

Projek Akhir

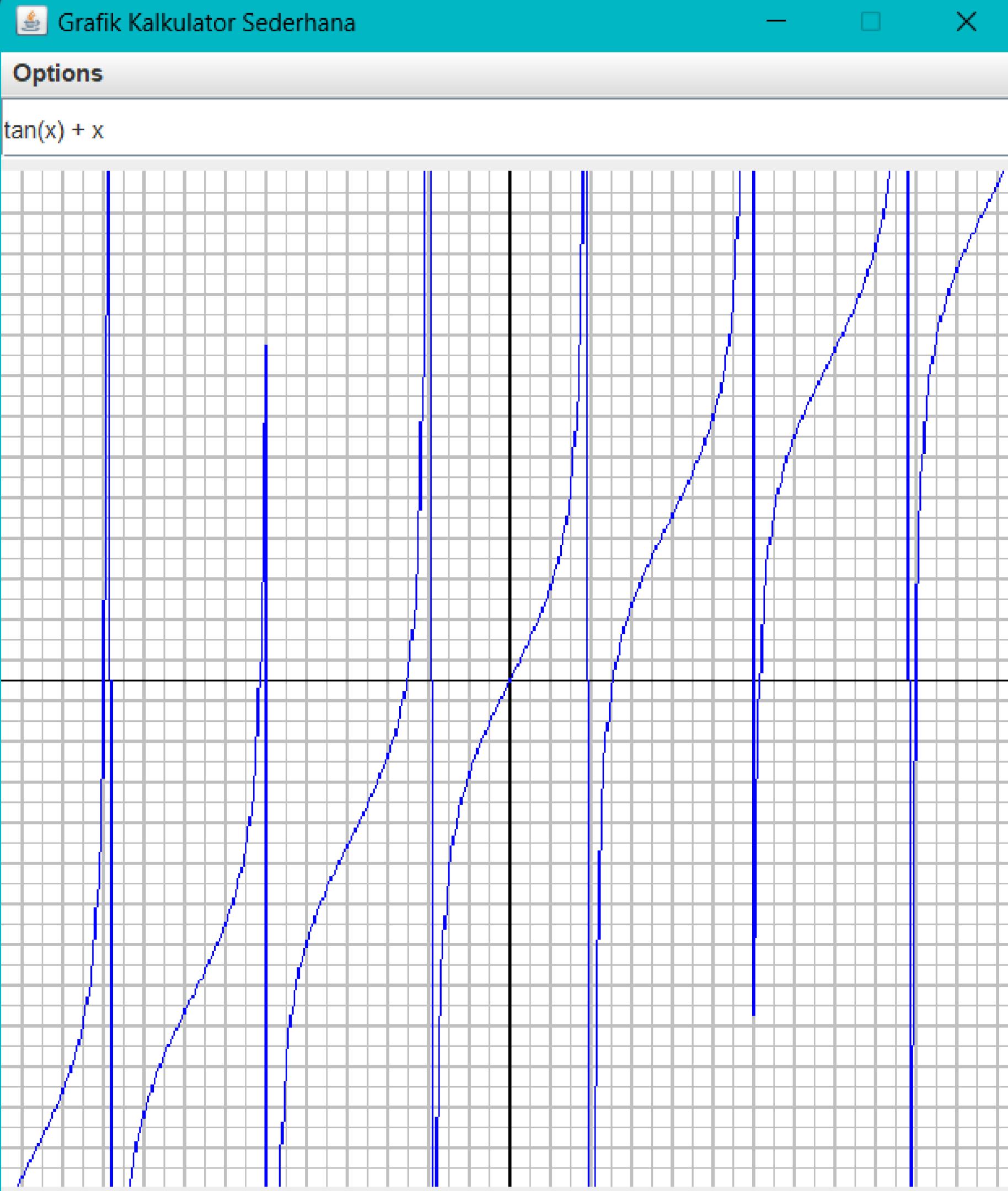
Pemograman Berorientasi Objek

Grafik Kalkulator Sederhana

- Achmad Hadi

Grafik Kalkulator

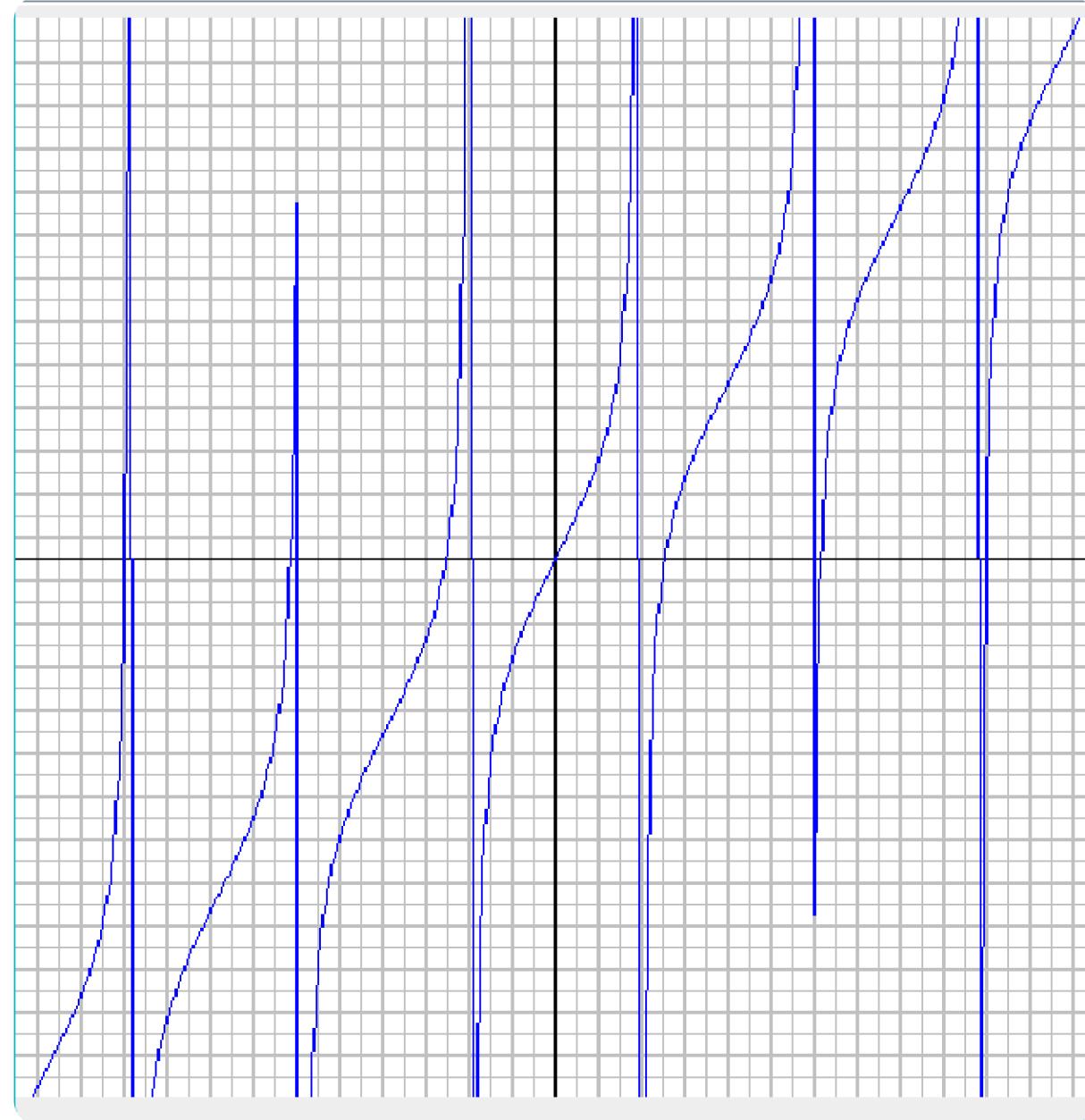
Di projek ini kalian dapat menginput fungsi grafik namun di kendalakan oleh programer pemula, input fungsi hanya fungsi basic, berupa penjumlahan perkalian dan trigonometry sederhana. di sini kalin juga bisa mengubah ukuran dari grafik.





pallette JTextField dengan nama
variable inputFungsiEkuasi

pallette JPanel dengan variable
panelGrafik, yang dimana di panel
ini ditambahkan metode Override
PaintComponent agar mampu
untuk menggambar grafik



pallette JMenuBar untuk tambahan
opsi seperti kembali ke menu atau
mengatur ukuran grafik

Kode Fungsi Gambar Grafik :

```
● ● ●

1  protected void gambarGrafik() {
2      String inputEquation = inputFungsiEkuasi.getText();
3
4      int lebar = panelGrafik.getWidth();
5      int tinggi = panelGrafik.getHeight();
6
7      offScreenImage = new BufferedImage(lebar, tinggi, BufferedImage.TYPE_INT_ARGB);
8      Graphics2D g2d = offScreenImage.createGraphics();
9
10     g2d.setColor(Color.WHITE);
11     g2d.fillRect(0, 0, lebar, tinggi);
12
13     garisKoordinat(g2d, lebar, tinggi);
14     garisKartesius(g2d, lebar, tinggi);
15
16     if (hasBracketsFormat(inputEquation)) {
17         String[] equations = inputEquation.split("],\\s*[[");
18         for (String equation : equations) {
19             equation = equation.replace("[", "").replace("]", "");
20             plotEquation(g2d, equation, lebar, tinggi);
21         }
22     } else {
23         plotEquation(g2d, inputEquation, lebar, tinggi);
24     }
25
26     g2d.dispose();
27     panelGrafik.repaint();
28 }
```

Penjelasan Kode

- Fungsi gambarGrafik di ubah menjadi acces modifier protected karena akan digunakan oleh kelas induk (Parent)
- Pertama kode akan meminta input dari pengguna
- setelah itu kode akan meminta tinggi dan lebar dari JPanel
- Setelah mendapat tinggi dan lebar kode akan menggambar seluruh JPanel menjadi warna putih, dan menggambar garis kartesius dan garis koordinat
- Kode akan mengecek input dari pengguna, apakah input pengguna merupakan multiple input fungsi atau single input fungsi
- setelah menentukan hal tersebut, kode akan mengkalkulasi di plotEquation
- Setelah kode selesai di kalkulasi, pertama kode akan melepas objek graphic sebelumnya dan kemudian menggambarnya kembali.

Kode Fungsi Plot Equation :



```
1 private void plotEquation(Graphics2D g, String equation, int lebar, int tinggi) {  
2     List<Double> xValues = getXValues(startX, endX, 0.1);  
3     List<Double> yValues = fungsiPlot(equation, xValues);  
4  
5     Path2D path = new Path2D.Double();  
6  
7     double scaleX = lebar / (xValues.get(xValues.size() - 1) - xValues.get(0));  
8     double scaleY = tinggi / (xValues.get(xValues.size() - 1) - xValues.get(0));  
9     path.moveTo(lebar / 2 + xValues.get(0) * scaleX, tinggi / 2 - yValues.get(0) * scaleY);  
10  
11    for (int i = 1; i < xValues.size(); i++) {  
12        path.lineTo(lebar / 2 + xValues.get(i) * scaleX, tinggi / 2 - yValues.get(i) * scaleY);  
13    }  
14  
15    g.setColor(Color.BLUE);  
16    g.draw(path);  
17 }
```

Kode Fungsi X value dan Y value :

```
● ● ●

1 private List<Double> getXValues(double start, double end, double interval) {
2     List<Double> xValues = new ArrayList<>();
3     for (double x = start; x <= end; x += interval) {
4         xValues.add(x);
5     }
6     return xValues;
7 }
8
9 private List<Double> fungsiPlot(String equation, List<Double> xValues) {
10    List<Double> yValues = new ArrayList<>();
11    for (double x : xValues) {
12        double y = kalkulasiFungsi(equation, x);
13        yValues.add(y);
14    }
15    return yValues;
16 }
```

Penjelasan Kode

- Pertama kita akan meminta kumpulan nilai xValue dan yValue, dimana nilai xValue adalah nilai dari kalkulasi dari xStart dan xEnd (cth: -10.0, -9.9, -9.8, ..., 10,0).
- setelah mendapatkan kumpulan nilai xValue, setiap nilai xValue akan di kalkulasi dengan ekuasi yang sudah user berikan
- Fungsi path.moveTo digunakan untuk menetukan titik untuk koordinat, yang akan digunakan untuk menggambar grafik nantinya
- Sedangkan fungsi path.lineTo digunakan untuk menggambar garis dari nilai xValue dan yValue.
- g.SetColor untuk menetukan warna grafik

Kode Fungsi Kalkulasi Fungsi :

```
● ● ●

1  private double kalkulasiFungsi(String equation, double x) {
2      String[] parts = equation.split("[+]");
3      double y = 0;
4      for (String part : parts) {
5          if (part.contains("x")) {
6              String[] KoefisienParts = part.split("x");
7              double Koefisien = KoefisienParts[0].isEmpty() ? 1.0 : parseDouble(KoefisienParts[0], 0);
8              if (part.contains("^")) {
9                  String[] exponentParts = part.split("\\^");
10                 int exponent = (int) parseDouble(exponentParts[1], 0);
11                 y += Koefisien * Math.pow(x, exponent);
12             } else if (part.contains("*")) {
13                 String[] perkalianPart = part.split("\\*");
14                 double perkalianKoefisien = parseDouble(perkalianPart[0], 1);
15                 y += perkalianKoefisien * trigonometryFungsi(perkalianPart[1], x);
16             } else {
17                 y += trigonometryFungsi(part, x);
18             }
19         } else {
20             y += parseDouble(part, 0);
21         }
22     }
23     return y;
24 }
```

Kode Fungsi Trigonometry Fungsi:

```
● ● ●

1  private double trigonometryFungsi(String part, double x) {
2      double y = 0;
3      if (part.contains("sin") || part.contains("cos") || part.contains("tan")) {
4          Pattern pattern = Pattern.compile("[+-]?\\d*\\.?\\d+?(sin|cos|tan)\\((x)\\)");
5          Matcher matcher = pattern.matcher(part);
6          if (matcher.find()) {
7              String Koefisien = matcher.group(1);
8              String function = matcher.group(2);
9              double KoefisienValue = (Koefisien != null && !Koefisien.isEmpty()) ? Double.parseDouble(Koefisien) : 1.0;
10             switch (function) {
11                 case "sin" -> y = KoefisienValue * Math.sin(x);
12                 case "cos" -> y = KoefisienValue * Math.cos(x);
13                 case "tan" -> y = KoefisienValue * Math.tan(x);
14                 default -> throw new AssertionError("Unexpected function: " + function);
15             }
16         }
17     } else {
18         String[] KoefisienParts = part.split("x");
19         double Koefisien = KoefisienParts[0].isEmpty() ? 1 : parseDouble(KoefisienParts[0], 1);
20         y = Koefisien * x;
21     }
22     return y;
23 }
```

Penjelasan Kode

- Pertama kita akan memisahkan nilai dari setiap penambahan fungsi di inputEkuaasi
- setelah memisahkan nilai kita melakukan perulangan dan mengecek nilai tersebut value x atau tidak.
- Jika value x memiliki nilai x maka kita akan memisahkan nilai n dan f (cth input: nx^f -> [n], [f], n adalah nilai dan f adalah fungsi perkalian, pangkat dan sebagainya)
- Di trigonometry kode mengecek apakah input user memiliki nilai sin, cos, dan tan. jika iya maka kode akan melakukan perhitungan menggunakan pustaka Math
- dan mengembalikan nilai ke yValue.

Demo GUI

github.com/MMOSHII/Grafik-Kalkulator

```
11 # now we can access the features
12 if "title" in theJSON["features"]:
13     print(theJSON["metadata"])
14
15 # output the number of events, then
16 count = theJSON["metadata"]["count"]
17 print(str(count) + " events recorded")
18
19 # for each event, print the place name
20 for i in theJSON["features"]:
21     print(i["properties"]["place"])
22     print("\n")
23
24 # print the events that only have a magnitude
25 for i in theJSON["features"]:
26     if i["properties"]["mag"] >= 4.0:
27         print("Magnitude: " + str(i["properties"]["mag"]))
28         print("\n")
29
30 # print the events where at least one person
31 # reported that they were felt:
32 for i in theJSON["features"]:
33     if i["properties"]["felt"] > 0:
34         print("Felt by: " + str(i["properties"]["felt"]))
35         print("\n")
```