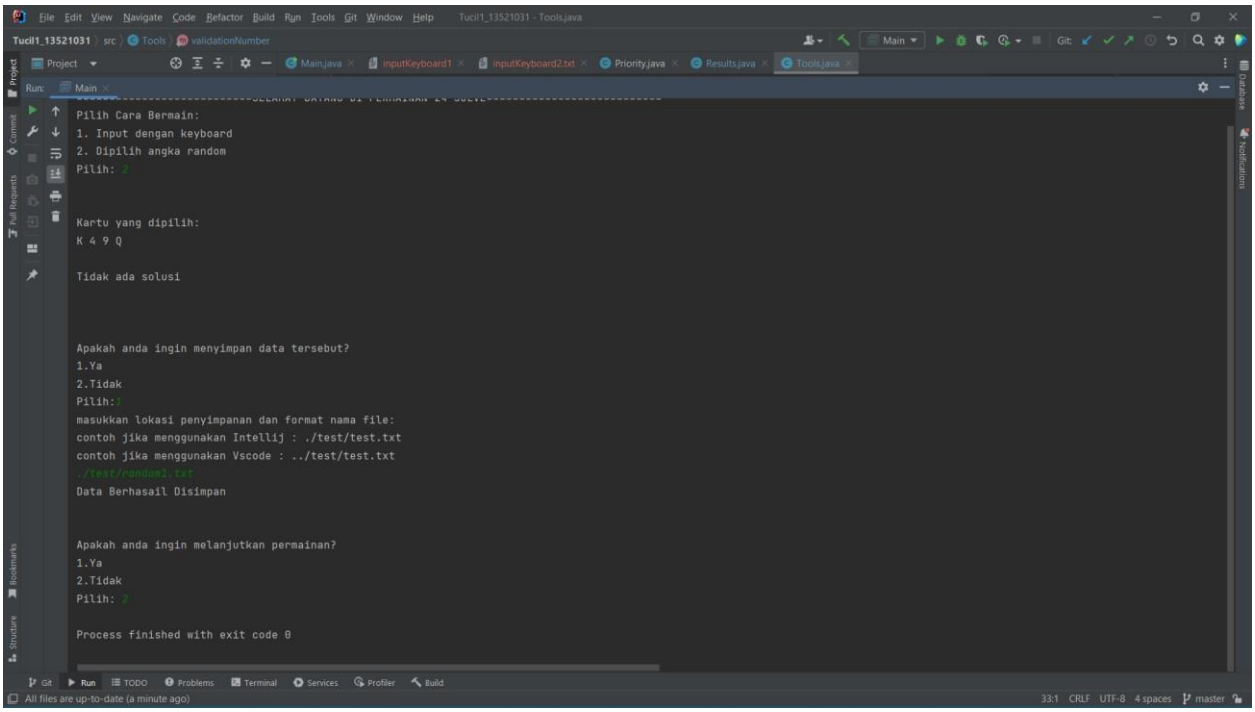

9+(10+7)-2
9+10+(7-2)
(9+10+7)-2
9+(10+7-2)
((9+10)+7)-2
(9+(10+7))-2
9+((10+7)-2)
9+(10+(7-2))
(9+10)+(7-2)
(9+10-7)*2
((9+10)-7)*2
(9+(10-7))*2
Apakah anda ingin menyimpan data tersebut?
1.Ya
2.Tidak
Pilih:1
masukkan lokasi penyimpanan dan format nama file:
contoh jika menggunakan Intellij : ./test/test.txt
contoh jika menggunakan Vscode : ../test/test.txt
./test/inputKeyboard2.txt
Data Berhasail Disimpan
Apakah anda ingin melanjutkan permainan?
1.Ya
2.Tidak
Pilih: 2
Process finished with exit code 0

c. IO/ keyboard3

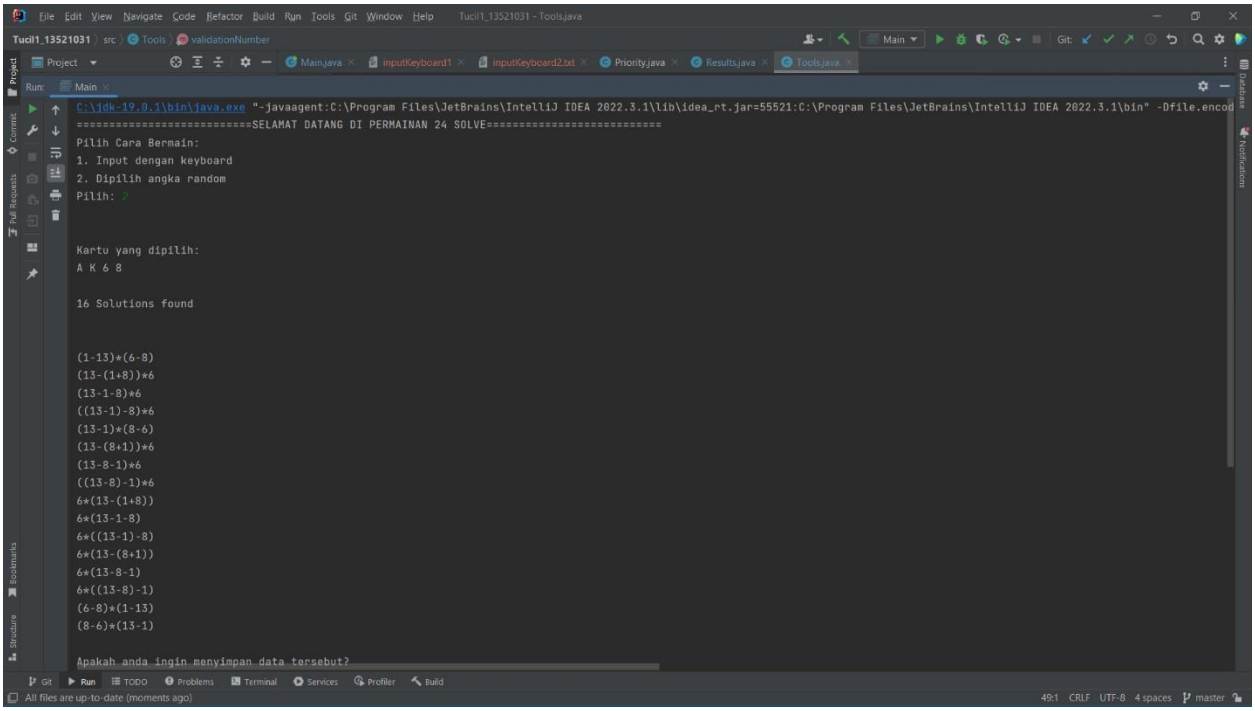


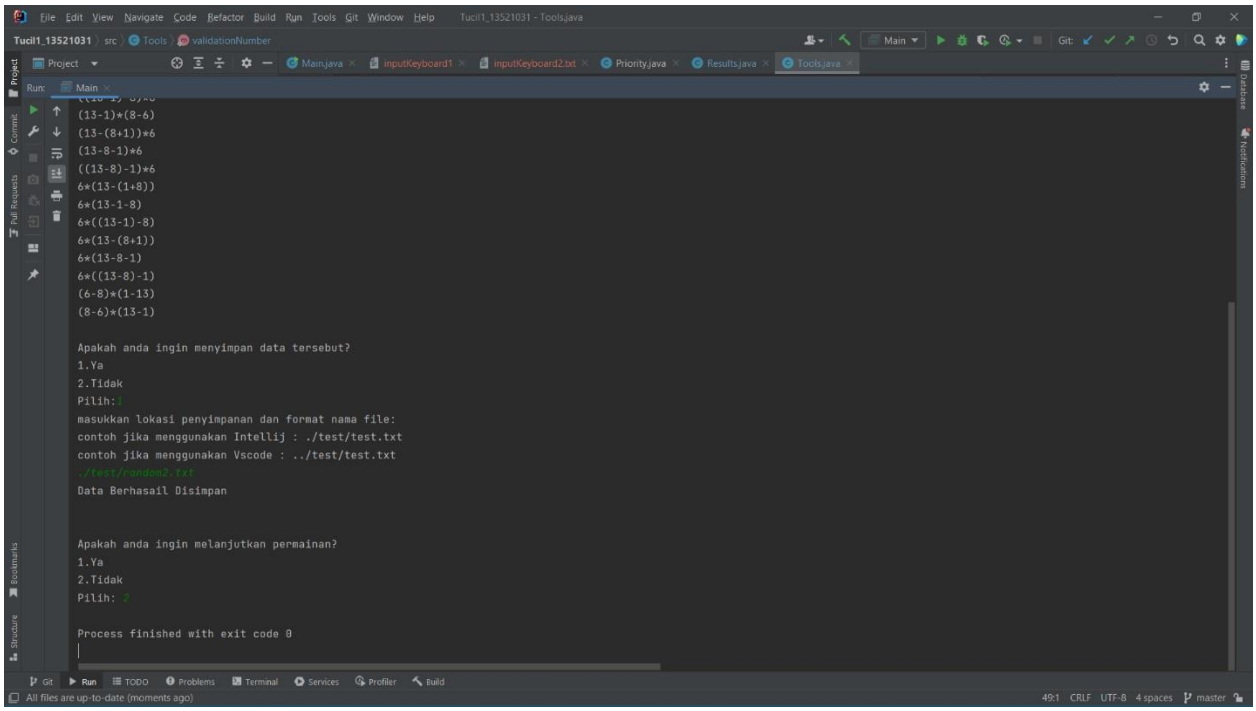


d. I/O with random1

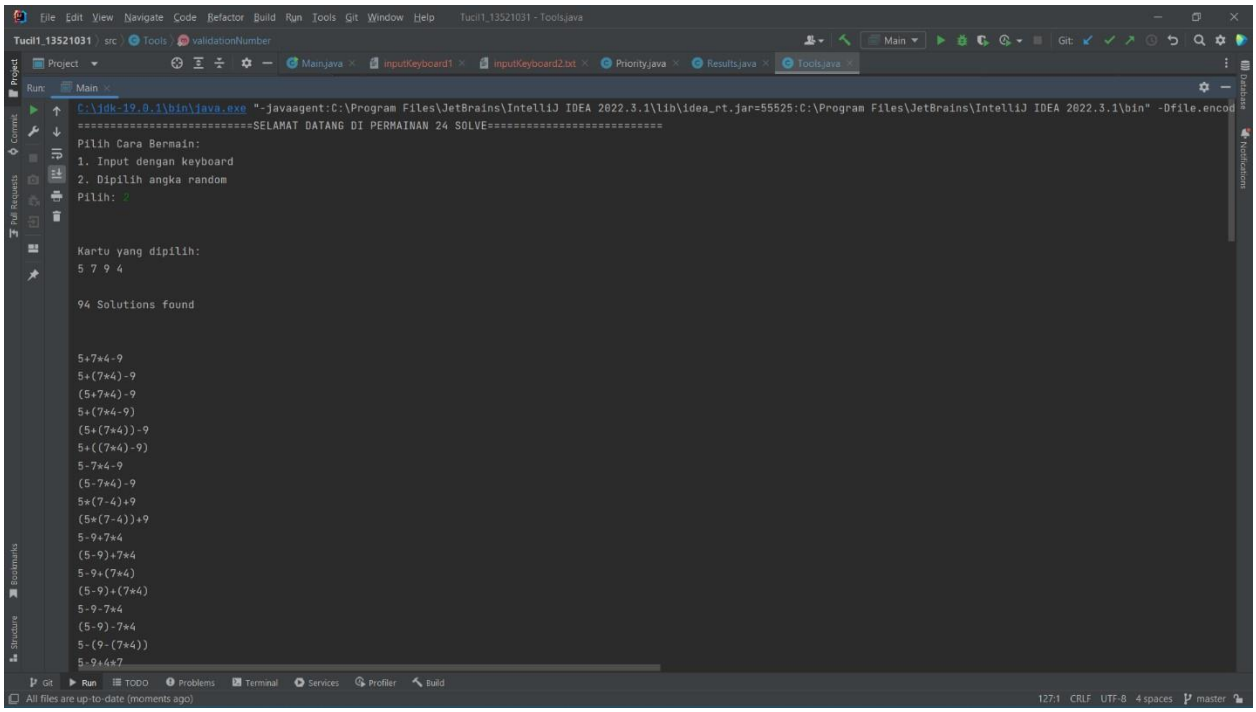


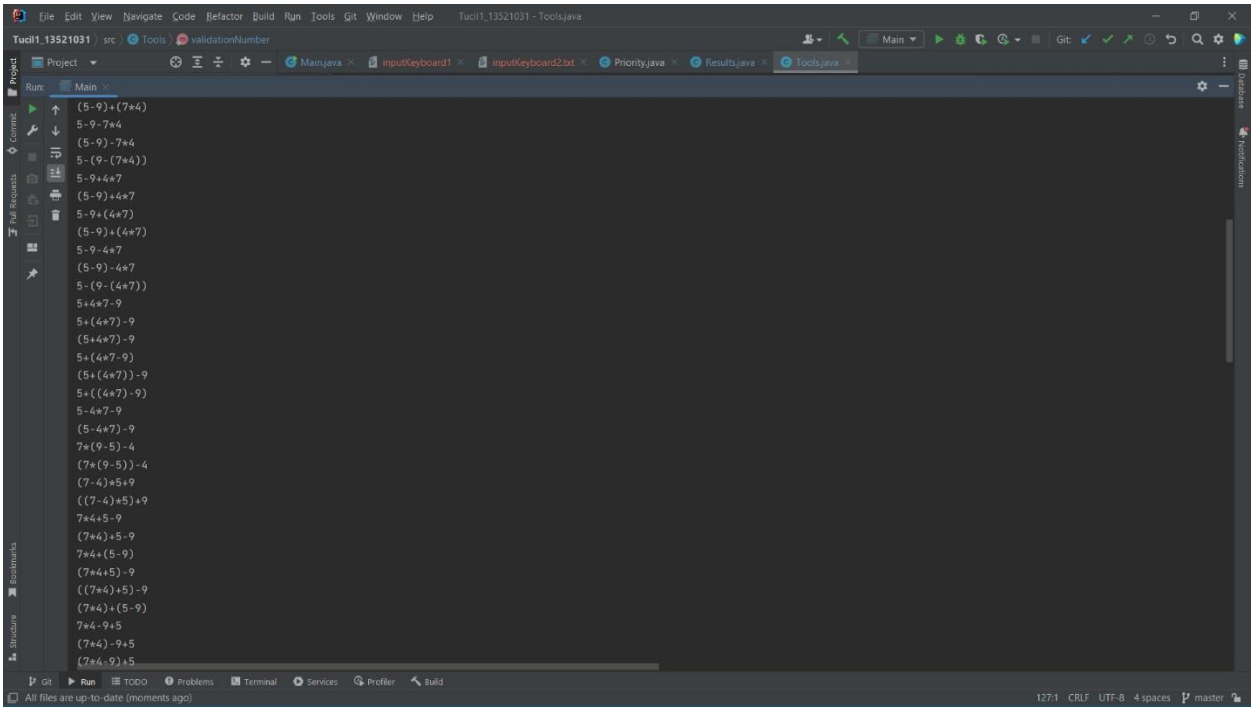
e. I/O with random2

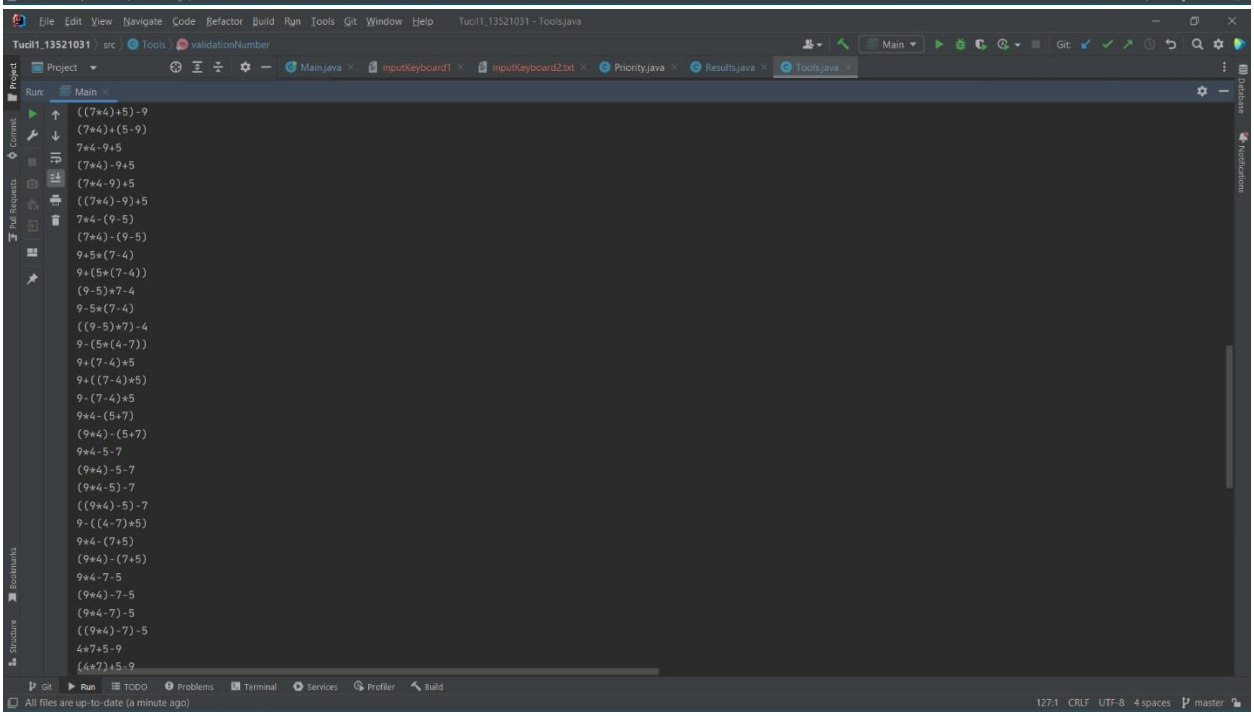


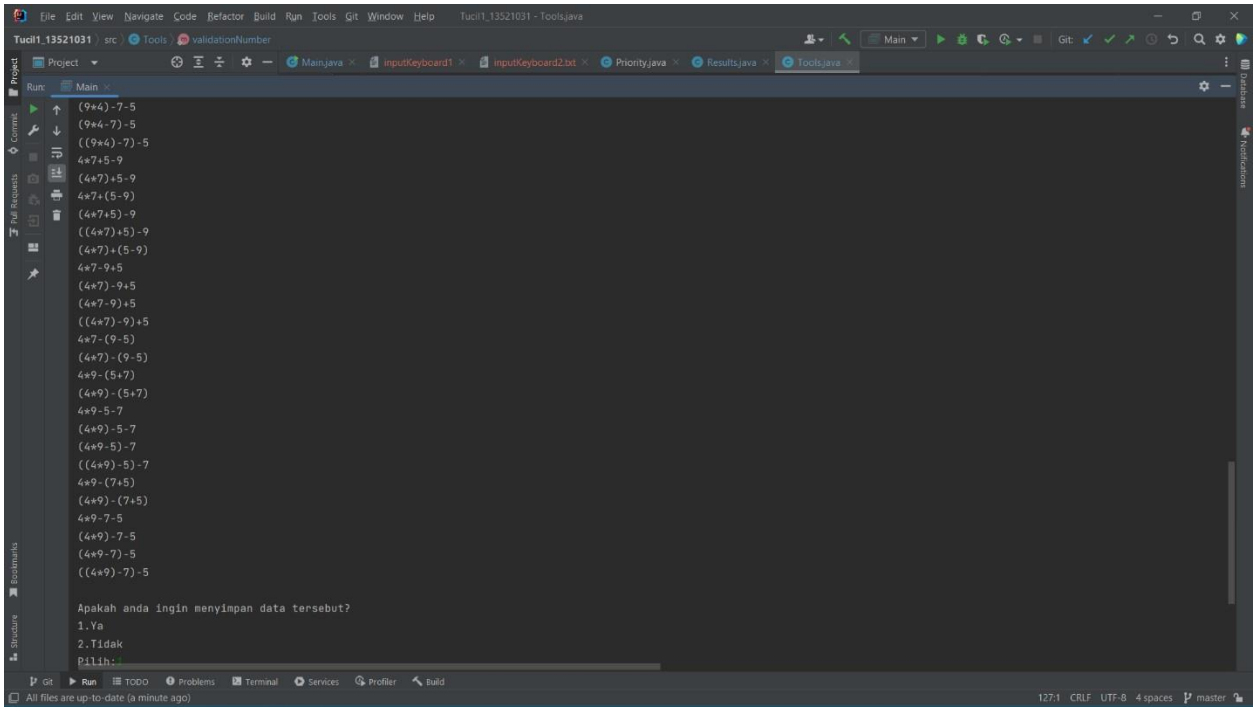


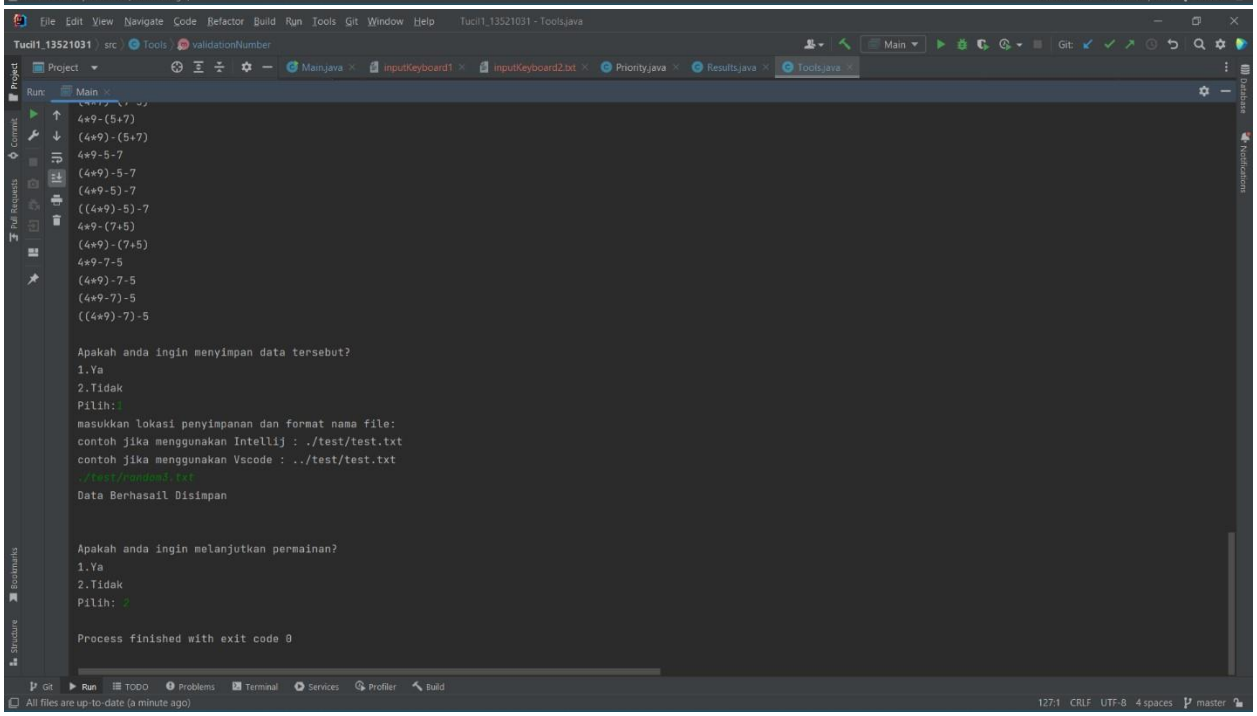
f. I/O with random3



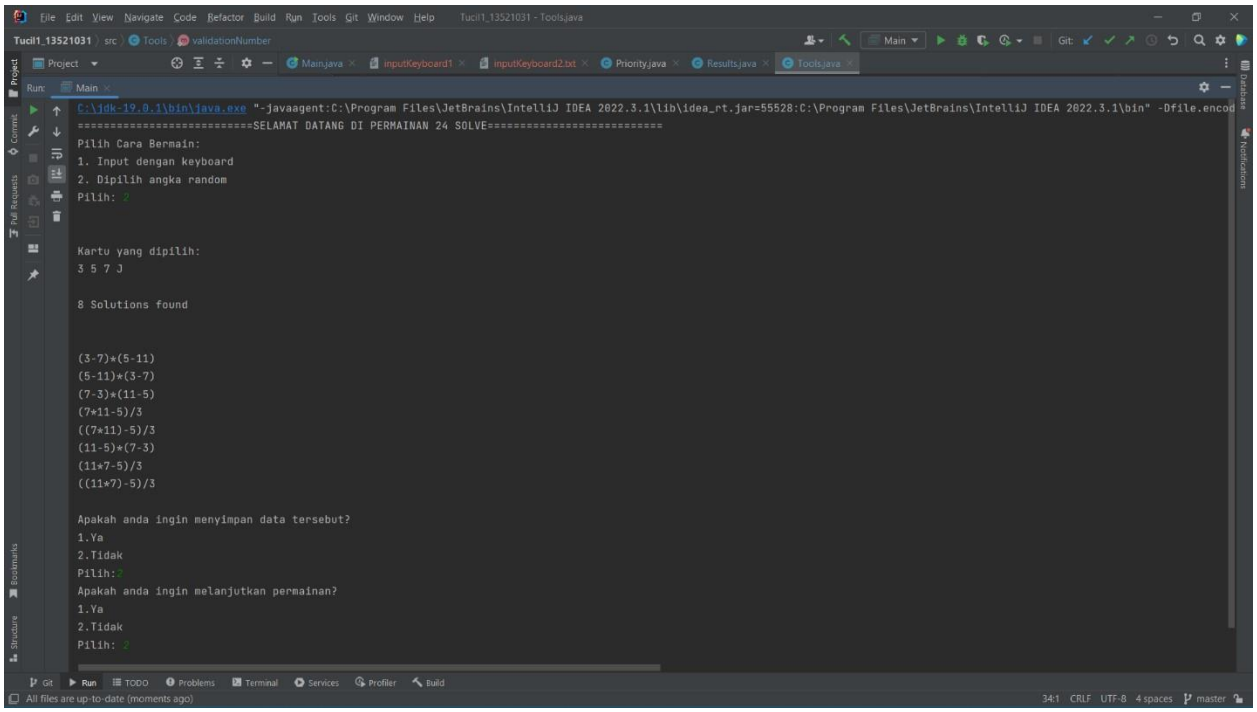

File Edit View Navigate Code Refactor Build Run Tools Git Window Help
Tucil1_13521031 - Tools.java
Tucil1_13521031 › src › Tools › validationNumber
Main
Git:
Project
Main.java
inputKeyboard1
inputKeyboard2.txt
Priority.java
Results.java
Tools.java
Run:
Main
(5-9)+(7*4)
5-9-7*4
(5-9)-7*4
5-(9-(7*4))
5-9+4*7
(5-9)+4*7
5-9+(4*7)
(5-9)+(4*7)
5-9-4*7
(5-9)-4*7
5-(9-(4*7))
5+4*7-9
5+(4*7)-9
(5+4*7)-9
5+(4*7-9)
(5+(4*7))-9
5+((4*7)-9)
5-4*7-9
(5-4*7)-9
7*(9-5)-4
(7*(9-5))-4
(7-4)*5+9
((7-4)*5)+9
7*4+5-9
(7*4)+5-9
7*4+(5-9)
(7*4+5)-9
((7*4)+5)-9
(7*4)+(5-9)
7*4-9+5
(7*4)-9+5
(7*4-9)+5
Git
Run
TODO
Problems
Terminal
Services
Profiler
Build
All files are up-to-date (moments ago)
127:1
CRLF
UTF-8
4 spaces
master


File Edit View Navigate Code Refactor Build Run Tools Git Window Help
Tucil1_13521031 - Tools.java
Tucil1_13521031 › src › Tools › validationNumber
Main
Git:
Project
Main.java
inputKeyboard1
inputKeyboard2.txt
Priority.java
Results.java
Tools.java
Run:
Main
((7*4)+5)-9
(7*4)+(5-9)
7*4-9+5
(7*4)-9+5
(7*4-9)+5
((7*4)-9)+5
7*4-(9-5)
(7*4)-(9-5)
9+5*(7-4)
9+(5*(7-4))
(9-5)*7-4
9-5*(7-4)
((9-5)*7)-4
9-(5*(4-7))
9+(7-4)*5
9+((7-4)*5)
9-(7-4)*5
9*4-(5+7)
(9*4)-(5+7)
9*4-5-7
(9*4)-5-7
(9*4-5)-7
((9*4)-5)-7
9-((4-7)*5)
9*4-(7+5)
(9*4)-(7+5)
9*4-7-5
(9*4)-7-5
(9*4-7)-5
((9*4)-7)-5
4*7+5-9
(4*7)+5-9
Git
Run
TODO
Problems
Terminal
Services
Profiler
Build
All files are up-to-date (a minute ago)
127:1
CRLF
UTF-8
4 spaces
master

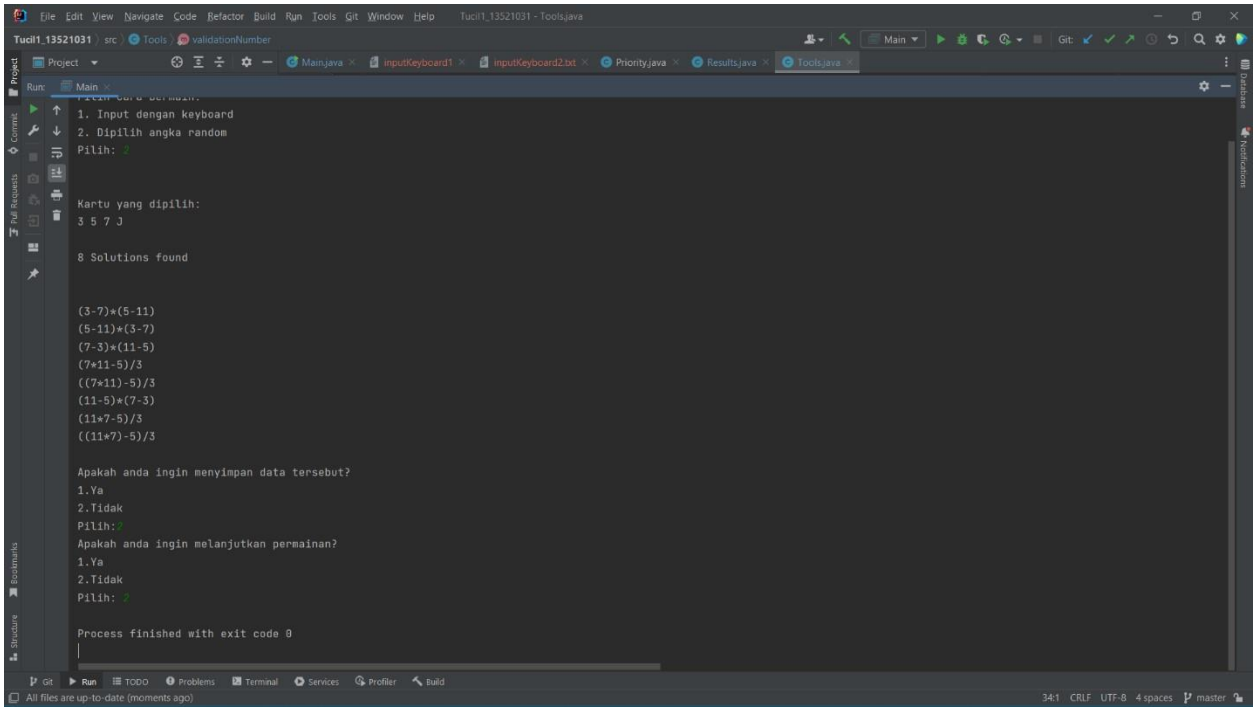

(9*4)-7-5
(9*4-7)-5
((9*4)-7)-5
4*7+5-9
(4*7)+5-9
4*7+(5-9)
(4*7+5)-9
((4*7)+5)-9
(4*7)+(5-9)
4*7-9+5
(4*7)-9+5
(4*7-9)+5
((4*7)-9)+5
4*7-(9-5)
(4*7)-(9-5)
4*9-(5+7)
(4*9)-(5+7)
4*9-5-7
(4*9)-5-7
(4*9-5)-7
((4*9)-5)-7
4*9-(7+5)
(4*9)-(7+5)
4*9-7-5
(4*9)-7-5
(4*9-7)-5
((4*9)-7)-5
Apakah anda ingin menyimpan data tersebut?
1.Ya
2.Tidak
Pilih:1


4*9-(5+7)
(4*9)-(5+7)
4*9-5-7
(4*9)-5-7
(4*9-5)-7
((4*9)-5)-7
4*9-(7+5)
(4*9)-(7+5)
4*9-7-5
(4*9)-7-5
(4*9-7)-5
((4*9)-7)-5
Apakah anda ingin menyimpan data tersebut?
1.Ya
2.Tidak
Pilih:1
masukkan lokasi penyimpanan dan format nama file:
contoh jika menggunakan Intellij : ./test/test.txt
contoh jika menggunakan Vscode : ../test/test.txt
./test/random5.txt
Data Berhasail Disimpan
Apakah anda ingin melanjutkan permainan?
1.Ya
2.Tidak
Pilih: 2
Process finished with exit code 0

g. I/O with random4



```
C:\jdk-19.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2022.3.1\lib\idea_rt.jar=55528:C:\Program Files\JetBrains\IntelliJ IDEA 2022.3.1\bin" -Dfile.encod
==========SELAMAT DATANG DI PERMAINAN 24 SOLVE==========
Pilih Cara Bermain:
1. Input dengan keyboard
2. Dipilih angka random
Pilih: 2


Kartu yang dipilih:
3 5 7 J

8 Solutions found


(3-7)*(5-11)
(5-11)*(3-7)
(7-3)*(11-5)
(7*11-5)/3
((7*11)-5)/3
(11-5)*(7-3)
(11*7-5)/3
((11*7)-5)/3

Apakah anda ingin menyimpan data tersebut?
1.Ya
2.Tidak
Pilih:2
Apakah anda ingin melanjutkan permainan?
1.Ya
2.Tidak
Pilih: 2
```



```
1. Input dengan keyboard
2. Dipilih angka random
Pilih: 2


Kartu yang dipilih:
3 5 7 J

8 Solutions found


(3-7)*(5-11)
(5-11)*(3-7)
(7-3)*(11-5)
(7*11-5)/3
((7*11)-5)/3
(11-5)*(7-3)
(11*7-5)/3
((11*7)-5)/3

Apakah anda ingin menyimpan data tersebut?
1.Ya
2.Tidak
Pilih:2
Apakah anda ingin melanjutkan permainan?
1.Ya
2.Tidak
Pilih: 2

Process finished with exit code 0
```

4. Link Repository Tucil1_13521031

5. Table

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	✓	