Single-Layer Electronic Circuit Design Utilizing Planar Graph Theory

Ivan Wirawan - 13523046[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]ivanwirawan25@gmail.com, 13523046@std.stei.itb.ac.id

***Abstract*—The concepts of graph theory have been widely used in many different professions sectors. Such example is electric circuits. In a single-layer electronic circuit, mainly Printed Circuits Board (PCB), the planar graph theory is applied actively. The conductive path in the electronic circuit should not intersect with each other to avoid short circuit. This study is carried out to deepen the understanding of planar graph theory as well as its application in single-layer electronic circuits. The methodology used in this circuit is literature review and digital programming. The output of this study is a digital program which can automatically determine the planarity of a digital circuit with components and connections from user's input. The program will be utilizing graph theory to determine the planarity status. This research is hoped to bridge the gap between theories and real-life challenge by offering a practical program to be used.**

***Keywords*—Graph, Planar, Electric Circuit, Programming, Practical.**

I. Introduction

In this modern era of technology, gadgets surrounding humankind life are more complex than ever. Electronic gadget's development has skyrocketed in the past decade. Humankind's life is now mostly based on an electronic device functioning. Such as smartphones, laptops, tablets, lamps, fans, and many other examples. Creating an inseparable environment of civilization and electronic devices.

The base of all electronic gadgets is an electronic circuit. The meaning behind a circuit is the connection between every electronic component that forms a functional circuit. The function of electronic circuit can be designed towards a specific purpose. Thus, encouraging civilization's utilization of electronic circuits to help operation in professional or personal life.

The electronic circuit design is also in the most sophisticated form ever. The compactness, functionality, modularity, and many more aspects of the electrical have developed significantly. Although the evolution of electronic circuits has created a much different than the early days, some of the basic concepts are still the same.

As the demand for more compact, efficient, effective, and better electronic equipment is increasing, further study and research is being carried out. The incorporation of other field concepts such as mathematics and informatics has always supported the evolution. As each of the study fields have also developed, the concepts that can be aggregated are much more. One of the examples of the basic concepts that is still used to this day is the planar graph concept in a circuit board.

II. Electronics

Electronics are branch of physics which is usually combined with engineering creating concepts of electrical engineering. In this field, people study the movement and control of electrons under several conditions. The purpose of this study is to create functional software or hardware, thus can be used in a real-life environment. The reason for this development in electrical engineering is to create support for humankind in either professional or personal life. Now, electrical engineering strive to find a better, more compact, efficient, effective, and functional electronic system.

An electronic device is defined as a functional hardware to process information or control system by controlling the flow of electrical energy. The most common example of an electronic device is a mobile phone. It is one of the most common gadgets found around the world. Other than that, the examples are refrigerators, air conditioner, television, dispenser, and many more. The world is now inseparable from the electrical engineering process. Most of the equipment used in this modern world is based on the electronics concepts.

In the wide field of electric, there are multiple concepts that are being utilized to create a desired outcome. The examples are conductors and insulators, voltage and resistance, Ohm's law, resistance and resistivity, energy and power, electrostatics, electric fields, magnetostatic, and many more. Each of these concepts is studied for its application in an electric device. The ability to modify electrical variables into an information or decision is the main aim of the study.

Electronic devices are built on top of an electrical circuit. The connection between each electric component will perform a designed functionality. Each of the components has their own utility in a greater view of a device. Such components are transistors, resistors, diodes, switches, capacitors, inductors, and many others. Those components will be assembled and connected to each corresponding component. The sophisticated form of an electric circuit is a Printed Circuit Board (PCB) or Integrated Circuit (IC). These examples simplify the madness created by adjoining cables connecting different components. It replaced cable's conductivity with copper field inside the base board. Enabling a complex system to be realized in a simplified form yet retaining the functionality.

The connection between all the components in an electric circuit can be represented as a graph. In graph theory, there are 2 parts that construct a graph, node and edges. In the representation of electric circuit, the electric components can be represented as a node in the graph. The connection between each component, usually connected by cables or copper field is represented as edges. With this representation, the electrical circuit can be studied in multiple disciplines, including graph theory. One of the graph theories which can be applied is the planar graph. In a single layer electronic circuit, or a 2-dimensional electronic circuit, the connection between each of the components should not be in one part. What this means is that the path should not be crossing other paths. The crossing paths could trigger a short-circuit that will damage the electric devices. In realization of this case, the planar graph theory can be applied to solve the problem.

III. Graphs

A. Graph

A graph is a mathematical object in discrete mathematics. It is used to represent the connections between discrete objects. As the main concept is to define connections between objects, it is widely used in other fields. Such as electronic circuits, drawing chemical compound, informatic, biology, and many more. A graph is formed by 2 components, vertex and edges. Vertex is a node which represents the object in a graph. Whereas edges are the connection between nodes that is usually represented in a form of line.

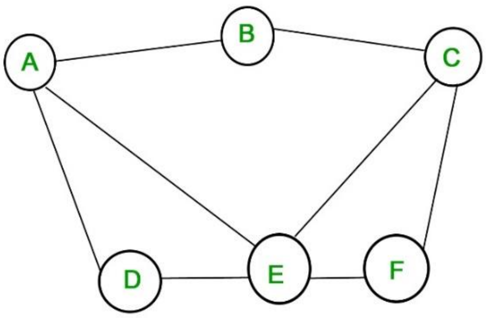


Figure 1. Graph

In mathematics, graphs can be presented or defined in many forms such as matrix, set, and figure. In the form of a set, graph is represented as G = (V, E). Where V is a not empty sets of vertices or nodes such as {v1, v2, …, v3}. The vertices set should not be empty as it is the main component of a graph. But the E or edges set can be empty or contain multiple edges such as {e1, e2, e2, …, en}. The definition of e1 could be defined as v1-v2. Which means the e1 edge connects vertex v1 and vertex v2.

There are many types of graphs. One of the examples is a simple graph which does not contain either a looping edge or a double edge. A looping edge is a connection between the same nodes. Where a double edge is 2 edges that connect the same nodes. The graph which contains either looping edge or double edge is called unsimple graph. A graph that only contains a double edge is called multi-graph and a graph that only contains a looping edge is called pseudo-graph. Another type of graph is undirected graph and directed graph. A directed graph is graph with directed edge to one specific node.

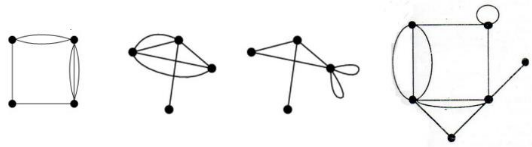
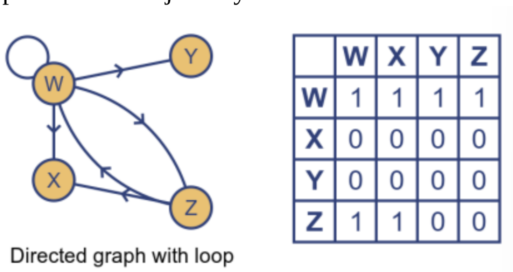


Figure 2. Unsimple Graphs[1]

There are also terms that are generally used in graphs, such as incidence, adjacent, isolated vertex, null graph, and degree. One unique type of graph is a bipartite graph. It is the graph that the vertices set can be divided into 2 sets. Where every edge in the graph will connect a vertex in V1 to a vertex in V2.

In the form of matrix, graph can be represented as incidence matrix and adjacent matrix. An incidence matrix is a matrix which defines the connection between two nodes. The horizontal and vertical variables are both list of the nodes. In another form of list, a graph can also be represented as adjacency list.



	W	X	Y	Z
W	1	1	1	1
X	0	0	0	0
Y	0	0	0	0
Z	1	1	0	0

Figure 3. Adjacency Matrix[1]

There are 2 terms that define the similarity between 2 graphs that are widely used and will be used further in this study. Isomorphic are the terms for 2 graphs which are the same yet retaining a different geometry form. The meaning behind the same graph is that in each of one node in one graph there is a corresponding node in the other graph which connects to the same nodes (in different geometric object). It can be seen below that G1, and G2 are isomorphic whereas G3 is not. Each node in G1 can be represented in each node in G2. G1 or G2 can be redrawn to be in the same geometric form as the other graph.

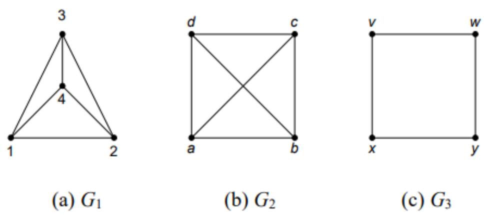


Figure 4. Isomorphics Graphs[1]

The other term that is generally used is homeomorphic. It is the term used to describe 2 graphs which can be represented in the same geometric form but contain a different number of nodes. A graph can also be defined as homeomorphic when both graphs can be obtained through edges division.

B. Planar Graph

The definition of planar graph is used for a graph that can be drawn in a 2 dimensional or flat plane without any edges crossing each other. Plane graph is used when a planar graph is drawn without any edges crossing.

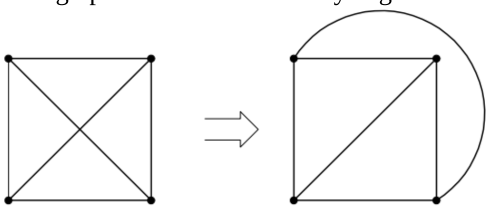


Figure 5. Planar Graphs[1]

With this unique characteristic of planar graph, it is very widely used to define connections in many forms. Such as electronics component in Printed Circuit Board (PCB) and Integrated Circuit (IC). The connection between components in PCB and IC is based on a conductive copper base. In order to avoid any interference, there should not be a crossing path. More examples of utilization of planar graph is urban planning, computational algorithms, telecommunications, and many more.

Euler's formula of planar graph defines the connection between the number of edges, nodes, and regions. When a plane graph is drawn, it divides the flat surface into several regions, including the outer region.

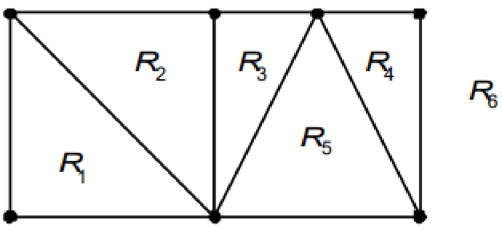


Figure 6. Plane Graphs[1]

Euler's formula defines the number of edges (*e*), the number of nodes (*n*), and the number of regions (*f*) as

$$n - e + f = 2$$

Take the example of the figure. In the drawn figure, the *e* = 11, *n* = 7, and *f* = 6. It always satisfies Euler's formula where 7-11+6 = 2. As Euler's formula is always true, there is Euler's inequality. Which defines a planar graph which contains *f* number of regions, *n* number of nodes, and *e* number of edges with the minimum number of edges is 2 (e >2), this inequality is always true.

$$e \leq 3n - 6$$

This inequality, with Euler's formula can be used to determine the planarity of a graph only using the graph's specifications. Yet, there is a special case in which Euler's inequality cannot define the actual planarity of a graph. The case is in Kuratowski's Graph.

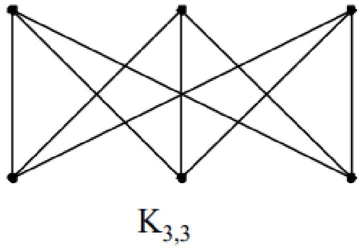


Figure 7. $K_{3,3}$ **Kuratowski Graph** [1]

This graph satisfies the Euler's inequality where *e* = 9, *n* = 6, and 9 ≤ 3*6 -6. Yet, the graph is not a planar graph. Therefore, the Euler's inequality was developed to the form of

$$e \leq 2n - 4$$

With the assumption of each region is surrounded by at least 4 edges. With this development, the Kuratowski's graph is determined as a non-planar graph.

Although Euler's formula and Euler's inequality can be used to determine the planarity of a graph, there is another theorem which can determine in more decisive way. It is founded by Kazimierz Kuratowski which was one of the leading representatives of the Warsaw School of Mathematics. He was born on February 2, 1896, and later died on June 18, 1980. The Kuratowski theorem defines a planar graph as a graph that does not contain a sub-graph that is either isomorphic or homeomorphic with any of the Kuratowski graphs. A sub-graph is a part of a graph which can be obtained through deletion of several nodes or edges.

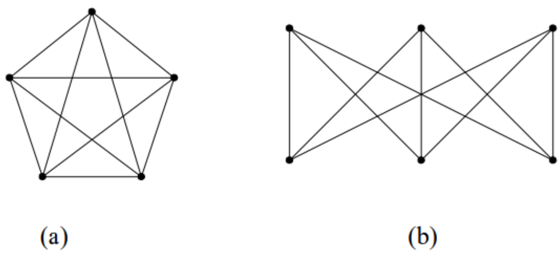


Figure 8. Kuratowski Graphs[1]

There are 2 Kuratowski graphs, K_5 which is in figure a above and $K_{3,3}$. K_5 is a graph formed by 5 nodes where each of the nodes is connected to every node. Where $K_{3,3}$ is a bipartite graph with 6 nodes where each node in different sets connects to each node in the other set. There are several characteristics of Kuratowski graphs. First, both Kuratowski graphs are regular graphs. Second, both Kuratowski graphs are non-planar graphs. The deletion of either a edge or a node from Kuratowski graphs will alter the graph to be a planar graph. The first Kuratowski graph, K_5 is a non-planar graph with minimum number of nodes. While

$K_{3,3}$ is a planar graph with the least number of edges. The Kuratowski theorem is used to define the planarity of a graph more definitely, it is preferable than utilizing Euler's formula and Euler's inequality. Take the example below, where graph G contains a sub-graph of $K_{3,3}$. The sub-graph can be obtained through deletion of a-c, a-b, b-c, f-d, f-e, and f-d edges. Therefore, it will form the same geometrical form as $K_{3,3}$ which means that it is isomorphic with $K_{3,3}$. It is determined clearly that the graph G is not a planar graph.

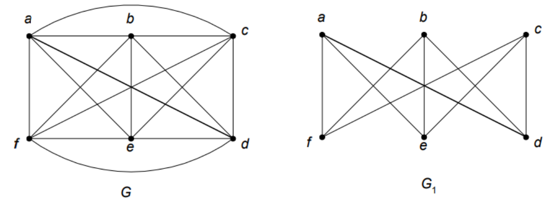


Figure 9. Isomorphic Kuratowski Graphs[1]

IV. Circuit Routing Strategy

As the planar graph concept is very applicable to the electric engineering domain, it is mainly used to routing components in Integrated Circuits (IC). But the application does not only apply to the complex design of an IC. The concept can also be applied to a single layered electronic design or single-sided Printed Circuits Boards (PCB). This type of design is mainly used in simple electronic devices such as household electronics, toys, lighting systems, and many more. The main reason of the usage of a single layered PCB is it is cost effective, easy to design, and more reliable. Yet, it also has several limitations such as functionality, space constraints, and performance when compared to layered PCB or IC.

There are several procedures in the circuit routing strategy. The main purpose is to represent the electric circuit in the form of a graph and determine whether the circuit is planar or not. The first step is to gather all the electrical circuit's components. The components that are going to be used are defined before placing it in the electrical circuit. It can be in the form of resistors, capacitors, inductors, diodes, switches, ICs, batteries, or many more. Then, the next step is to define the connections between each electrical component. To obtain the connection, the electric circuit can be manually drawn without the consent of intersection of conductive path. Otherwise, using digital software to note and draw the electric circuit.

After the data is fully gathered, the next step is to determine whether it can be realized in a single-layer electronic circuit or not by using the graph theory. There are 2 theories that can be used. That is using the Euler's formula and Euler's inequality or using Kuratowski Theorem. When using Euler's formula and Euler's equality, the number of nodes, symbolized as n, the number of edges, symbolized as e, and the number of regions, symbolized as f should always satisfy the rule which are

$$n - e + f = 2$$
$$e \leq 2n - 4$$
$$e \leq 3n - 6$$

These three rules must be satisfied in order to determine the planarity of a graph. Yet, it is still not defined as bold. In order to determine in a decisive manner, Kuratowski theorem can be applied. The graph should be searched for a sub-graph which is isomorphic or homeomorphic to any of Kuratowski graphs. Therefore, it can be defined clearly, whether a graph is planar or not. Meaning it can be realized as a single-layer electric circuit or not.

V. Electronic Circuit Design Program

As the realization of the circuit routing strategy, a computer program is developed. The algorithm of the program is based on the algorithm used in the routing strategy before. It is incorporating technologies with several other fields to fulfill a demand of effectiveness. It is hoped that this program will help the determination of realization of an electric circuit in single-layer PCB.

This program will be able to take user's input of components of the electrical circuit and the connection of each. Then the program will be able to determine whether the circuit is a planar graph or not. If the graph is planar and realizable in single-layer electrical circuit, the program will printout a plane graph with the components and connections defined.

This program is developed with Python 3 programming language. Some libraries were added in order to help the functionality such as network, matplotlib, and itertools. The program is divided into several functions. They are input_components, input_connections, is_k5_k33, and check_planarity_and_draw. Each of the function has their own purpose and part in contributing to the algorithm. When the program is being run, the first function called is input_components, followed by input_conenctions, and check_planarity_and_draw. In the function of check_planarity_and_draw, the function is_k5_k33 is called.

```python
def input_components():
    print("Type the Electrical Components in the circuit. Type 'done' when finished.")
    components = []
    while True:
        component = input("Component: ").strip()
        if component.lower() == 'done':
            break
        if not component:
            continue
        if component in components:
            print(f"Error: Component {component} already exists in the Circuit")
            continue
        components.append(component)
    return components
```

Figure 10. Source Code of Function Input_Components

In the figure above, it can be seen how the function input_connections is defined. It started with the printout function of the command for the user's input. Then, declaring an array variable of components. While the user's input is not "done", the program will repeat the input from the user. This enables the user to input various amounts of components. An error message will be displayed upon inputting a stated component. This mitigates a double name for a single component in a circuit.

```python
def input_connections(components):
    print("\nAvailable components in the Circuit:", ", ".join(components))
    print("\nDefine the connections between components. For each connection, enter two 
components separated by a space.")
    print("Type 'done' when finished.")
    connections = []
    while True:
        connection = input("Connection (example: R1 C1): ").strip()
        if connection.lower() == 'done':
            break
        parts = connection.split()
        if len(parts) != 2:
            print("Invalid format.")
            continue
        if not all(part in components for part in parts):
            invalid_components = [part for part in parts if part not in components]
            print(f"Invalid component(s): {', '.join(invalid_components)}")
            print("Available components:", ", ".join(components))
            continue
        if tuple(parts) in connections or tuple(reversed(parts)) in connections:
            print(f"Connection between {parts[0]} and {parts[1]} already exists in the 
Circuit.")
            continue
        connections.append(tuple(parts))
    return connections
```

Figure 11. Source Code of Function Input_Connections

The figure above shows the source code of the function input_connections. The first algorithm is printing the available components in the circuit. It takes the data from input_components and displays it to inform the user. An array variable called connections is also defined and will be filled. The input format should be "Component1 Component2." When the format is invalid, or the component is not defined, or else the connections already exist, the program will send out an error message.

```python
def is_k5_or_k33(graph):
    if len(graph) >= 5:
        for clique in nx.find_cliques(graph):
            if len(clique) >= 5:
                subgraph = graph.subgraph(clique[:5])
                if nx.is_isomorphic(subgraph, nx.complete_graph(5)):
                    return True

    if len(graph) >= 6:
        for nodes in itertools.combinations(graph.nodes(), 6):
            subgraph = graph.subgraph(nodes)
            if nx.is_bipartite(subgraph):
                try:
                    left, right = nx.bipartite.sets(subgraph)
                    if len(left) == 3 and len(right) == 3:
                        if len(subgraph.edges) == 9:
                            return True
                except:
                    continue

    return False
```

Figure 12. Source Code of Function Is_K5_Or_K33

In the function is_k5_or_k33 above contains the algorithm to check the planarity of the graph. The Kuratowski theorem is used upon determining planarity. The graph of the circuit will be searched for sub-graph that is isomorphic or homeomorphic with Kuratowski theorem. The algorithm will search for a sub-graph that is isomorphic with K_5 when the node numbers is greater or equal 5. If it is found, it will return true. For graph with 6 or more nodes, the sub-graph will be checked whether it is bipartite or not. If it is, then the number of nodes in the left sets and right sides are checked. When the number of nodes shows 3 on each side, the number of edges is checked. 9 edges will show that the graph is a isomorphic with $K_{3,3}$, determining that it is not a planar graph.

```python
def check_planarity_and_draw(components, connections):
    G = nx.Graph()
    G.add_nodes_from(components)
    G.add_edges_from(connections)

    is_planar, _ = nx.check_planarity(G)
    contains_k5_k33 = is_k5_or_k33(G)

    if is_planar and not contains_k5_k33:
        print("\nThe circuit is a planar graph. Generating illustration...")
        try:
            pos = nx.planar_layout(G)
        except:
            pos = nx.spring_layout(G)

        plt.figure(figsize=(10, 8))
        nx.draw(G, pos, with_labels=True, node_color='lightblue',
                edge_color='gray', node_size=2000, font_size=10)
        plt.title("Electronic Circuit Diagram")
        plt.show()
    else:
        print("\nThe circuit is not a planar graph.")
        if contains_k5_k33:
            print("The circuit contains either a K5 or K3,3 subgraph.")
```

Figure 13. Source Code of Function Check_Planarity_And_Draw

The first part of function check_planarity_and_draw is shown above. In the main function of check_planarity_and_draw, the graph will be determined as planar or not by calling the function is_k5_or_k33. If the result is the graph is planar, the program will output an illustration of the graph. The algorithm starts with combining the components and connections data from user's input earlier. Then determining the planarity of graph using library networkx tools and function is_k5_k33. If the result is the graph is planar, the program will proceed to draw figures using networkx function library. Else if the graph is not planar, the program will print out a message that state the circuit is not a planar graph.

```python
if __name__ == "__main__":
    print("Circuit Illustration Program\n")
    components = input_components()
    if not components:
        print("No components were defined. Exiting program.")
    else:
        connections = input_connections(components)
        if not connections:
            print("No connections were defined. Exiting program.")
        else:
            check_planarity_and_draw(components, connections)
```

Figure 14. Source Code of Main Function

The last part of this function shown above is the main part. When the program is first started, it will display the message "Circuit Illustration Program." It continues to the function input_components. When there is no input of the components, the program will exit itself. Else, it will proceed to the input_conenction function. The program will also exit when there are no connections. As the variables needed are already covered, the program will proceed to the check_planarity_and_draw function.

VI. Cases of Usage

1. Random Graph/Circuit
   To create a random graph, the components and connections to input are shown in the table below.

Components	R1, C1, T1
Connections	R1-C1, R1-T1, C1-T1

The input process is shown in the picture below.

```
Circuit Illustration Program

Type the Electrical Components in the circuit. Type 'done' when finished.
Component: R1
Component: C1
Component: T1
Component: done

Available components in the Circuit: R1, C1, T1

Define the connections between components. For each connection, enter two comp
onents separated by a space.
Type 'done' when finished.
Connection (example: R1 C1): R1 C1
Connection (example: R1 C1): R1 T1
Connection (example: R1 C1): C1 T1
Connection (example: R1 C1): done
```

Figure 15. Input Case 1

The output picture is shown in the picture below

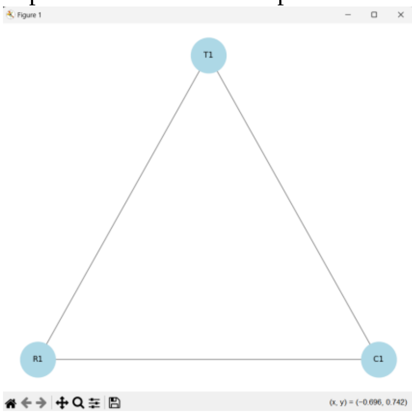


Figure 16. Output Case 1

2. Complex Planar Graph/Circuit
   To create a random graph, the components and connections to input are shown in the table below.

Components	R1, R2, R3, R4, R5, C1, C2, C3, L1, L2, D1, D2
Connections	R1 - R2, R2 - R3, R3 - R4, R4 - R5, R5 - R1, R1 - C1, R2 - C2, R3 - L1, R4 - L2, R5 - D1, C1 - C2, C2 - L1, L1 - L2, L2 - D2, D2 - C1

The output illustration picture is shown in the picture below.

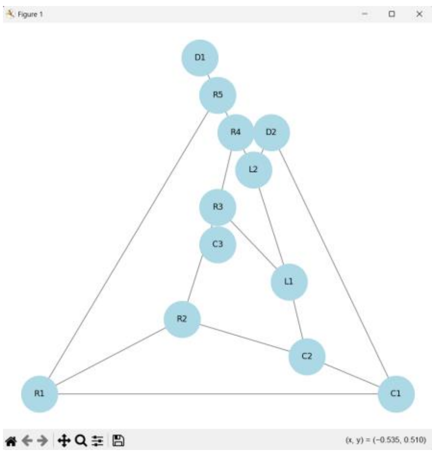


Figure 17. Output Case 2

3. Non-Planar Graph/Circuit (Isomorphic with K_5)
   To create a random graph, the components and connections to input are shown in the table below.

Components	R1, R2, R3, R4, R5
Connections	R1 - R2, R1 - R3, R1 - R4, R1 - R5, R2- R3, R2- R4, R2 – R5, R3 - R4, R3- R5, R4 - R5.

The output illustration picture is shown in the picture below.

```
Circuit Illustration Program

Type the Electrical Components in the circuit. Type 'done' when finished.
Component: R1
Component: R2
Component: R3
Component: R4
Component: R5
Component: DONE

Available components in the Circuit: R1, R2, R3, R4, R5

Define the connections between components. For each connection, enter two comp
onents separated by a space.
Type 'done' when finished.
Connection (example: R1 C1): R1 R2
Connection (example: R1 C1): R1 R3
Connection (example: R1 C1): R1 R4
Connection (example: R1 C1): R1 R5
Connection (example: R1 C1): R2 R3
Connection (example: R1 C1): R2 R4
Connection (example: R1 C1): R3 R4
Connection (example: R1 C1): R3 R5
Connection (example: R1 C1): R4 R5
Connection (example: R1 C1): R2 R5
Connection (example: R1 C1): done

The circuit is not a planar graph.
The circuit contains either a K5 or K3,3 subgraph.
```

Figure 18. Input & Output Case 3

4. Non-Planar Graph/Circuit (Isomorphic with $K_{3,3}$)
   To create a random graph, the components and connections to input are shown in the table below.

Components	R1, R2, R3, L1, L2, L3, C1, C2
Connections	R1 - L1, R1 - L2, R1 - L3, R2 - L1, R2 - L2, R2 - L3, R3 - L1, R3 - L2, R3 - L3, R1 - C1, R2 - C1, R3 - C1, C1 - C2, C2 - L1, C2 - L2, C2 - L3

The output illustration picture is shown in the picture below.

```
Available components in the Circuit: R1, R2, R3, L1, L2, L3, C1, C2

Define the connections between components. For each connection, enter two comp
onents separated by a space.
Type 'done' when finished.
Connection (example: R1 C1): R1 L1
Connection (example: R1 C1): R1 L2
Connection (example: R1 C1): R1 L3
Connection (example: R1 C1): R2 L1
Connection (example: R1 C1): R2 L2
Connection (example: R1 C1): R2 L3
Connection (example: R1 C1): R3 L1
Connection (example: R1 C1): R3 L2
Connection (example: R1 C1): R3 L3
Connection (example: R1 C1): R1 C1
Connection (example: R1 C1): R2 C1
Connection (example: R1 C1): R3 C1
Connection (example: R1 C1): C1 C2
Connection (example: R1 C1): C2 L1
Connection (example: R1 C1): C2 L2
Connection (example: R1 C1): C2 L3
Connection (example: R1 C1): done

The circuit is not a planar graph.
The circuit contains either a K5 or K3,3 subgraph.
```

Figure 19. Input & Output Case 4

VII. CONCLUSION

By this research, it can be concluded that graph theory is very applicable to many other sectors. Sectors such as electric circuiting benefited heavily through planar graph concept. In building a single-layer electric circuit, the connections between each component should be able to be represented as a planar graph. By this ability, it means that there is no conductive path that is crossing each other, and the chance of short circuit is lowered. Realizing this concept in real life problems can be done through digital programming which automatizes the planar determination process. The best planar graph theory to be applied in the program is Kuratowski theory as it defines the planarity of a graph in a more decisive manner than other theory.

VIII. APPENDIX

- Source Code: https://github.com/ivan-wirawan/CircuitRoutingProgram
- Video: https://youtu.be/_3lUNsYeSzA

IX. ACKNOWLEDGMENT

I want to acknowledge and express my sincere gratitude to Mr. Rila Mandala, my instructor in discrete Mathematics. He has explained the material clearly in the lecture given. By gaining understanding through the lectures I attend, I can develop my knowledge into the form of this research paper.

I also want to thank Mr. Rinaldi Munir, whose website is the primary source of this paper's material. Mr. Rinaldi Munir is also a lecturer in discrete mathematics. I am sincerely thankful for the material provided freely and easy to access on Mr. Rinaldi Munir Website. By that, the learning of discrete mathematics topic has become easier and enjoyable.

REFERENCES

[1] R. Munir, (2024). "IF2120 Matematika Diskrit." Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/matdis.htm

[2] D.Y. Ariyanti, (2019, December). "Application of Planar Graph to Design Printed Circuit Board." Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2019-2020/Makalah2019/13518142.pdf

[3] J. A. Svoboda, R.C. Dorf, "Introduction to Electric Circuits." 9th ed. Danvers, 2014. Accessed: Dec 31, 2024. [Online]. Available: https://libraryguides.vu.edu.au/ieeereferencing/books-ebooks

[4] L. Toscano, S. Stella, E.Milloti. "Using Graph Theory for Automated Electric Circuit Solving." Available: https://arts.units.it/bitstream/11368/2836192/5/article-LC.pdf

[5] M. Tooley. "Electronic Circuits: Fundamentals and Applications." 2nd ed. Oxford, 2002. Accessed: Dec 31, 2024. [Online]. Available: https://fmipa.umri.ac.id/wp-content/uploads/2016/03/Tooley_Electronic_Circuits_-_Fundamentals_and_ApBookZZ.org_.pdf

[6] J. A. Bondy, U.S. R. Murty. "Graph Theory With Applications." 2nd ed. New York. 1976. Accessed: Jan 1, 2025. [Online]. Available: https://snscourseware.org/snsrcas/files/CW_5d16efa85c2e3/BondyMurtyGTWA-compressed.pdf

DECLARATION

Hereby, I declare that this paper I have written is my own work, not a reproduction or translation of someone else's paper, and not plagiarized.

Bandung, 8 Januari 2024

Ivan Wirawan, 13523046