AN IMPROVED ALGORITHM FOR CHANNEL ALLOCATION ON DIRECT SEQUENCE SPREAD SPECTRUM

HANDRIZAL

MASTER OF SCIENCE (COMPUTER)

UNIVERSITI MALAYSIA PAHANG

AN IMPROVED ALGORITHM FOR CHANNEL ALLOCATION ON DIRECT SEQUENCE SPREAD SPECTRUM

HANDRIZAL

Thesis submitted in fulfillment of the requirements
For the award of the degree of
Master of Science in Computer

Faculty of Computer Systems & Software Engineering
UNIVERSITI MALAYSIA PAHANG

JULY 2011

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Master of Science in Computer.

Signature:

Name of Supervisor: DR. NORAZIAH AHMAD

Position: SENIOR LECTURER OF FACULTY COMPUTER SYSTEMS & SOFTWARE ENGINEERING

Date: JULY 07, 2011

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries, which have been duly acknowledged. The thesis has not been accepted for any degree and is not concurrently submitted for the award of other degree.

Signature:
Name: HANDRIZAL
ID Number: MCC 09001
Date: JULY 07, 2011

ACKNOWLEDGMENTS

I am grateful and would like to express my sincere gratitude to my supervisor Dr. Noraziah Ahmad for her ideas, invaluable guidance, continuous encouragement and constant support in making this research possible. She always impressed me with her outstanding professional conduct, her strong conviction for science, and belief that a Master program is only started of a life-long learning experience. I appreciate her consistent support from the first day I applied to a graduate program to these concluding moments. I am truly grateful for her progressive vision about my training science, her tolerance of my naive mistakes, and her commitment to my future career. I also would like to express very special thanks to my former co-supervisor Assoc. Prof. Dr. Ahmed N. Abd. Alla for his suggestion and co-operation throughout the study. Thank you for the time spent proof reading and correcting my many mistakes.

My sincere thanks go to all my laboratory mates and members of the staff of the Computer and Software Engineering Department, Universiti Malaysia Pahang (UMP), who helped me in many ways and made my stay at UMP pleasant and unforgettable. Many specials thank go to a member of the wireless network research group for their co-operation, inspirations and support during this study.

I acknowledge my sincere indebtedness and gratitude to my parents, Lukman and Lamnur, my wife, Rini Hartati and my daughter Intan Handayani Rizal for their love, dream and sacrifice throughout my life. I acknowledge the sincerity of my brothers, sisters for their sacrifice, patience, and understanding that were inevitable to make this work possible. I cannot find the appropriate words that could properly describe my appreciation for their devotion, support and faith in my ability to attain my goals. Special thanks should be given to my committee members. I would like to acknowledge their comments and suggestion, which was crucial for the successful completion of this study.

ABSTRACT

Graph coloring is an assignment a color to each vertex, which each vertex that adjacent is given a different color. Graph coloring is a useful algorithm for channel allocation on Direct Sequence Spread Spectrum (DSSS). Through this algorithm, each access point (AP) that adjacent gives different channels based on colors available. Welsh Powell algorithm and Degree of saturation (Dsatur) are the popular algorithms being used for channel allocation in this domain. Welsh Powell algorithm is an algorithm that tries to solve the graph coloring problem. Dsatur algorithm is an algorithm coloring sorted by building sequence of vertices dynamically. However, these algorithms have its weaknesses in terms of the minimum number of channel required. In this study, channel allocation called Vertex Merge Algorithm (VMA) is proposed with aim to minimize number of required channels. It is based on the logical structure of vertex in order to a coloring the graph. Each vertex on the graph arranged based on decreasing number of degree. The vertex in the first place on the set gives a color, and then this vertex is merged with non-adjacent vertex. This process is repeatedly until all vertices colored. The assignment provides a minimum number of channels required. A series of an experiment was carried out by using one computer. Vertex Merge Algorithm (VMA) simulation is developed under Linux platform. It was carried out in Hypertext Preprocessor (PHP) programming integrated with GNU Image Manupulation Program (GIMP) for open and edit image. The experimental results show that the proposed algorithm work successfully in channel allocation on DSSS with the minimum of channels required. The average percentage reduction in the number of required channels among the VMA, Dsatur algorithm and Welsh Powell algorithm in the simple graph is equivalent to 0.0%. Meanwhile between the VMA and Dsatur algorithm in the complex graph is equivalent to 18.1%. However, VMA and Welsh Powell algorithm is not compared in the complex graph since its drawback in terms of not fulfill the graph coloring concept. This is because there are two adjacent vertices have the same color. Overall, even there is no reduction for number of required channel among VMA, Dsatur algorithm and Welsh Powell algorithm in the simple graph, but the outstanding significant contribution of VMA since it has reduction in the complex graph.

ABSTRAK

Pewarnaan graf merupakan pemberian warna untuk setiap verteks, untuk verteks yang berdekatan diberikan warna yang berbeza. Pewarnaan graf merupakan suatu algoritma yang digunakan bagi penempatan saluran pada Direct Sequence Spread Spectrum (DSSS). Melalui algoritma ini, setiap akses point (AP) yang berdekatan diberikan saluran yang berbeza berdasarkan warna tertentu. Algoritma Welsh Powell dan algoritma Darjah Kejenuhan (Dsatur) merupakan algoritma-algoritma terkenal yang telah digunakan untuk penempatan saluran dalam bidang ini. Algoritma Welsh Powell merupakan algoritma yang cuba menyeselaikan masalah pewarnaan graf. Algoritma Dsatur adalah algoritma pewarnaan graf yang dibuat dengan urutan verteks secara dinamik. Walaubagaimanapun, algoritma-algoritma ini mempunyai kelemahan dari segi jumlah minimum saluran yang diperlukan. Dalam kajian ini, dicadangkan penempatan saluran yang dikenali sebagai Algoritma Vertex Gabung (VMA) dengan tujuan untuk meminimumkan jumlah saluran yang diperlukan. Hal ini berdasarkan susunan logikal verteks untuk membentuk pewarnaan suatu graf dalam penempatan saluran. Setiap verteks dalam graf disusun berdasarkan penurunan jumah darjah. Verteks dalam turutan pertama suatu set diberikan warna, kemudian verteks tersebut digabungkan dengan verteks yang tidak berdekatan. Proses ini terus berulang sehingga kesemua verteks diberikan warna. Ketetapan ini telah menyediakan jumlah saluran minimum yang diperlukan. Suatu siri eksperimen telah dijalankan dengan menggunakan sebuah komputer. Simulasi Algoritma Verteks Gabung (VMA) telah dibangunkan di bawah platfom Linux. Ia telah dibina dengan bahasa pengaturcaraan Prapemroses Hiperteks (PHP) serta berintegrasikan GNU Image Manipulation Program (GIMP) untuk memaparkan dan mengemaskinikan gambar. Keputusan eksperimen menunjukkan bahawa algoritma yang dicadangkan telah berjaya dalam penempatan saluran pada Direct Sequence Spread Spectrum (DSSS) dengan jumlah saluran minimum yang diperlukan. Purata peratusan penurunan jumlah saluran di antara VMA, Dsatur algoritma dan Welsh Powell algoritma dalam graf sederhana adalah bersamaan dengan 0.0%. Sementara itu, di antara VMA dan Dsatur algoritma dalam graf kompleks adalah bersamaan dengan 18.1%. Namun, VMA dan Welsh Powell algoritma tidak dibandingkan dalam kompleks graf disebabkan kelemahan dalam hal tidak memenuhi konsep pewarnaan graf. Hal ini disebabkan terdapat dua verteks bersebelahan memiliki warna yang sama. Secara keseluruhan, tidak ada penurunan jumlah saluran yang diperlukan antara VMA, Dsatur algoritma dan Welsh Powell algoritma di dalam graf sederhana. Walaubagaimanapun, sumbangan penting VMA adalah penurunan jumlah saluran yang diperlukan di dalam kompleks graf.

TABLE OF CONTENTS

CHAPTER 5 CONCLUSION AND RECOMMENDATIONS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

$A(G)$	Adjacent matrix of graph
c	Color of graph
$C(V)$	Admissible coloring minimizing
E	Edge of graph
E_G	Set edge of graph
F	Set of colors
F'	Subset of colors
G	A graph
M	Set of merge vertex
i	Index of vertex or edge
I	Set index of element graph
j	Index of vertex or edge
k	Number of colors used
n	Number of vertex
$n(G)$	Normal order of graph
$m(G)$	Normal size of graph
l	Initial row in the VMT
R	Set of all possible $\lvert V \rvert$ size integers valued rows
S	Set of initial rows the VMT
V_G	Set vertex of graph
$\chi(G)$	Chromatic numbers of graph

LIST OF ABBREVIATION

AP	Access Point
DSatur	Degree of Saturation
DSSS	Direct Sequence Spread Spectrum
EOG	Eye of GNOME
FCC	Federal Communications Commission
GHz	Gigahertz
GIMP	GNU Image Manipulation Program
GNU	GNU's Not Unix
GPL	General Public License
IEEE	Institute of Electronics and Electrical Engineers
ISM	Industrial Scientific and Medical
Mbps	Megabits per second
MHz	Megahertz
PNG	Portable Network Graphic
RGB	Red Green Blue
SS	Spread Spectrum
USA	United State of America
VMA	Vertex Merge Algorithm
VMO	Vertex Merge Operation
VMT	Vertex Merge Table
WLAN	Wireless Local Area Network

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The spectacular development of network and the internet has a big impact to the companies in various types and sizes. The advanced wireless technologies support the development of network, the internet and intranet capability for the mobile workers, isolated area and temporary facilities. Wireless expands and increases the capability of computer networking. The new technologies enable the wireless networking as one of the access in higher velocity and qualified for the computer network and the internet.

The wireless networks based on the Institute of Electrical and Electronics Engineers (IEEE) 802.11 standards (IEEE, 2007). This technology presents important disadvantages if compared with other radio networks, as a shorter range, security issue, difficulties providing inter-network or inter-operator roaming. However, the networks based on the IEEE 802.11 are unquestionably being chosen by most of the users (Gracia, 2005). Its popularity comes noticeable, beyond the evident advantages of the radio networks in contrast to the wired networks.

Nowadays, it is a common thing to see the coexistence of many Wireless Local Area Network (WLAN) networks in densely populated areas, either of private or of public nature (Hot-Spots), corporative. Domestic users use it to avoid the installation of new wires in their homes and to communicate with other users, in offices or university campuses are used to provide access to the internet or corporative intranets to their employees or students and, lately, operators and Internet Service Provider (ISP) have found a new market offering this access in public places, like hotels, airports or convention centers. With a high density of nodes, the presence of interference increases causing the performance perceived by users to degrade. In order to reduce the effect of interferences in cellular networks, a channel planning is traditionally used. In cases of 802.11b and 802.11g WLAN networks, frequency channels are a scarce resource, since we can only count on three non overlapping channels. Therefore, the Channel Allocation Problem (CAP) is an important issue to solve.

1.2 CHANNEL ALLOCATION

Channel allocation can be defined as choosing, which channel the access point should use to communicate with the wireless terminals in its subnet without interfering with the transmissions from the other's access points (Mahonen et al. 2004). Understanding of channel allocation is very important because it relates to the overall capacity of wireless local area networks (Carpenter, 2008).

Channel allocation in WLAN 802.11 networks is studied as a part on the design of multicellular WLANs working in infrastructure mode. A good design can be evaluated according to two basic requirements: full coverage of the required area and the provision of a capacity suitable to support the traffic that is generated, without degrading the service as the number of user's increases. Although there is an endless list of parameters to consider, the requirements mentioned above can be obtained with an exhaustive selection of Access Points (*APs*) locations and the proper set of channels and power levels.

Several research articles have been published regarding channel allocation. Among them were those by (Al Mamun et.al. 2009; Chen et.al. 2008; Duan et.al. 2010; Mahonen et.al. 2004; Malone et.al. 2007; Raj, 2006; Riihijarvi et.al. 2005; 2006; Yuqing et.al. 2010; Yue et.al. 2010; and Zhuang et.al. 2010) Those articles revealed that channel allocation for DSSS is one of the current issues that still unsolved in channel allocation. Therefore, the study on this basis is initiated.

Wireless Local Area Networks (WLANs) operate in unlicensed portions of the frequency spectrum allotted by a regulatory body, like the Federal Communications Commission (FCC) in the United States of America (Goldsmith, 2005; IEEE, 2007). Each WLAN standard (802.11/a/b/g) defines a fixed number of channels for use by access point (AP) and mobile users. For example, the 802.11b/g standard defines a total of 14 frequency channels of which 1 through 11 are permitted in the United States of America (Guizani, 2004). Actually, channel represents the center of frequency. There is only 5 MHz separation between the center frequencies. The signal falls within about 15 MHz of each side of the center frequency. Consequently, an 802.11b/g signal overlaps with several adjacent channel frequencies. Therefore, only three channels (channels 1, 6, and 11) available for use without causing interference (Gracia, 2005; Rackley, 2007; Theodore, 2001).

The Wireless LAN standard 802.11b and 802.11g in the process of distributing data using Direct Sequence Spread Spectrum (DSSS) technology (Goldsmith, 2005). DSSS works by taking a data stream of zeros and ones and modulating it with a second pattern, the chipping sequence. Various other electronic devices in a home, such as cordless phones, garage door openers, and microwave ovens, maybe use same frequency range. Any such device can interfere with a WLANs network, slowing down its performance and potentially breaking network connections (Perez, 1998; Theodore, 2001).

1.3 PROBLEM STATEMENT

According to the overview in the Section 1.1, DSSS system only has three none overlapping channels are located (Rackley, 2007; Theodore, 2001). When more than three access points (*APs*) are in same location, it may cause interference with another AP. Interference in communications is anything, which alters, modifies, or disrupts a signal as it travels along a channel between a source and a receiver (James, 2008; Stallings, 2003). In addition, interference decreased the network performance.

Thus, some of the research problems that arise in channel allocation can be stated as follows:

- How does allocate channel while there is more than three access points in same location?
- How to reduce the number of channels while there is more than three access points in same location?

Figure 1.1 shows the channel allocation problem when there exist more than three access points in the same area.

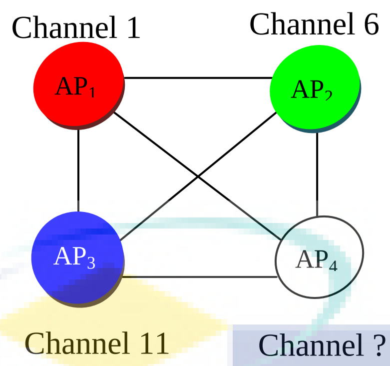


Figure 1.1: An example channel allocated in DSSS

From Figure 1.1 it can be seen that any interference with each other *AP*. AP_1 interference with AP_2, AP_3 and AP_4. AP_2 was interference with AP_1, AP_3 and AP_4. AP_3 was interference with AP_1, AP_2 and AP_4. AP_4 was interference with AP_1, AP_2 and AP_3. Every AP is given one channel. Based on DSSS channel allocation, the AP_1 given channel 1, AP_2 channel 6, AP_3 channel 11 and AP_4 another channel. When AP_4 is given channel, there was interference between *APs*. It leads to degradation overall network performance.

Currently, the internet is a very fast growth. Installation of AP more than three is inevitable. One way to solve this problem is minimizing the number of channels used. Several research articles have been published regarding this problem. Dsatur algorithm (Brelaz, 1979) and Welsh Powell algorithm (Welsh, 1967) has been proposed to determine the minimal number of color in the graph coloring theory. Riihijarvi (2006) enhanced Dsatur algorithm by determining the number of channel, based on the high degree of saturation access point. Meanwhile, Rohit (2008) enhance the Welsh Powell algorithm by determining the number of channel, based on high degree of access point. Both of these algorithms, has worked perfectly to determine the number of required channels. However, the number of channels is obtained using this algorithm is still large. Thus, create a new algorithm is motivation in this research to minimize the number of required channels.

1.4 RESEARCH OBJECTIVES

The objectives of the research are as follows:

a) To propose a new algorithm to allocate and reduce the number of required channels.
b) To test the performances of the proposed algorithm.

1.5 RESEARCH SCOPE

In this research, the following scope has been identified:

a) The new algorithm channel allocated for Direct Sequence Spread Spectrum is simulated by using the PHP.
b) The simulation results will be compared with the Welsh Powell algorithm (Rohit, 2008), and Degree of Saturation (DSatur) algorithm (Riihijarvi, 2006).

The Vertex Merge Algorithm (VMA) is proposed based on the graph coloring in order to reduce the required channel. The motivations of this simulation are to show the clarity of the algorithm, and provide the VMA simulator in order to minimize number of required channels.

1.6 ORGANIZATION OF THESIS

This thesis is organized as follows: Chapter 1 presents the background, problem statement, research objectives, research scope and organization of this thesis. Chapter 2 reviews basic concepts, definitions and theorems in graph theory, wireless technology, as well as algorithms used in this thesis. Chapter 3 proposes the Vertex Merge Algorithm (VMA). The experiment and result are elaborated in Chapter 4. Finally, the conclusion and recommendations for the future research are presented in Chapter 5.

1.7 CONCLUSION

This chapter introduces the channel allocation and Direct Sequence Spread Spectrum (DSSS) technology. Channel allocation in DSSS is the main focus attention this research has been elaborated. Together with advantages of channel allocation brings specific problems. It occurs since the bigger number of access points. Determine of channel mechanisms to reducing the number of required channel become the issues. Thus suggest that proper strategies are required to solve the problems, which are the significance of this research.

CHAPTER 2

FUNDAMENTAL CONCEPTS AND THEORY

2.1 INTRODUCTION

This chapter describes the fundamental philosophical foundation which is a part of methodology. The basic concepts of the Direct Sequence Spread Spectrum (DSSS), channel on direct sequence spread spectrum, graph theory, graph coloring, vertex coloring and graph coloring in channel allocation are presented in this chapter. This chapter also reviews some of the foremost graph coloring algorithm namely the Welsh Powell Algorithm (Rohit, 2008) and Degree of Saturation (Dsatur) (Riihijarvi, 2006). These algorithms are then compared to the proposed Vertex Merge Algorithm (VMA) in Chapter 4, in terms of number of channel required. In particular, the reviewed channel allocation algorithm and proposed channel allocation algorithm are based on these concepts.

2.2 DIRECT SEQUENCE SPREAD SPECTRUM (DSSS)

An increasingly popular form of communications is known as the Spread Spectrum (SS). The Spread Spectrum technique was developed initially for military and intelligence requirements. The essential idea is to spread the information signal over a wider bandwidth in order to make jamming and interception more difficult. The first type of spread spectrum developed became known as frequency hopping. A more recent version is the Direct Sequence Spread Spectrum (DSSS). Both techniques are used in various wireless data network products. They also find use in other communications applications, such as cordless telephones (Bensky, 2008).

DSSS is one of the most widely used types of spread spectrum technology, owing its popularity to its ease of implementation and high data rates (Carpenter, 2008). Most of the equipment or the Wireless LAN device on the market uses DSSS technology. DSSS is a method for sending data in which sender and recipient system is both on the set wide frequency is 22 MHz divides the available 83.5 MHz spectrum (in most countries) into 3 wide-band 22 MHz channels. The Institute of Electronics and Electrical Engineers (IEEE) 802.11 standard calls for use of the 2.4 GHz ISM band ranging from 2.400 to 2.497 GHz (Theodore, 2001; Goldsmith, 2005).

IEEE set on the use of DSSS data rates 1 or 2 Mbps in the 2.4 GHz Industrial Scientific and Medical (ISM) band, under the 802.11 standard. Meanwhile, the 802.11b standard specified data rate of 5.5 and 11 Mbps. IEEE 802.11b tools that work on 5.5 or 11 Mbps capable of communicating with the tools that 802.11 work on 1 or 2 Mbps 802.11b standard provides for backward compatibility. In 2003, the IEEE 802.11g standard providing a 54 Mbps data rate using the ISM frequencies. The advantage of 802.11g over 802.11a is that it is backward-compatible with 802.11b (IEEE, 2007).

2.2.1. Channel

Channel is a section on the band frequency (Rackley, 2007). This part is very important so that each frequency does not overlap. DSSS system uses a definition of more conventional channels. Each channel is a band that wide frequency the adjacent 22 MHz. Channel 1, for example, the work frequency of 2.401 GHz to 2.423 GHz (2.412 GHz ± 11 MHz); channel 2 working of 2.406 to 2.429 GHz (2.417 ± 11 MHz). Figure 2.1 illustrates this description.

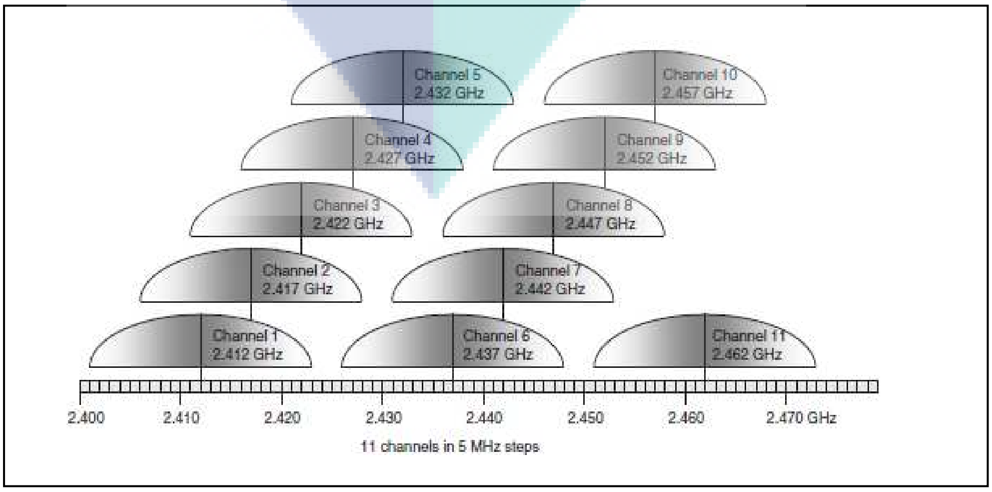


Figure 2.1: DSSS channel allocation (Carpenter, 2008)

An understanding of the legacy 802.11, 802.11b and 802.11g radio is used, its the same important to understand how the IEEE standard divides the 2.4 GHz ISM band into 14 separate channels, as listed in Table 2.1. Although the 2.4 GHz ISM band is dividing into 14 channels, the Federal Communications Commission (FCC) or local regulatory body designate which channels are allowed to be used. Table 2.1 also shows what channels are supported in a sample of a few countries. The regulations can vary greatly between countries (Ahmad, 2003; Carpenter, 2008). Table 2.1 shows a complete list of channels that are used in the United State of America, Europe and Japan. The 802.11b standard defines total number of frequency is 14 channels which 1 through 11 are permitted in the United State of America (USA), 13 in Europe and 14 in Japan. From Table 2.1 can be seen that channels 1 and 2 overlapping with a significant scale. Each frequency listed in this chart is considered a central frequency. From this central frequency, is added and reduced to get an 11 MHz channel with a width of 22 MHz used. Now easily is seeing that channel at the nearness can be overlapping significantly.

Table 2.1: Determining the channel allocation DSSS (Carpenter, 2008)

Channel	Center Frequency (GHz)	USA	Europe	Japan
1	2,412	x	x	x
2	2,417	x	x	x
3	2,422	x	x	x
4	2,427	x	x	x
5	2,432	x	x	x
6	2,437	x	x	x
7	2,442	x	x	x
8	2,447	x	x	x
9	2,452	x	x	x
10	2,457	x	x	x
11	2,462	x	x	x
12	2,467		x	x

13	2,472		x	x
14	2,484			x

Use of DSSS system with the channels overlapping will cause interference between the-system. Because the frequencies node distance of 5 MHz and channels has a wide 22 MHz, the channels may only be placed in the same location if distance of channels 5, separate from one another. For example, channels 1 and 6 does not overlapping. Channel 2 and 7 does not overlapping, etc. There is a maximum of three systems sequence may direct that can be placed on the same location as channels 1, 6 and 11 are channels that are not overlapping theoretically. Three non-overlapping channels that are depicted in Figure 2.2 follows:

Figure 2.2: DSSS non-overlapping channels (Carpenter, 2008)

To use DSSS systems with overlapping channel (e.g., channel 1 and 2) in the same physical space would cause interference between the systems. DSSS systems using overlapping channel should not be co-located because there will almost always be a drastic or complete reduction in throughput. Because the center frequencies are 5 MHz apart, and the center frequencies for non-overlapping channels must be at least 25 MHz apart, channel should be co-located only if the channel numbers are at least five apart.

2.3 DEFINITION OF GRAPH

We define here all the common graphs and graph properties same as standard notation used by Ballobas (1998).

A graph is a mathematical structure consisting of two sets V and E (Diestel, 2010). The elements of V are called *vertices* and the elements of E are called *edges*.

Each edge is identified with a pair of vertices. If the edges of the graph G are identified with ordered pairs of vertices, then G is called a *directed* graph. Otherwise G it is called *an undirected* graph (Koster, 2010). Our discussions in this thesis are concerned with undirected graphs.

We use the symbols $v_1, v_2, v_3, \ldots$ to represent the vertices and the symbols $e_1, e_2, e_3, \ldots$ to represent the edges of a graph. The vertices v_i and v_j associated with and edge e_i are called the *end vertices* of e_i. The edge e_i is then denoted as $e_1 = v_i v_j$. Note that while the elements of E are distinct, more than one edge in E may have the same pair of end vertices. All edges having the same pair of end vertices are called *parallel* or *multiple edges*. Further, the end vertices of an edge need not be distinct. If $e_1 = v_i v_j$, then the edge e_1 is called a *self-loop* at the vertex v_i.

An edge is said to be incident on its end vertices. Two vertices are *adjacent* if they are the end vertices of and edge. If two edges have a common end vertex then these edges are said to be *adjacent* (Bacak, 2004).

For example, in the Figure 2.3 edge e_1 is incident on vertices v_1 and v_2; v_3 and v_4 are two adjacent vertices, while e_1 and e_2 are two adjacent edges.

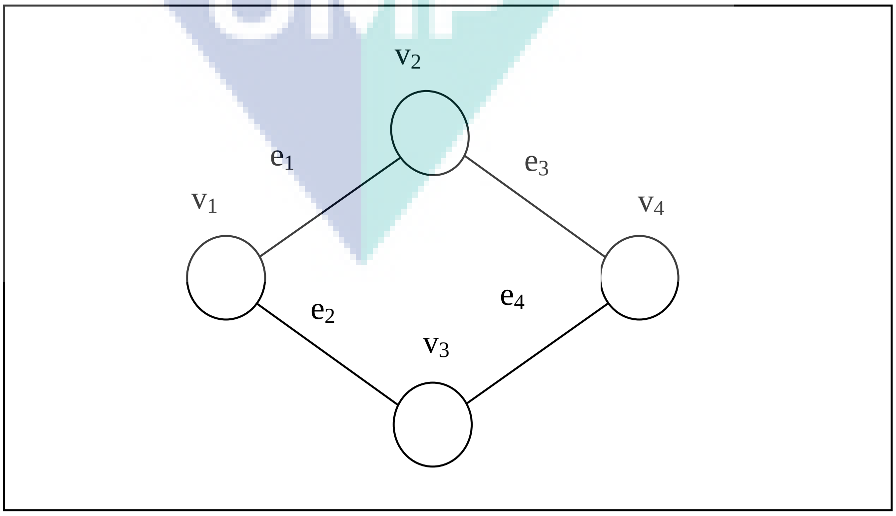


Figure 2.3: A Graph G

The cardinality of the vertex set of a graph G is called the *order* of G and is commonly denoted by $n(G)$, or more simply by n when the graph under considerations is clear. Meanwhile, the cardinality of its edge set is the size of G and is often denoted by $m(G)$ or m. An (n, m) graph has ordered n and size m. A graph with no edges is called an *empty graph*. A graph with no vertex is called a *null graph*. A *subgraph* of a graph G is a graph whose vertex set is a subset of that of G, and whose adjacency relation is a subset of that of G restricted to this subset.

The number of edge incident on a vertex v_i is called the *degree* of the vertex, and it is denoted by $\deg(v_i)$. Sometimes the degree of a vertex is also referred to as its valence. By definition, a self-loop at a vertex v_i contributes 2 to the degree of v_i. A vertex is called *even* or *odd* according to whether its degree is even or odd. A vertex of degree 0 in G is called *isolated vertex* and a vertex of degree 1 is an end-vertex of G. The minimum degree of G is the minimum degree among the vertices of G and is denoted by $\delta(G)$. The maximum degree is defined similarly and is denoted by $\Delta(G)$.

Theorem 2.1 (Euler): The sum of the degrees of a graph is twice the number of edges.

Corollary 2.1: In a graph, there is an even number of vertices having an odd degree.

Proof: Consider separately, the sum of the degrees that are odd and the sum of those that are even. The combined sum is even by the previous theorem, and since the sum of the even degrees is even, the sum of the odd degrees must also be an even. Hence, there must be even number of vertices of odd degree.

To further assumed that G is a simple graph. Graph G with vertex set $V_G = (v_1, v_2, ..., v_n)$ and edge set $E_G = (e_1, e_2, ..., e_m)$ can be depicted in the matrix. One matrix is the matrix n x n adjacent of $A(G) = [a_{ij}]$, where:

$$a_{ij} = \begin{cases} 1, & \text{if } v_i v_j \in E_G \\ 0, & \text{if } v_i v_j \notin E_G \end{cases}$$

Figure 2.4 shows the adjacent matrix of graph *G* from Figure 2.3

$$
\begin{array}{c|cccc}
 & v_1 & v_2 & v_3 & v_4 \\
\hline
v_1 & 0 & 1 & 1 & 0 \\
v_2 & 1 & 0 & 0 & 1 \\
v_3 & 1 & 0 & 0 & 1 \\
v_4 & 0 & 1 & 1 & 0
\end{array}
$$

Figure 2.4: A Graph *G* with the adjacent matrix

2.4 COMMON FAMILIES OF GRAPHS

2.4.1 Simple Graph

A *simple graph*, also called a *strict graph* is an *undirected graph* that has no loops or multiple edges and no more than one edge between any two different vertices. In a simple graph with *n* vertices every vertex has a degree that is less than *n*. Figure 2.5 shows an example of the simple graph.

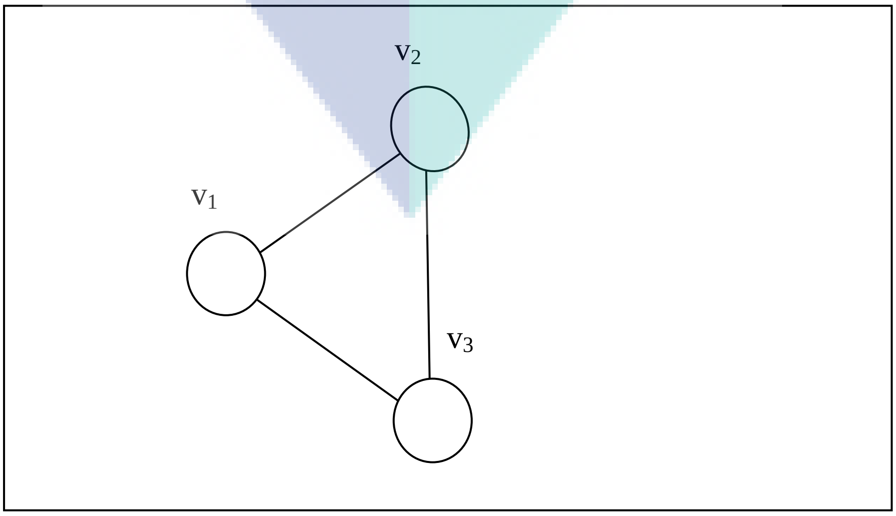


Figure 2.5: A simple graph

2.4.2 Complete Graph

A *complete graph*, sometimes called *universal graph* is a simple graph where every pair of vertices is adjacent. A complete graph on *n* vertices is denoted by K_n. It may be seen that K_n has $n(n-1)/2$ edges. Figure 2.6 shows an example of the complete graph.

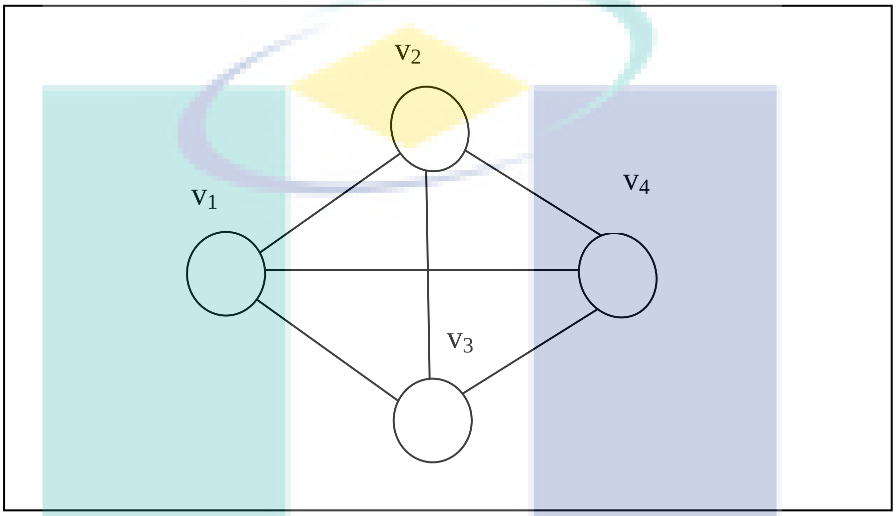


Figure 2.6: A complete graph

2.4.3 Bipartite Graph

A *bipartite graph,* also called a *bigraph* is a graph whose vertices can be divided into two subsets *U* and *W,* such that each edge of *G* has one endpoint in *U* and one endpoint in *W, U* and *W* are independent sets. The pair *U, W* is called a *vertex bipartition* of *G,* and *U* and *W* are called the *bipartition subsets.*

A complete bipartite graph is a simple bipartite graph such that every vertex in one of the bipartition subsets is joined to every vertex in the other bipartition subset. Any complete bipartite graph that has *m* vertices in one of its bipartition subsets and *n* vertices in the other is denoted by $K_{m,n}$. Figure 2.7 shows an example of the complete graph.

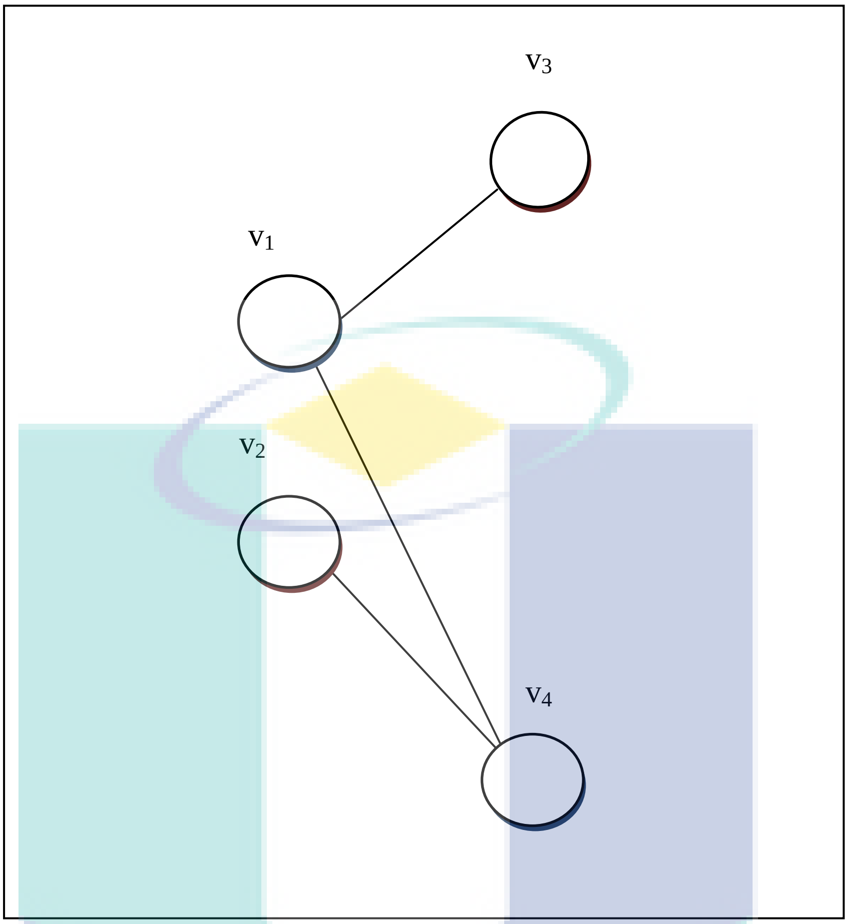


Figure 2.7: A bipartite graph

2.4.4 Complement Graph

The *complement of graph G,* sometimes called the *edge-complement* is a graph *G'* on the same vertices such that two vertices of *G'* are adjacent if and only if they are non-adjacent in *G*. That is to find the complement of graph, fill in all the missing edges to get a complete graph, and remove all the edges that were already there. The graph sum $G + G'$ on n vertices graph G is therefore the complete graph K_n. Figure 2.8(a) shows a simple graph, Figure 2.8(b) shows complement graph. Meanwhile, Figure 2.8(c) shows combination of simple graph and complement graph.

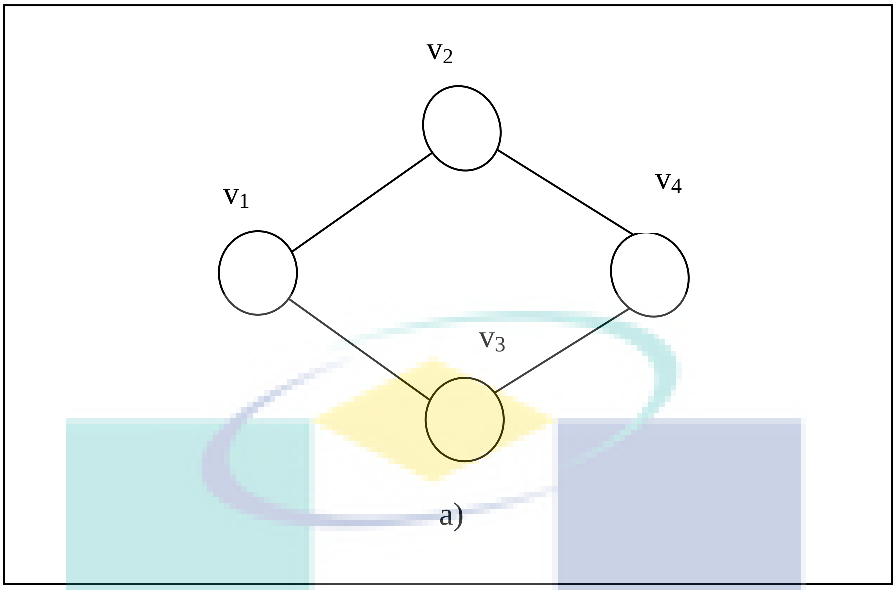


Figure 2.8 (a): A simple graph

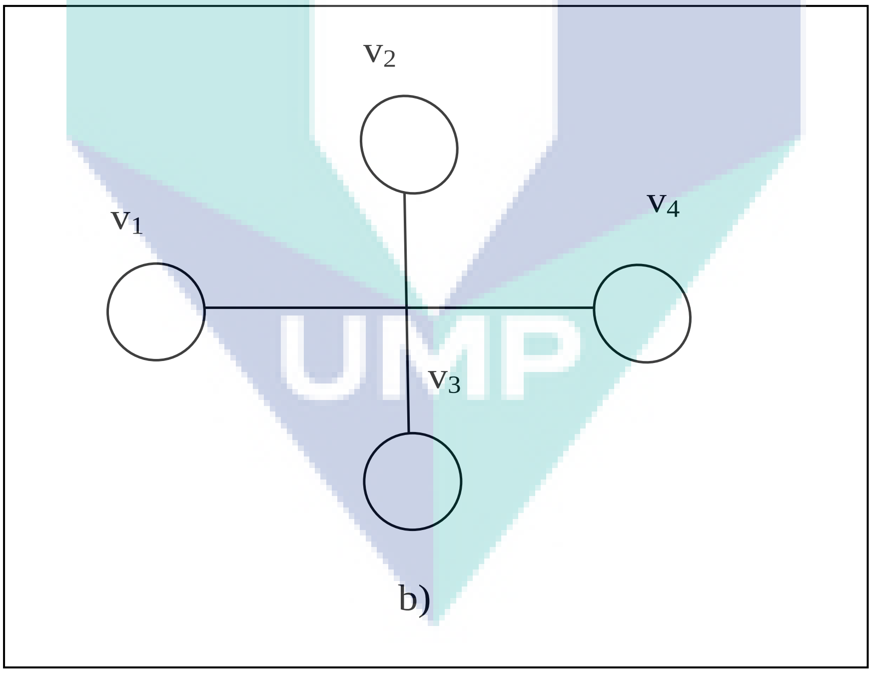


Figure 2.8 (b): A complement graph

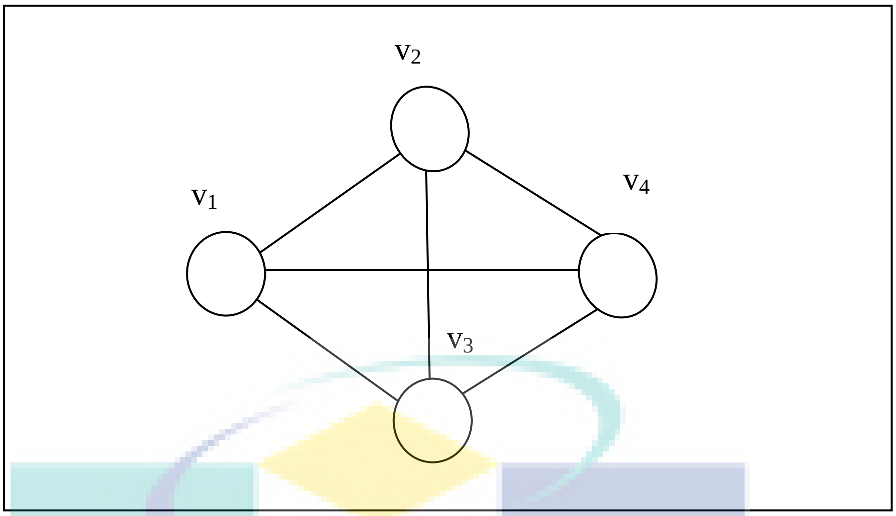


Figure 2.8(c): Combination of simple graph and complement graph

2.4.5 Complex Graph

A *complex graph* is an undirected graph that has number of vertex and edge is large. A complex graph is difficult to be created manually, because of the complicated. A complex graph contains many different subgraph. Figure 2.9 shows a complex graph.

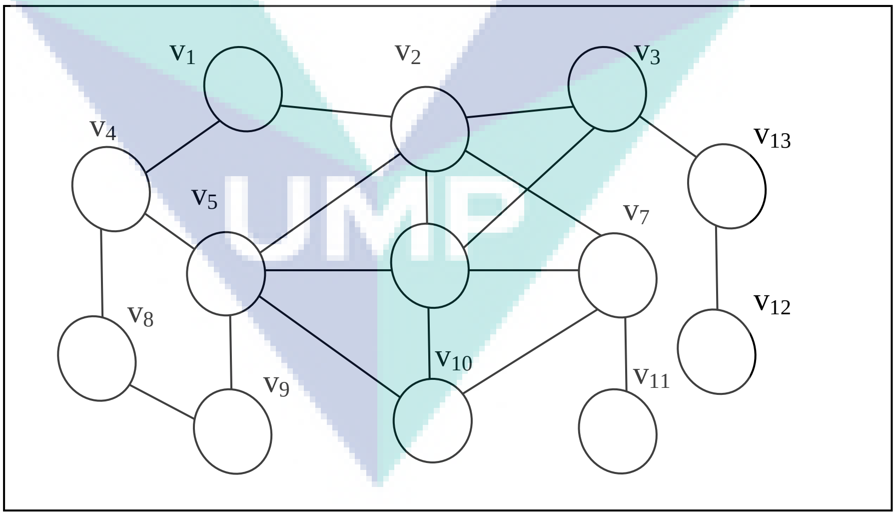


Figure 2.9: A complex graph

2.5 GRAPH OPERATIONS

The union of two graphs is formed by taking the union of the vertices and edges of the graphs. Thus the union of two graphs is always disconnected. The join of the graph G and H is obtained from the graph union $G \cup H$ by adding an edge between each vertex of G and each vertex of H.

The Cartesian product $G = G_1 \times G_2$ has $V(G_1) \times V(G_2)$, and two vertices (u_1, u_2) and (v_1, v_2) of G are adjacent if and only if either $u_1 = u_2$ and $u_1v_2 \in E(G_1)$. A convenient way of drawing $G_1 \times G_2$ is first to place a copy of G_2 at each vertex of G_1 and then to join corresponding vertices of G_2 in those copies of G_2 placed at the adjacent vertices of G_1.

2.6 GRAPH COLORING

A coloring of a graph G is an assignment of a color to each vertex of G (Bolobas, 1998). It has been used to solve problems in school timetabling (Dandashi, 2010), computer register allocation, electronic bandwidth allocation, and many other applications. Graph coloring is an important component of many networking algorithms (Koljanen, 2010). Traditionally, the most prominent use of the graph coloring in networking has been in the context of frequency assignment (Riihijarvi, 2005; 2006).

Vertex coloring of graph $G = (V_G, E_G)$ is a function of V_G to set color, usually represented by a set of real numbers, in such a way so that for every vertex that adjacent be given a different color. Furthermore, if $f(V_G) \leq \{1, 2, \ldots, k\}$, Then f is called the *k-coloring.* A graph G is called *k-colorable* if there is a *coloring s* for an $s \leq k$. If graph G has ordered n, then G is *n-colorable.* Number the minimum round k so a graph G is called *k-colorable* is called chromatic numbers, denoted by $\chi(G)$. If G is a graph with $\chi(G) = k$, then G is called *k-chromatic.*

Definition 2.1: The chromatic number of a graph G is the minimum number of colors needed for a proper vertex coloring of G. If $\chi(G) = k$, G is said to be *k-chromatic*. For example, the chromatic number of the graph in Figure 2.10 is 2.

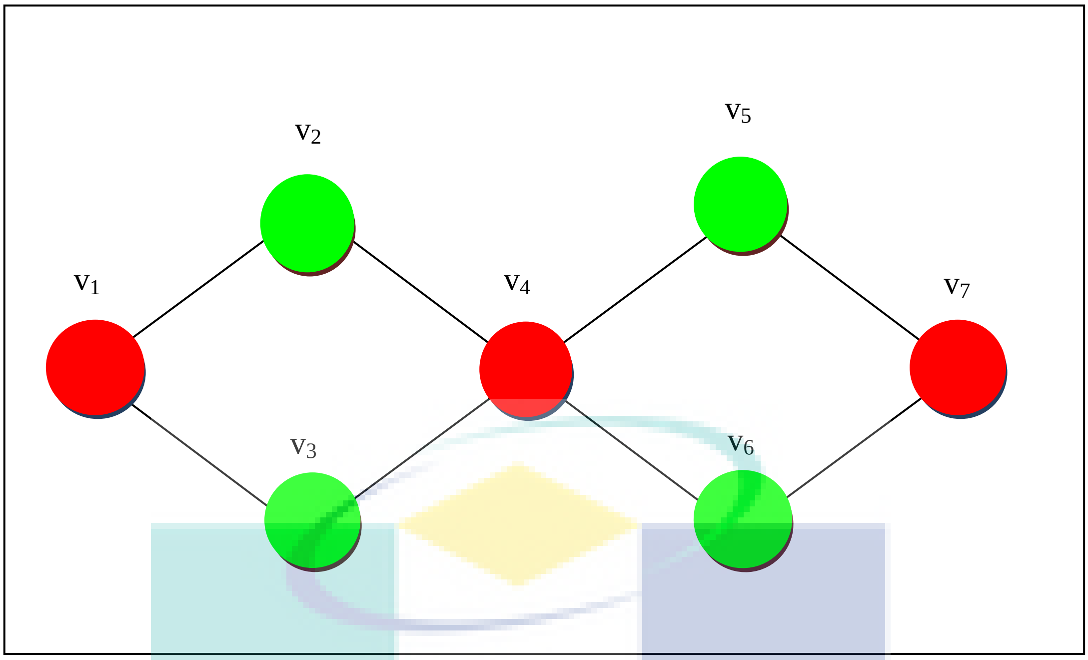


Figure 2.10: A 2 chromatic graph

Definition 2.2: *A k-coloring* of a graph *G* is a vertex coloring of *G* that uses *k* colors.

Definition 2.3: A graph *G* is said to be *k-colorable*, if *G* admits a proper vertex coloring using *k* colors. Thus, $\chi(G) = k$ if graph *G* is *k-colorable* but not *(k − 1)-colorable*. In considering the chromatic number of a graph only the adjacency of vertices is taken into account. A graph with a self-loop is regarded as uncolorable, since the endpoint of the self-loop is adjacent to itself. Moreover, a multiple adjacency has no more effect on the colors of its endpoints than a single adjacency. As a consequence, we may restrict ourselves to simple graphs when dealing with chromatic numbers. Clearly that $\chi(G) = 1$, if and only if *G* has no edges and $\chi(G) = 2$, if and only if *G* is bipartite.

2.6.1 Chromatic Numbers for Common Graph Families

It is straightforward to establish the chromatic number of graphs in some of the most common graph families, by using the basic principles given above.

Proposition 2.1: For complete graphs, $\chi(K_n) = n$

Proof: Since *n* complete graphs have *n* mutually adjacent vertices using fewer than *n* colors result in a pair of mutually adjacent vertices being assigned the same color.

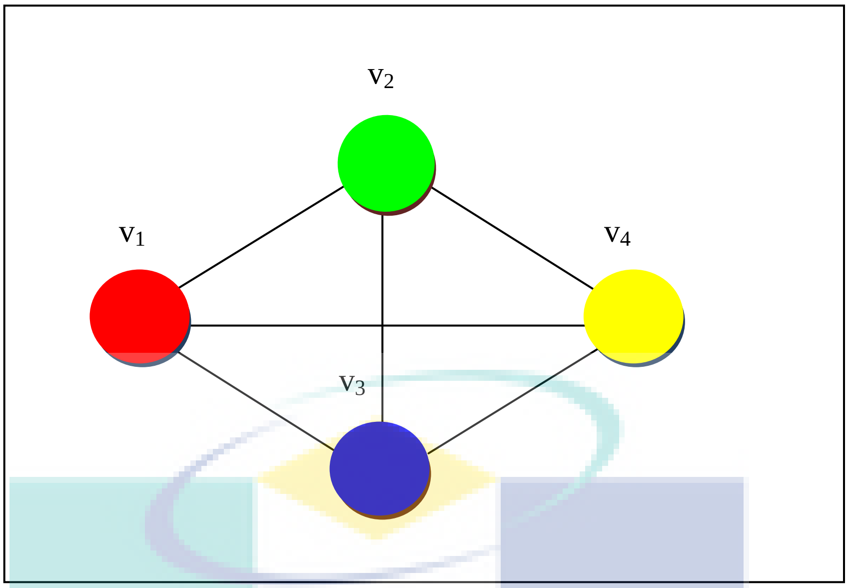


Figure 2.11: A complete graph with color

Proposition 2.2: For bipartite graphs, $\chi(G) = 2$.

Proof: A 2-coloring is obtained by assigning one color to every vertex in one of the bipartition parts and another color to every vertex in the other part, as depicted in Figure 2.11.

Corollary 2.1: Even cycles have the chromatic number $\chi(C2n) = 2$ since they are bipartite.

Proposition 2.3: For odd-cycle graphs, $\chi(C2n+1) = 3$.

Proof: Let v_1, v_2, ..., v_2n+1 be the vertices of the cycle graph $C2n+1$. If two colors were to suffice, then they would have to be alternate around the cycle. Thus, the odd-subscripted vertices would have to be one color and the even-subscripted ones the other. However, vertex $v2n+1$ are adjacent to v_1, which means that the odd cycle graph $C2n+1$ is not 2-colorable, as depicted in Figure 2.13.

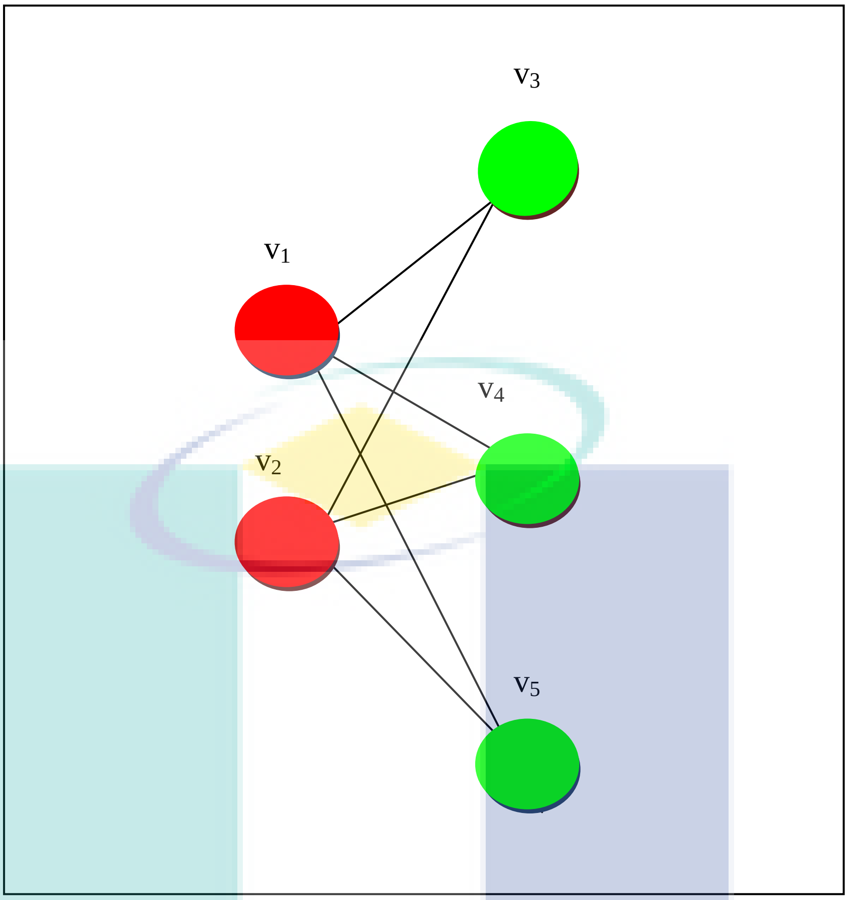


Figure 2.12: A complete bipartite graph

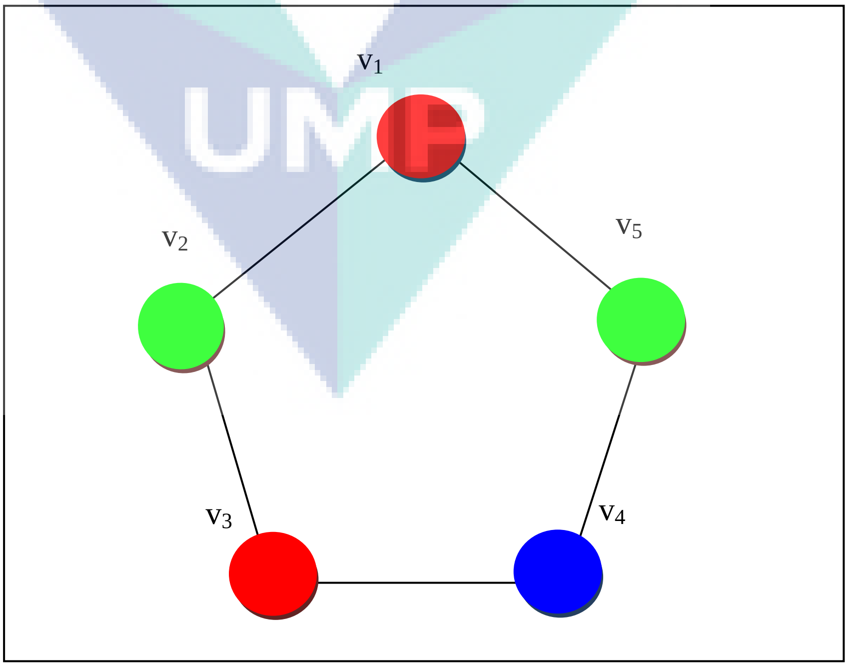


Figure 2.13: An odd cycle

2.7 WELSH POWELL ALGORITHM

Welsh Powell algorithm is an algorithm that tries to solve the graph coloring problem (Rohit, 2008). Graph coloring is a way of coloring the vertices of a graph such that no two adjacent vertices shared the same color (Diestel, 2010). Welsh Powell algorithm can be used for coloring of a graph efficiently. This algorithm not always provides the minimum number of colors required to coloring a graph, but this algorithm practical enough to use in coloring simple graph. Welsh Powell algorithm is only suitable for a graph with a small number of degrees (Rohit, 2008). As for the steps Welsh Powell algorithm is as follows:

1. Arrange the vertex by decreasing order of degrees.
2. Choose the first uncolored vertex from the set.
3. Color the chosen vertex with the least possible color.
4. Color the next vertex that is not adjacent to the first vertex by the same color with first vertex.
5. Then proceed with the second color, and onwards, until all vertices has been given a color.
6. If the entire vertex is colored, stop. Otherwise, return to Step 2.

2.7.1 An Example of Welsh Powell Algorithm

We demonstrate the steps of the algorithm with a small example. The input graph is shows below in Figure 2.14 which has eight vertices labeled $V = \{1, 2, 3, 4, 5, 6, 7, 8\}$. The algorithm required three colors of the vertices using the set of colors {1, 2, 3} represented by red, green and blue respectively, as depicted in Figure 2.14.

Graph coloring problem in Figure 2.14, will be solved by Welsh Powell algorithm with the steps as follows:

1. Arrange the vertex by decreasing order of degrees. Largest degree vertex is v_5, that is 5 (having five segments), then vertex v_3 with degree 4, v_2, v_4, v_6 of each with degree 3 and v_1, v_7, v_8 respectively with degree 2. Thus the sequence is $\{v_5, v_3, v_2, v_4, v_6, v_1, v_7, v_8\}$.

2. Choose the first uncolored vertex from the set that is v_5.

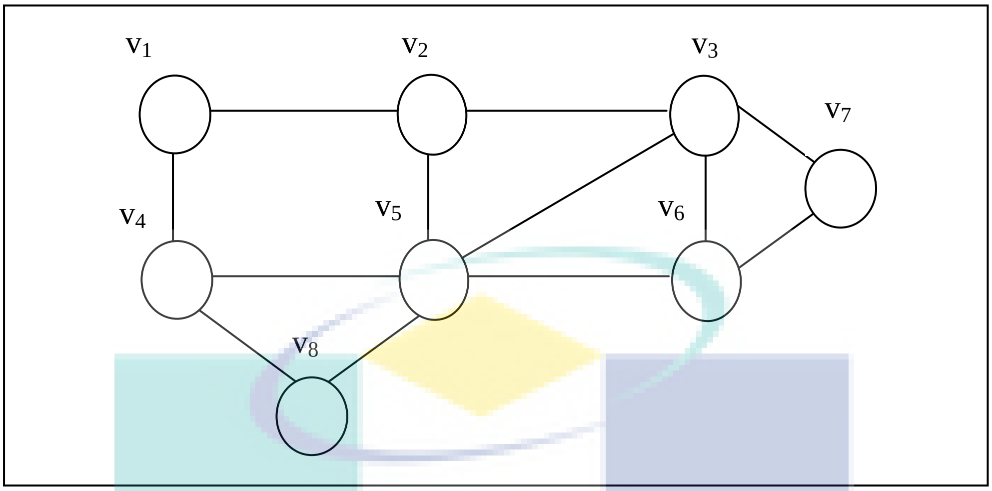


Figure 2.14: An example of graph which has eight vertices

3. Color the chosen vertex with the least possible color, used red.

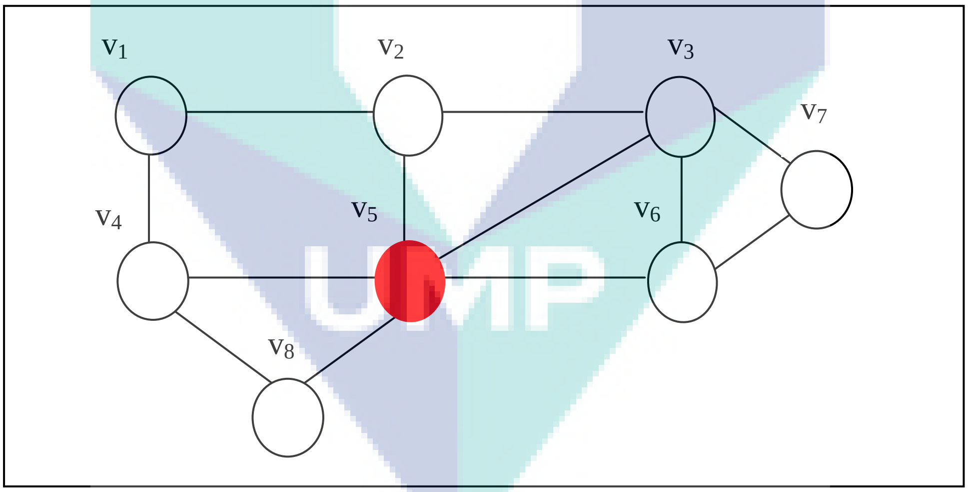


Figure 2.15: Result of an example graph with first color

4. Color the next vertex that is not adjacent to the first vertex by the same color with first vertex, thus v_1 and v_7 will be color with red.

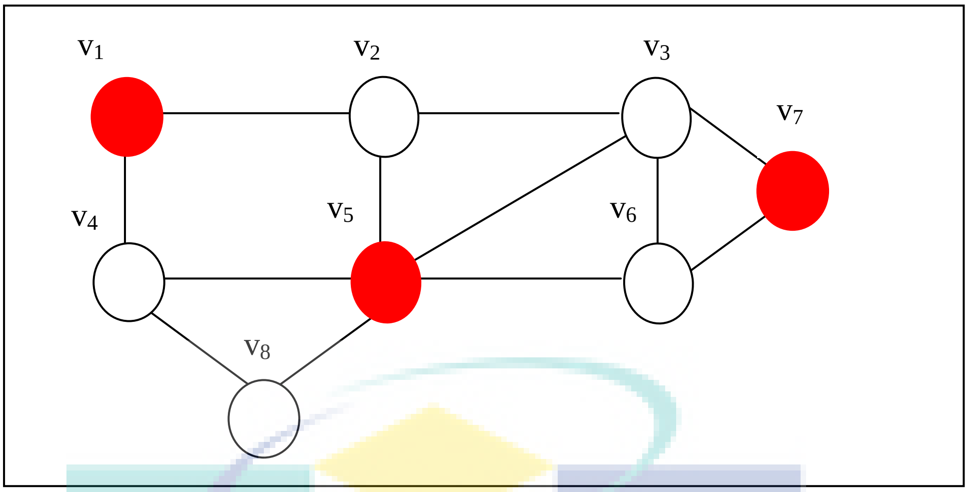


Figure 2.16: Result of an example graph with first color on not adjacent vertex

5. Now sequence uncolored vertex by decreasing order of degrees is $\{v_3, v_2, v_4, v_6, v_8\}$.
6. Choose the first uncolored vertex from the set that is v_3.
7. Color the chosen vertex with the least possible color, used green.

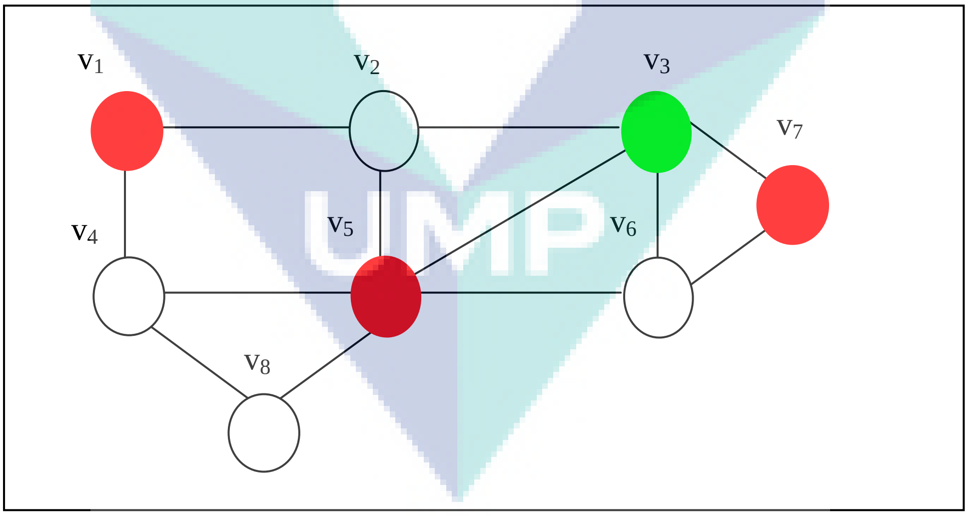


Figure 2.17: Result of an example graph with first color and second color

8. Color the next vertex that is not adjacent to the first vertex by the same color with first vertex, thus v_4 and v_8 will be color with green.

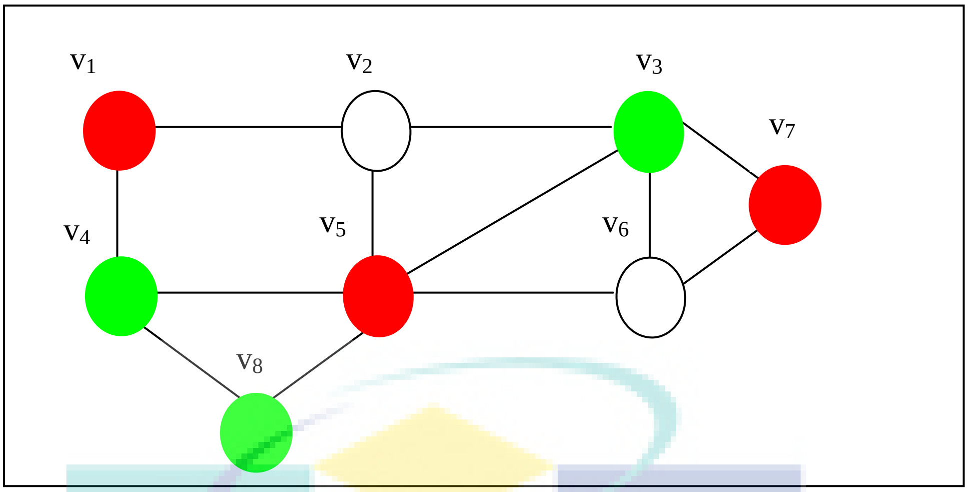


Figure 2.18: Result of an example graph with first color and second color on not adjacent

9. Now sequence uncolored vertex by decreasing order of degrees is $\{v_2, v_6\}$.
10. Chose the first uncolored vertex from the set that is v_2.

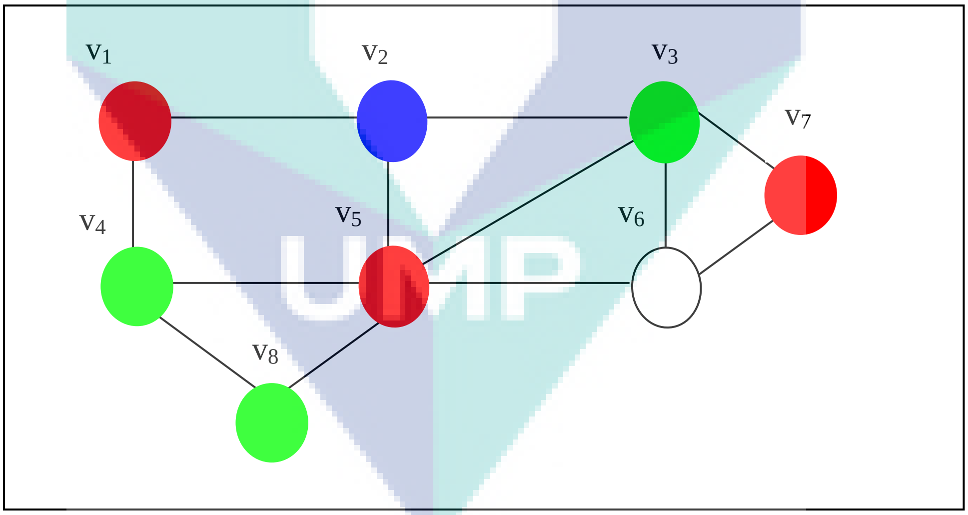


Figure 2.19: Result a simple graph with first color, second color and third color

11. Color the next vertex that is not adjacent to the first vertex by the same color with first vertex, thus v_6 will be color with blue. The graph coloring results are:

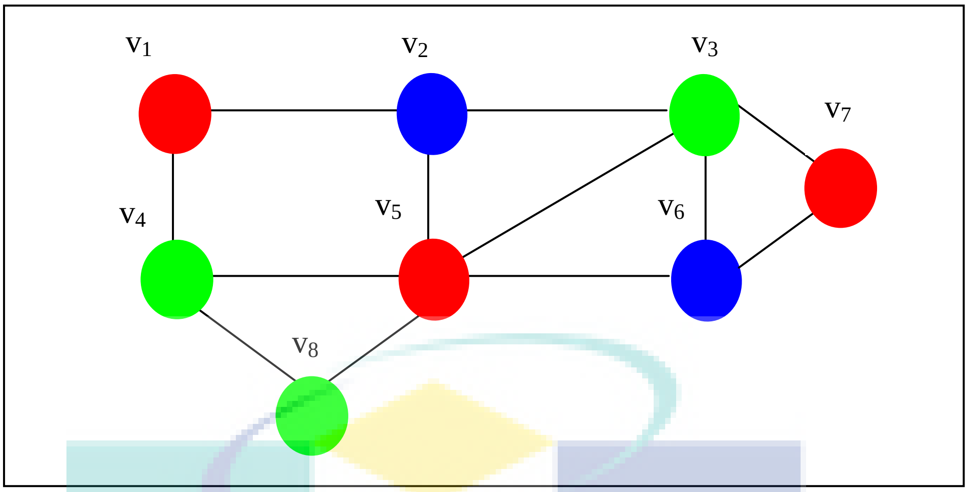


Figure 2.20: Result of an example graph using Welsh Powell algorithm

From the Figure 2.20 can be seen that the vertex v_4 and v_8 adjacent but have the same color. This is contrary to the definition of the graph coloring where every vertex adjacent will be given a different color.

2.8 DEGREE OF SATURATION ALGORITHM

In 1979 Brelaz proposed Degree of Saturation (Dsatur). Dsatur algorithm is an algorithm coloring sorted by building sequence of vertices dynamically. Dsatur algorithm has been an implementation to channel allocation by Riihijarvi, et al (2006). Dsatur algorithm is practical enough to solve problems of graph coloring. However, this algorithm impractical enough for graphs with a large number of degrees (Brelaz, 1979; Riihijarvi, 2005; 2006). As for the steps degree of saturation algorithm are as follows:

1. Arrange the vertex by decreasing order of degrees.
2. Color a vertex of maximal degree with color 1.
3. Choose a vertex with the maximal saturation degree. If there is equality, choose any vertex of maximal degree in the uncolored subgraph.
4. Color the chosen vertex with the least possible color.
5. If the entire vertex is colored, stop. Otherwise, return to Step 3.

2.8.1 An Example of Dsatur Algorithm

We demonstrate the steps of the algorithm with a small example. The input graph is shows below in Figure 2.21 which has six vertices labeled *V* = {1, 2, 3, 4, 5, 6}. The algorithm required four colors of the vertices using the set of colors {1, 2, 3, 4} represented by red, green, blue and yellow respectively.

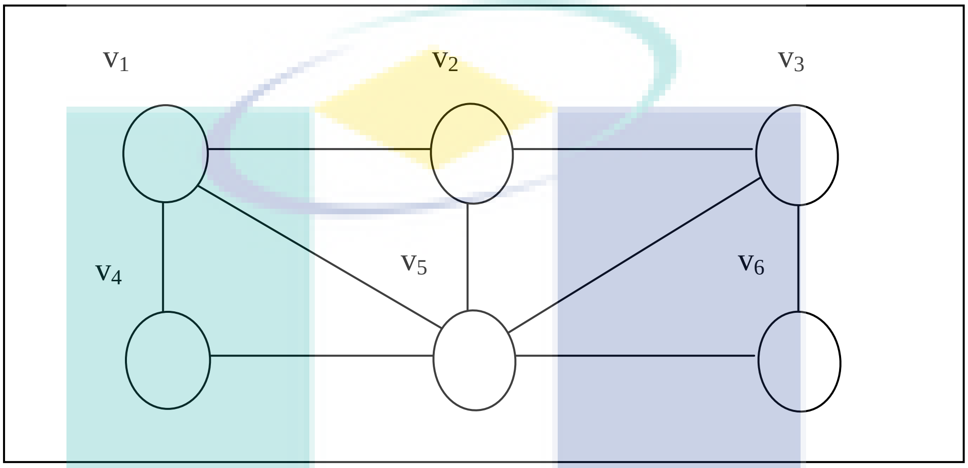


Figure 2.21: An example of graph with six vertices

Graph coloring problem in Figure 2.21, will be solved by Dsatur algorithm with the steps as follows:

1. Arrange the vertex by decreasing order of degrees. The vertex with the largest degree is v_5, that is 5 (having five segments), then vertex v_1, v_2, v_3 with the degree 3 and v_4, v_6 respectively with degree 2. Thus the sequence is {v_5, v_1, v_2, v_3, v_4, v_6}.
2. Color a vertex of maximal degree with color 1, used red, as depicted in Figure 2.21.
3. Choose a vertex with the maximal saturation degree. If there is equality, choose any vertex of maximal degree in the uncolored subgraph. Table 2.2 shows the degree saturation from Figure 2.22.

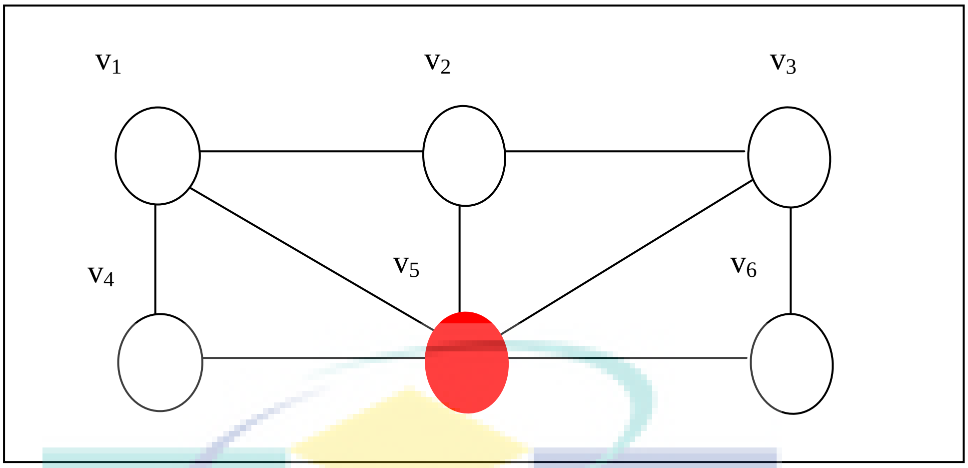


Figure 2.22: Result of an example graph with first color

Table 2.2: Saturation degree with first color

Vertex	**v_1**	**v_2**	**v_3**	**v_4**	**v_5**	**v_6**
v_1	0	1	0	1	1	0
v_2	1	0	1	0	1	0
v_3	0	1	0	0	1	1
v_4	1	0	0	0	1	0
v_5	1	1	1	1	0	1
v_6	0	0	1	0	1	0
Degree	3	3	3	2	5	2
Saturation Degree	2	2	2	1	-	1
Uncolor Degree	2	2	2	1	-	1

From Table 2.2 the vertex with the maximal saturation degree is v_1 with saturation degree 2.

4. Color the chosen vertex with the least possible color, used green.

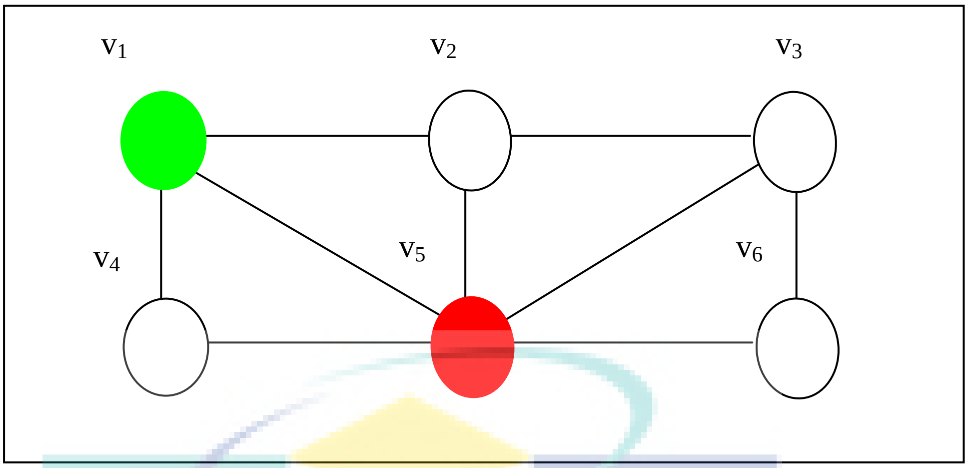


Figure 2.23: Result of an example graph with second color

5. Choose a vertex with the maximal saturation degree. If there is equality, choose any vertex of maximal degree in the uncolored subgraph. Table 2.3 shows the degree saturation from Figure 2.23.

Table 2.3: Saturation degree with second color

Vertex	v_1	v_2	v_3	v_4	v_5	v_6
Degree	3	3	3	2	5	2
Saturation Degree	-	1	2	0	-	1
Uncolor Degree	-	1	2	0	-	1

From Table 2.3 the vertex with the maximal saturation degree is v_3 with saturation degree 2.

6. Color the chosen vertex with the least possible color, used blue.

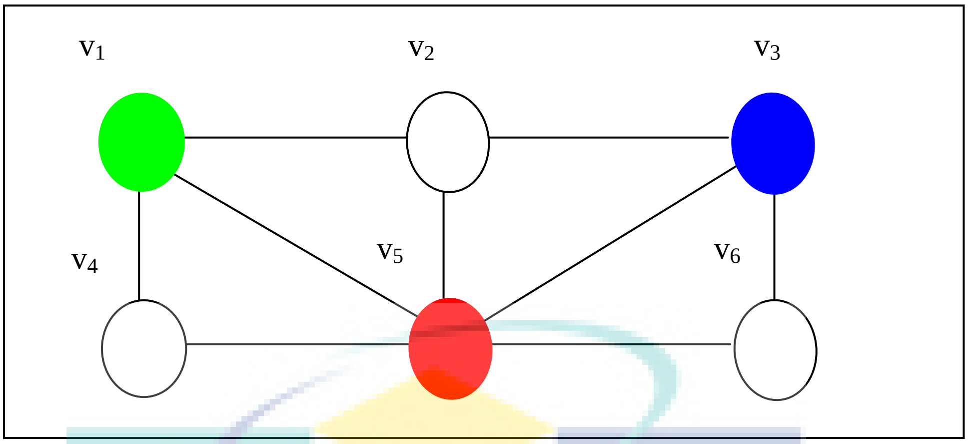


Figure 2.24: Result of an example graph with third color

7. Choose a vertex with the maximal saturation degree. If there is equality, choose any vertex of maximal degree in the uncolored subgraph. Table 2.4 shows the degree saturation from Figure 2.24.

Table 2.4: Saturation degree with third color

Vertex	v_1	v_2	v_3	v_4	v_5	v_6
Degree	3	3	3	2	5	2
Saturation Degree	-	0	-	0	-	0
Uncolor Degree	-	0	-	0	-	0

From Table 2.4 seen on the row saturation degree and uncolor degree all columns valuable 0 (v_2, v_4, v_6). In this case, coloring process for v_2, v_4, v_6 based on the index vertex. Thus choose the vertex with minimal index that is v_2.

8. Color the chosen vertex with the least possible color, used yellow.

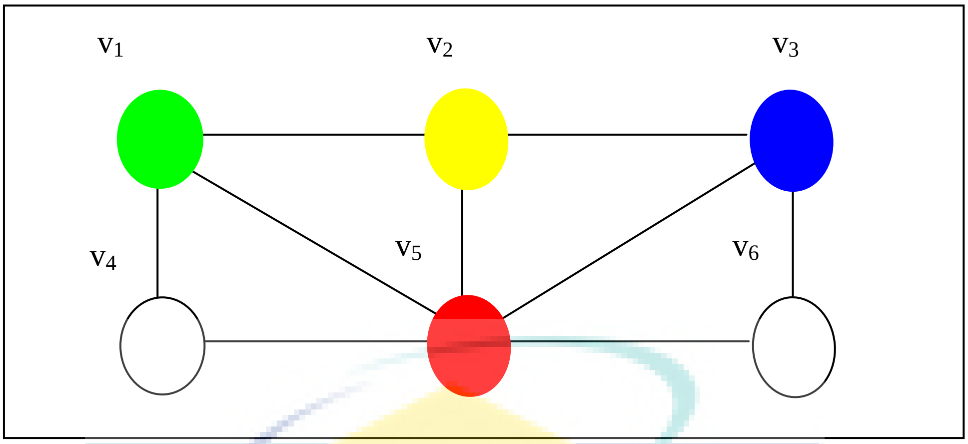


Figure 2.25: Result of an example graph with fourth color

9. Choose a vertex with the minimal index. If there is equality, choose any vertex of maximal degree in the uncolored subgraph. Table 2.5 shows the degree saturation from Figure 2.25.

Table 2.5: Saturation degree with fourth color

Vertex	**v_1**	**v_2**	**v_3**	**v_4**	**v_5**	**v_6**
Degree	3	3	3	2	5	2
Saturation Degree	-	-	-	0	-	0
Uncolor Degree	-	-	-	0	-	0

From Table 2.5 the vertex with the minimal index is v_4.

10. Color the chosen vertex with the least possible color, used blue.

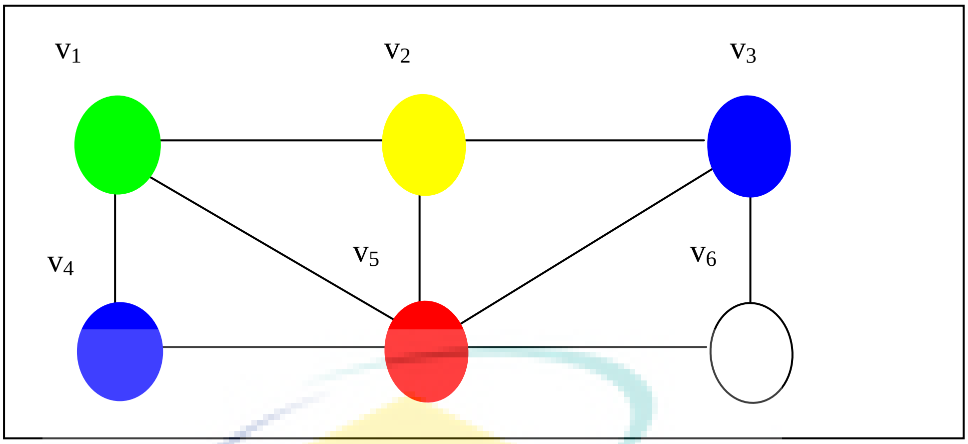


Figure 2.26: Result of an example graph with two blue colors

11. Choose a vertex with the minimal index. If there is equality, choose any vertex of maximal degree in the uncolored subgraph. Table 2.6 shows the degree saturation from Figure 2.26.

Table 2.6: Saturation degree with last step

Vertex	v_1	v_2	v_3	v_4	v_5	v_6
Degree	3	3	3	2	5	2
Saturation Degree	-	-	-	-	-	0
Uncolor Degree	-	-	-	-	-	0

From Table 2.6 the vertex with the minimal index is v_6.

12. Color the chosen vertex with the least possible color, used green.

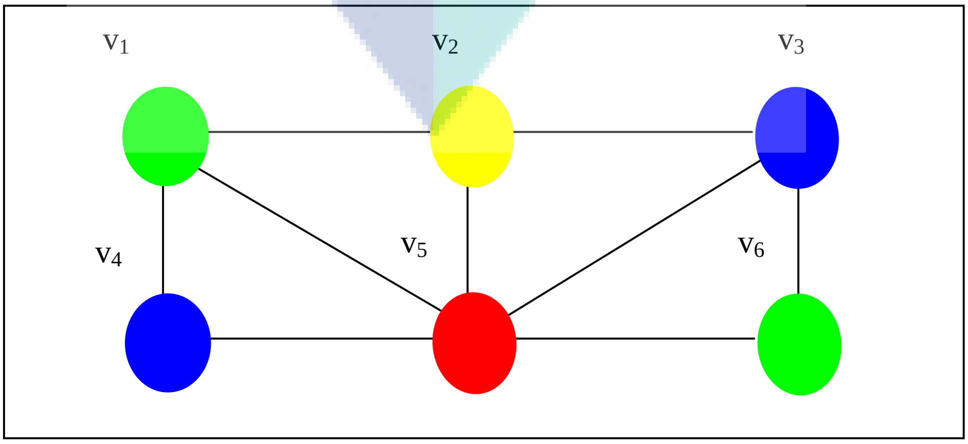


Figure 2.27: Result of an example graph using Dsatur algorithm

From Figure 2.27, it can be seen that example of graph in Figure 2.17 can be solved using Dsatur algorithm which requires four colors.

2.10 RESEARCH METHODOLOGY

The methodology of this research as illustrated in flowchart of Figure 2.28. The literature review is fundamental of this research. Design a new algorithm of channel allocation for DSSS are elaborated in Chapter 3. Simulation the algorithm, comparison with existing algorithm, analysis the performance and produce of product are elaborated in Chapter 4.

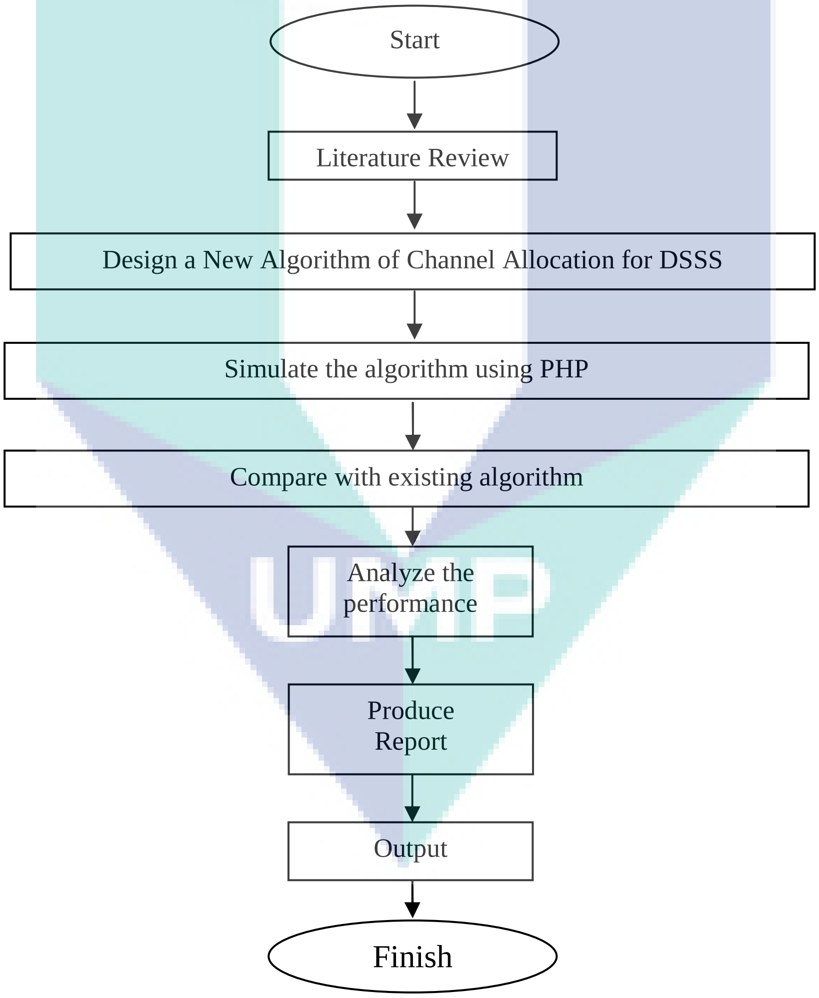


Figure 2.28: Flowchart of methodology research

Table 2.7 shows the comparison between Welsh Powell algorithm and Dsatur algorithm. Welsh Powell algorithm and Dsatur algorithm can be used for coloring of a graph efficiently. Both algorithms are practical enough to use in coloring simple graph. Welsh Powell algorithm is only suitable for a graph with a small number of degrees. Dsatur algorithm not always provides the minimum number of required channel in large number of degrees.

Table 2.7: Comparison between Welsh Powell and Dsatur algorithm

No.	Compare	Algorithm	
		Welsh Powell	**Dsatur**
1	Advantages	- Enough to use in coloring simple graph	- Enough to use in coloring simple graph
2	Disadvantages	- Only suitable for a graph with a small number of degrees - Not always provide the minimum number of colors required	- Impractical enough for graphs with a large number of degrees - Not always provide the minimum number of colors required

2.10 CONCLUSION

In this chapter, the fundamental concepts such as the Direct Sequence Spread Spectrum (DSSS), channel on DSSS, graph theory, graph coloring, vertex coloring and graph coloring in channel allocation have been elaborated.

DSSS is one of the most widely used types of spread spectrum technology. The DSSS system based of 802.11b and 802.11g IEEE standard. In DSSS wide the frequency each channel is 22 MHz. DSSS defines total number of frequency is 14 channels which 1 through 11 are permitted in the USA, 13 in Europa and 14 in Japan.

A graph is a mathematical structure consisting of two sets *V* and *E*. The elements of V are called *vertices* and the elements of E are called *edges*. A simple graph is an undirected graph that has no loops or multiple edges and no more than one edge between any two different vertices. A complex graph is an undirected graph that has number of vertices and edge is large. A complex graph is difficult to be created manually, because of the complicated.

The union of two graphs is formed by taking the union of the vertices and edges of the graphs. Thus the union of two graphs is always disconnected. The join of the graph G and H is obtained from the graph union *G U H* by adding an edge between each vertex of *G* and each vertex of *H*.

A coloring of a graph *G* is an assignment of a color to each vertex of *G* that for every vertex that adjacent be given a different color. The chromatic number of a graph *G* is the minimum number of colors needed for a proper vertex coloring of G. If $\chi(G) = k$, *G* is said to be *k-chromatic*.

This chapter also reviews some of the foremost graph coloring namely the Welsh Powell algorithm and Degree of Saturation algorithm (Dsatur). Welsh Powell algorithm is an algorithm that tries to solve the graph coloring problem. Welsh Powell algorithm can be used for coloring of a graph efficiently. This algorithm not always provides the minimum number of colors required to coloring a graph, but this algorithm practical enough to use in coloring simple graph.

Dsatur algorithm is an algorithm coloring sorted by building sequence of points dynamically. Dsatur algorithm has been an implementation to channel allocation by Riihijarvi, et al (2006). Dsatur algorithm it is practical enough to solve problems of graph coloring. However, this algorithm impractical enough for graphs with a large number of degrees.

CHAPTER 3

VERTEX MERGE ALGORITHM

3.1 INTRODUCTION

In this chapter, a new channel allocation algorithm is proposed in order to reduce the number of required channel based on graph theory. It focuses on the channel allocation for direct sequence spread spectrum. Specifically, this algorithm is called Vertex Merge Algorithm (VMA), which will be elaborated below.

3.2 VERTEX MERGE ALGORITHM

The Vertex Merge Algorithm (VMA) implicitly uses merge vertex and merge edge. A merge vertex is a set of vertex, which will be assigned the same color as any pair of vertex in that set is never connected. Merge vertex can be generalized for normal vertex by considering a one element color set, i.e., one color.

Definition 3.1: Given two colored vertex (v_1, c_1) and (v_2, c_2) we create a merge vertex $\{v_1, v_2\}$ if $v_1 \neq v_2$ and $c_1 = c_2$. Generally, $M = \bigcup\{v_i\} i\varepsilon I$, where $I \subseteq [1, n]$ is an index set having at least two elements, can form a merge vertex, when $\{v_i) i\varepsilon I x \{v_i) i\varepsilon I \bigcap E = 0$.

Here, a merge edge can connect only two merge vertices if and only if they are connected by at least two normal edges. We can generalize merge edge to normal edges as well, if we allow a merge edge to consist of one edge only.

Definition 3.2: Let $I_1, I_2 \subseteq \{1, n\}$ distinct a non empty index sets, i.e. $I_1 \bigcap I_2 = 0$, furthermore $M_1 = \{v_i) i\varepsilon I_1$ and $M_2 = \{v_j) j\varepsilon I_2$ form merge vertex, which is there is no intra edge between the vertex in a set: $M_1 x M_1 \bigcap E = 0$ and $M_2 x M_2 \bigcap E = 0$. H_1 and H_2 are connected by a merge edge (M_1, M_2) if there are inter edges between the vertices in the different sets, i.e. $M_1 x M_2 \cap E > 1$.

When using graphs in this thesis, we shall represent to merge edge with two parallel lines to make them distinct from regular edges for which a single line is used. Similarly, for merge vertex we will add another slightly larger circle around the vertex, and annotate the vertex with a set of variables. Although, the coloring of vertex is implicit in our model, as it starts with assigning unique colors to every vertex, then removing colors as vertex are merged into merge vertex, we will color vertex to make the process easier to follow. Colored vertex is shows by coloring half the vertex. If the vertex is colored with the same color, they have the same half colored.

The VMA concentrates on the operations between merge vertex and normal vertex. We try to merge the normal vertex with another vertex, and when the latter is a merge vertex. A reduction in adjacency checks is possible. These checks can be performed along merge edge instead of normal edges, whereby we can introduce significant savings. This is because the initial set of normal edges is folded into merge edge. The coloring data is stored in a Vertex Merge Table (VMT). Every cell (i,j) in this table has non negative integer values. The columns refer to the vertex, and the rows refer to the colors. A value in cell (i,j) is greater than zero if and only if the vertex j cannot be assigned a color *i* because of the edges in the original graph $G(V,E)$.

The initial VMT is the adjacency matrix of the graph; hence a unique color is assigned to each of the vertices. If the graph is not a complete graph, then it might be possible to reduce the number of necessary colors. This corresponds to the reduction of rows in the VMT. To reduce the rows, we introduce a Vertex Merge Operation (VMO), which attempts to merge two rows. When this is possible, the number of colors is decreased by one. When it is not, the number of colors remains the same. It is achievable only when two vertices are not connected by a normal edge or a merge edge.

The Vertex Merge Operation merges an initial row l_i into an arbitrary (initial or merged) row l_j if and only if $(j,i) = 0$ (i.e., none of the vertices in the merge vertex

$\{v_{j_1},...,v_{j_m}\}$ are connected to the vertex v_i) in the VMT. If rows l_i and l_j can be merged then the result is the union of these rows, which in the context of the graph $G(V, E)$ amounts to either creating a merge vertex $\{v_i, v_{j_1},...,v_{j_m}\}$ or merging two merge vertex $\{v_{i_1},...,v_{i_t}, v_{j_1},...,v_{j_m}\}$ with $2 \le m,t \le n$ and $m+t \le n$.

Definition 3.3: The Vertex Merge Operation Let S is the set of initial rows of the VMT and R is the set of all possible $|V|$ size integers valued rows (vectors). Then a vertex merge operation is defined as,

Merge (l_i, l_j) : $RxS \to R$

l'_j := $l_j + l_i, l'_j, l_j \varepsilon R, l_i \varepsilon S$

or by components,

$l'_j(1)$:= $l_j(1) + l_i(1), 1, 2, ..., |V|$

A merge can be associated with an assignment of a color to a vertex, because two vertices are merged if they have the same color. Hence, we need as many merge operation as the number of the vertex in a valid coloring of the graph, apart from the vertex which is colored initially and then never merged, i.e., a color is used only for one vertex. If k numbers of rows are left in the VMT, i.e., the number of colors used, then the number of Vertex Merge Operations is $|V| - k$, where $k \varepsilon \{\chi, ..., |V|\}$.

The following is a step by step VMA.

1. Arrange the vertex by decreasing order of degrees.
2. Choose the first uncolored vertex from the set.
3. Color the chosen vertex with the least possible color.
4. Merge the vertex with the first non-adjacent vertex.
5. Color the chosen vertex with the same color. If there is the no more non-adjacent vertex, return to step 2.
6. If the entire vertex is colored, stop. Otherwise, return to Step 2.

3.3 VMA IN CHANNEL ALLOCATION

In this section, DSSS channel allocation problem is formulated in terms of the graph coloring problem theory. The first is recalling the statement coloring problem based on graph theory (Diestel, 2010) is interesting in the context of channel allocation.

Suppose it is given a simple graph $G = (V, E)$, that is, a graph consisting of a set of vertex V, and set of edges E connecting the vertex. Loops and multiple edges between vertices are not allowed. Then a vertex coloring of G is a map: $V(G) \rightarrow F$, where F is a set of colors, usually some small subset of positive integers. It shall call a coloring admissible, if $C(V_i) = C(V_j)$ for all adjacent V_i and V_j (that is, for those vertices connected by an edge). It is called an *admissible coloring minimizing* $|C(V)|$ an optimal coloring. The number of colors used by the optimal coloring is called the *chromatic number* of the graph.

3.3.1 Interference Graphs

It shall now formulate the channel allocation problem in terms of the terminology introduced in the previous section. Given a collection $\{v_i\}$ of access points, it shall form an interference graph $G = (V, E)$ as follows. The vertex set V is simply identified with the set $\{v_i\}$. The set of edges E is constructed as the union of those pairs $\{v_k, v_l\}$ of vertex, that correspond to access points v_k and v_l that would interfere with each other's radio traffic should they be assigned to use the same channel. Finally, let F, the set of colors, to be the collection of channels available to the access points. Now the channel allocation problem is simply finding of an admissible coloring of G with the color set F.

Naturally, the size of the color set is greatly technology and legislation dependent. In most European countries, $F = \{1, 2\ldots 13\}$ for DSSS technologies, of which the subset $F' = \{1, 6, 11\}$ corresponds to the non-overlapping channels. Table 3.1 shows estimated of channel in DSSS.

Table 3.1: Estimation of channel

Number of Color	**Allocation of Channel**												
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
1	x												
2	x					x							
3	x					x					x		
4	x				x				x				x
5	x			x			x			x			x
6	x			x			x		x		x		x
7	x		x		x		x		x		x		x
8	x		x		x		x		x		x	x	x
9	x		x		x		x		x	x	x	x	x
10	x		x		x		x	x	x	x	x	x	x
11	x		x		x	x	x	x	x	x	x	x	x
12	x		x	x	x	x	x	x	x	x	x	x	x
13	x	x	x	x	x	x	x	x	x	x	x	x	x

Table 3.1 shows that,

1. If only one color is needed, then the channel provided for the access point is to channel 1.
2. If there are two colors needed, then the channel provided for the access point is to channel 1 and channel 6.
3. If there are three colors needed, then the channel provided for the access point is to channel 1, channel 6 and channel 11.
4. If there are four colors needed, then the channel provided for the access point is to channel 1, channel 5, channel 9 and channel 13.
5. If there are five colors needed, then the channel provided for the access point is to channel 1, channel 4, channel 7, channel 10 and channel 13.
6. If there are six colors needed, then the channel provided for the access point is to channel 1, channel 4, channel 7, channel 9, channel 11 and channel 13.

7. If there are seven colors needed, then the channel provided for the access point is to channel 1, channel 3, channel 5, channel 7, channel 9, channel 11 and channel 13.
8. If there are eight colors needed, then the channel provided for the access point is to channel 1, channel 3, channel 5, channel 7, channel 9, channel 11, channel 12 and channel 13.
9. If there are nine colors needed, then the channel provided for the access point is to channel 1, channel 3, channel 5, channel 7, channel 9, channel 10, channel 11, channel 12 and channel 13.
10. If there are ten colors needed, then the channel provided for the access point is to channel 1, channel 3, channel 5, channel 7, channel 8, channel 9, channel 10, channel 11, channel 12 and channel 13.
11. If there are eleven colors needed, then the channel provided for the access point is to channel 1, channel 3, channel 5, channel 6, channel 7, channel 8, channel 9, channel 10, channel 11, channel 12 and channel 13.
12. If there are twelve colors needed, then the channel provided for the access point is to channel 1, channel 3, channel 4, channel 5, channel 6, channel 7, channel 8, channel 9, channel 10, channel 11, channel 12 and channel 13.
13. If there are thirteen colors needed, then the channel provided for the access point is to channel 1, channel 2, channel 3, channel 4, channel 5, channel 6, channel 7, channel 8, channel 9, channel 10, channel 11, channel 12 and channel 13.

3.4 AN EXAMPLE OF VMA

We demonstrate the steps of the algorithm with a small example. The input graph is shows below in Figure 3.1 which has seven vertices labeled $V = \{1, 2, 3, 4, 5, 6, 7\}$. The algorithm required two coloring of the vertices using the set of colors $\{1, 2\}$ represented by red and green respectively.

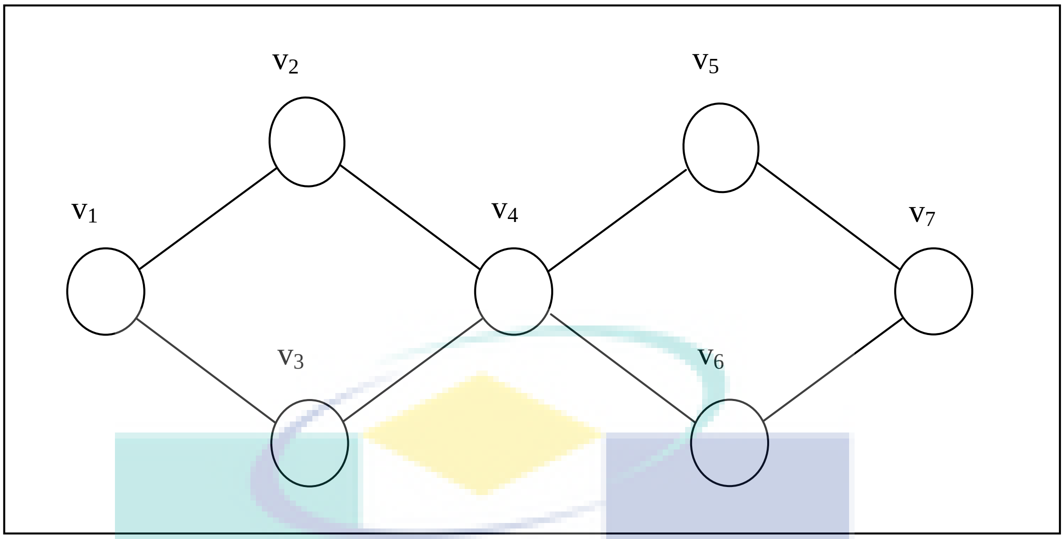


Figure 3.1: An example of graph

2) Arrange the vertex by decreasing order of degrees. Vertex with the highest degree is v_4 (which 4 edge) then v_1, v_2, v_3, v_5, v_6, v_7 each with 2 degree. Then the sequence is {v_4, v_1, v_2, v_3, v_5, v_6, v_7}. Table 3.2 shows vertex merge table first step.

Table 3.2: Vertex merge table first step

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
l_1	0	1	1	0	0	0	0
l_2	1	0	0	1	0	0	0
l_3	1	0	0	1	0	0	0
l_4	0	1	1	0	1	1	0
l_5	0	0	0	1	0	0	1
l_6	0	0	0	1	0	0	1
l_7	0	0	0	0	1	1	0

3) Choose the first uncolored vertex from the set. This causes vertex v_4 to be colored with color red, as depicted in Figure 3.2.

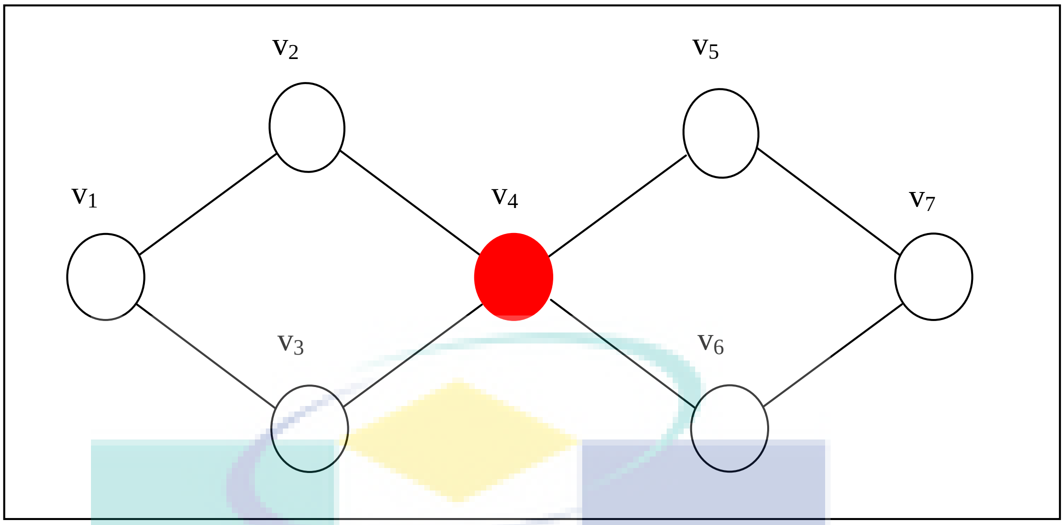


Figure 3.2: An example of graph with first step VMA

4) Merge with the first non-adjacent vertex, in this case the vertex are non-adjacent with v_4 are v_1 and v_7. Thus that the v_4 merger into v_1 and v_4. Then v_1 and v_7 same color with v_4, as depicted in Figure 3.3. Meanwhile, Table 3.3 shows vertex merge table second step.

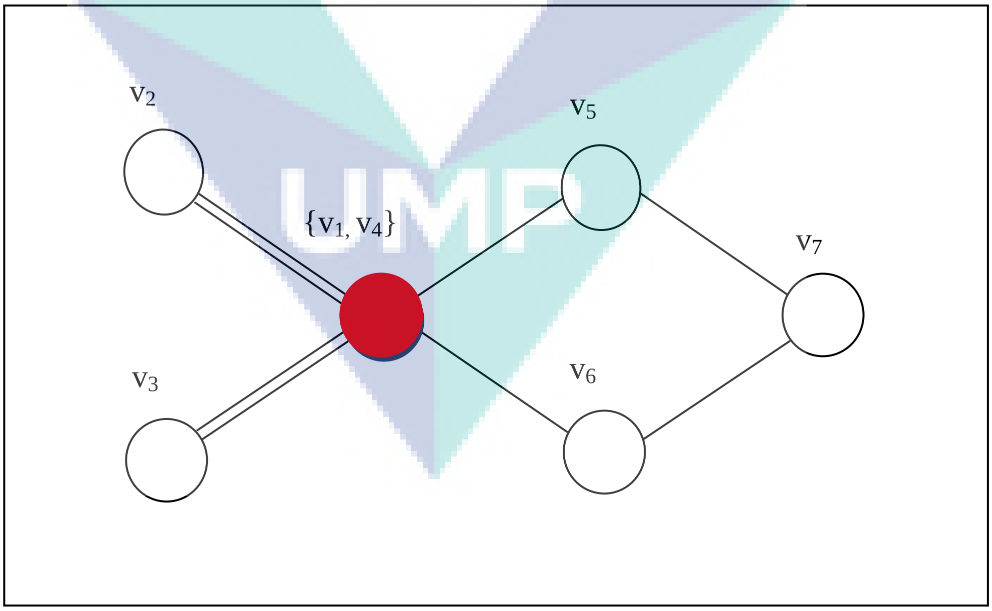


Figure 3.3: An example of graph with second step VMA

Table 3.3: Vertex merge table second step

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
(l_1, l_4)	0	2	2	0	1	1	0
l_2	1	0	0	1	0	0	0
l_3	1	0	0	1	0	0	0
l_5	0	0	0	1	0	0	1
l_6	0	0	0	1	0	0	1
l_7	0	0	0	0	1	1	0

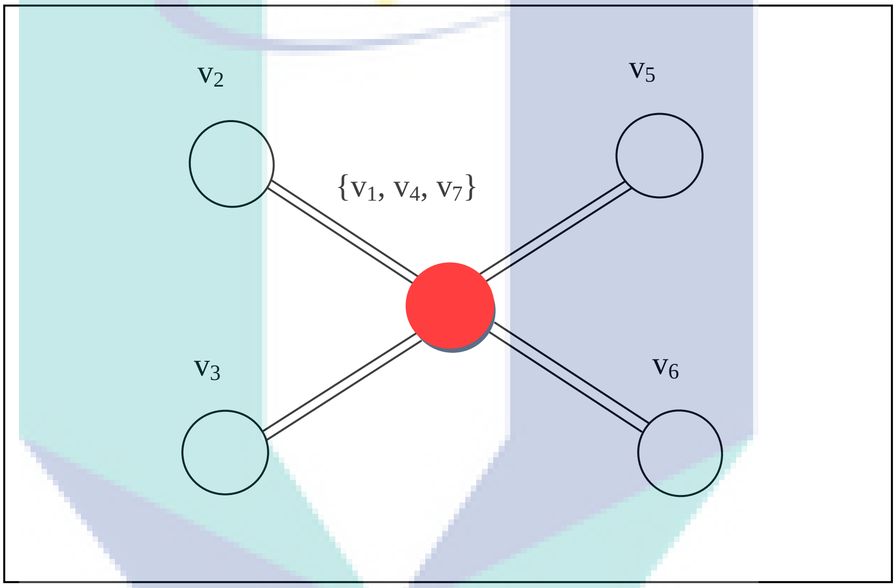


Figure 3.4: An example of graph with third step VMA

Table 3.4: Vertex merge table third step

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
(l_1, l_4, l_7)	0	2	2	0	2	2	0
l_2	1	0	0	1	0	0	0
l_3	1	0	0	1	0	0	0
l_5	0	0	0	1	0	0	1
l_6	0	0	0	1	0	0	1

5) If there is no more non-adjacent vertex, choose first uncolored vertex from the set.

a) Now sequence uncolored vertex which have the highest degree is $\{v_2, v_3, v_5, v_6\}$.

b) Choose the first uncolored vertex from the set. This cause's vertex v_2 to be colored with color two as depicted in Figure 3.5.

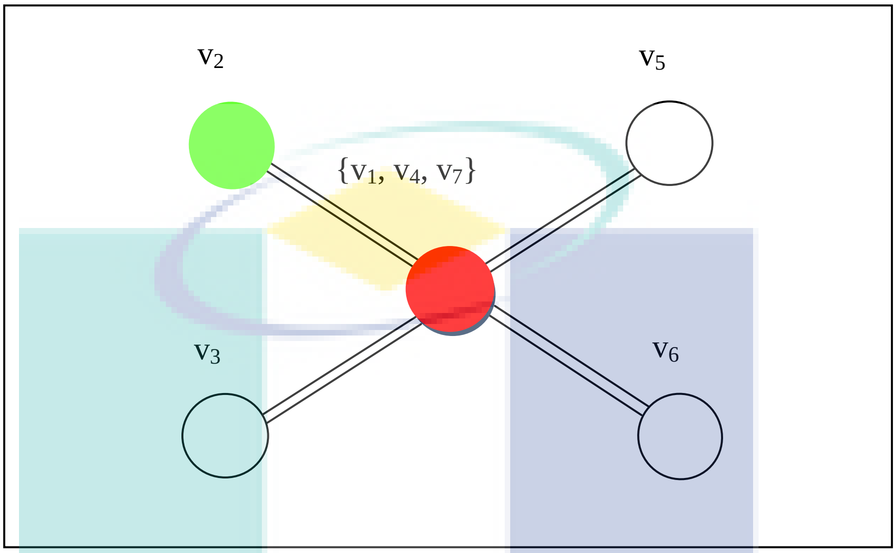


Figure 3.5: An example of graph with fourth step VMA

c) Merge with the first non-adjacent vertex, in this case the vertex are non-adjacent with v_2 are v_3, v_5 and v_6, so that the v_2 merger into v_3, v_5 and v_6. Then v_3, v_5 and v_6 same color with v_2, as depicted in Figure 3.6. Meanwhile, Table 3.6 shows the vertex merge table fourth step.

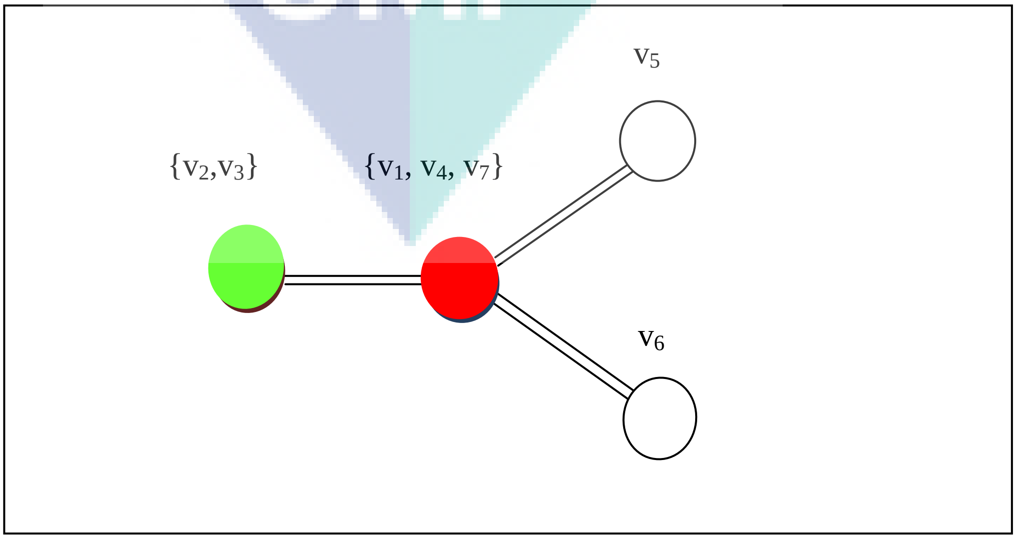


Figure 3.6: An example of graph with fifth step VMA

Table 3.5: Vertex merge table fourth step

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
(l_1, l_4, l_7)	0	2	2	0	2	2	0
(l_2, l_3)	2	0	0	2	0	0	0
l_5	0	0	0	1	0	0	1
l_6	0	0	0	1	0	0	1

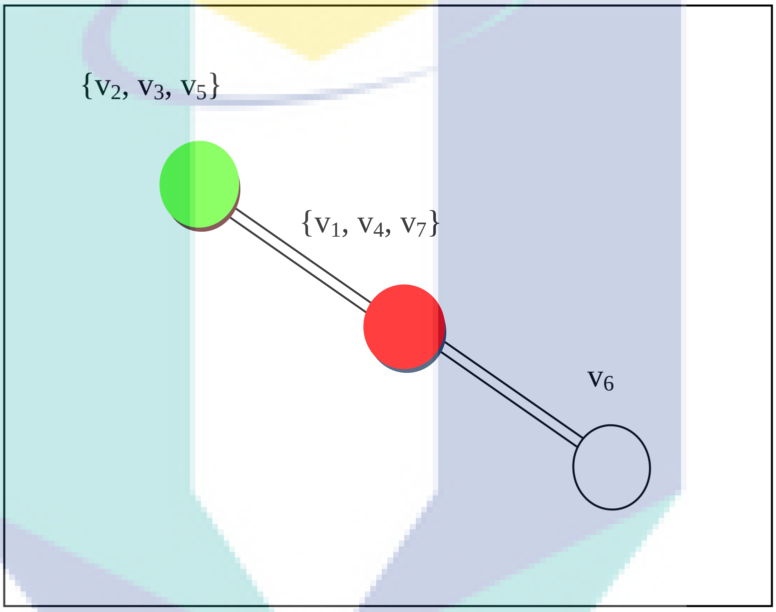


Figure 3.7: An example of graph with sixth step VMA

Table 3.6: Vertex merge table fifth step

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
(l_1, l_4, l_7)	0	2	2	0	2	2	0
(l_2, l_3, l_5)	2	0	0	3	0	0	1
l_6	0	0	0	1	0	0	1

Figure 3.8 shows the last step VMA with output two colors that are red and green. Meanwhile, Table 3.7 shows the last step vertex merge table with output all the vertex has been merged.

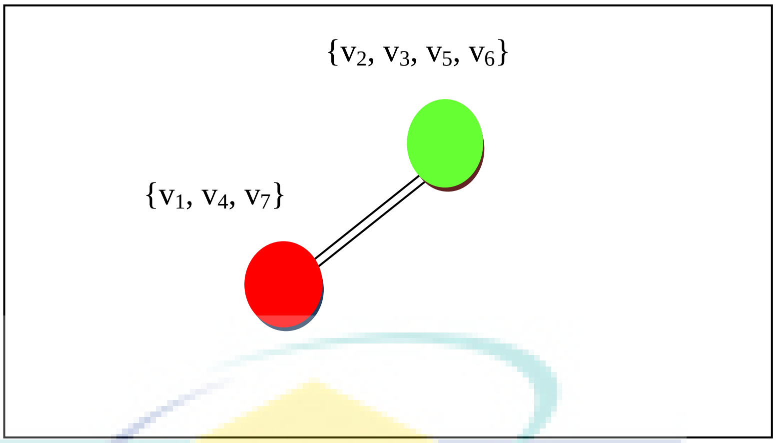


Figure 3.8: An example of graph with last step VMA

Table 3.7: Vertex merge table last step

	v_1	v_2	v_3	v_4	v_5	v_6	v_7
(l_1, l_4, l_7)	0	2	2	0	2	2	0
(l_2, l_3, l_5, l_6)	2	0	0	4	0	0	2

6) If all vertex are colored, thus the algorithm terminates, and the final coloring is given by $C(v_1)=1$, $C(c_2)=2$, $C(v_3)=2$, $C(v_4)=1$, $C(v_5)=2$, $C(v_6)=2$, $C(v_7)=1$.

Thus, from the Figure 3.8, the VMA requires two colors for coloring the graph. It means that required two channels for this case that is channel 1 and channel 6.

3.5 CONCLUSION

In this chapter, a new algorithm called Vertex Merge Algorithm (VMA) has been proposed for channel allocation in the Direct Sequence Spread Spectrum (DSSS). It is based on the graph coloring theory. This algorithm can be viewed as a merge the vertex if an only if the vertex non-adjacent. The VMA implicitly uses merge vertex and merge edge. A merge vertex is a set of vertex, which will be assigned the same color as any pair of vertex in that set is never connected.

The VMA concentrates on the operations between merge vertex and normal vertex. We try to merge the normal vertex with another vertex, and when the latter is a merge vertex. A reduction in adjacency checks is possible. These checks can be performed along merge edge instead of normal edges, whereby we can introduce significant savings. This is because the initial set of normal edges is folded into merge edge.

The coloring data is stored in a Vertex Merge Table (VMT). The initial VMT is the adjacency matrix of the graph; hence a unique color is assigned to each of the vertices. If the graph is not a complete graph, then it might be possible to reduce the number of necessary colors. This corresponds to the reduction of rows in the VMT. To reduce the rows, we introduce a Vertex Merge Operation (VMO), which attempts to merge two rows. When this is possible, the number of colors is decreased by one. When it is not, the number of colors remains the same. It is achievable only when two vertices are not connected by a normal edge or a merge edge.

In this chapter has demonstrate an example of VMA with input graph is shows in Figure 3.1 which have seven vertices labeled $V = \{1, 2, 3, 4, 5, 6, 7\}$. This algorithm required two coloring of the vertices using the set of colors $\{1, 2\}$ represented by red and green respectively. It means that required two channels for this example that is channel 1 and channel 6.

CHAPTER 4

EXPERIMENT AND RESULT ANALYSIS

4.1 INTRODUCTION

In this chapter, experiment and result analysis will be described. The purposes of these experiments are to illustrate the algorithm described in previous chapter and to shows that Vertex Merge Algorithm (VMA) can be used in practical applications.

4.2 HARDWARE AND SOFTWARE SPECIFICATION

The experiment requires both hardware and software components. To simulate among VMA, Dsatur and Welsh Powell required one computer with specification as shows in Table 4.1.

Table 4.1: Computer components specification

Hardware	Specifications
Processor	2.0 Giga Hertz Intel Core 2 Duo
Memory	1 Gigabyte
Hard Disk	320 Gigabyte
Wireless NIC	Invilink 802.b/g
Graphics Controller	Intel 4100

The experiment of VMA simulator, Dsatur simulator and Welsh Powell simulator was done in PHP programming and Eye of GNOME integrated with Ubuntu. PHP is selected because the command-line editing facilities and object oriented capabilities (Zandstra, 2010). The object oriented provides greater flexibility in dealing with background processes. Meanwhile, an automated Eye of GNOME is used for open and saved image. In this experiment, Linux Ubuntu 9.04 is used as a platform to computer (Ubuntu, 2010). All applications for computer are available to these particular Linux platforms. The applications include *gedit*, *vi* and *vim* editor. Table 4.2 shows the system development tools specification for this experiment.

Table 4.2: System development tools specification

System Development Tools	PHP 5.0
	Linux OS: Ubuntu 9.04
	Utilities : Eye of GNOME and GIMP (open and save image), *gedit* (editor text)

4.3 EXPERIMENT

The main process of the proposed VMA experiment involves the following steps:

1. Survey location the position of an access point.
   Survey location the position of an access point is important to determine number of required channel in network. Position of an access point is crucial the channel to be used in the access point.
2. Create the position of an access point to graph model.
   Position of an access point in network should be transform in graph model. Based on the graph model will be shows the relationship between access points.
3. Create the adjacent matrix from the graph model.
   The adjacent matrix can be created based on the graph model. Every vertex that adjacent will be assigned a value of 1, otherwise given a value of 0.
4. Testing data to obtain the number prediction of channel required.
   The value obtained on the graph model used as input to the simulator. The input processed using simulator to obtain the number prediction of channel required.

Figure 4.1 shows the flowchart of Welsh Powell algorithm. Meanwhile, Figure 4.2 and Figure 4.3 show the flowcharts of Dsatur and VMA algorithms respectively.

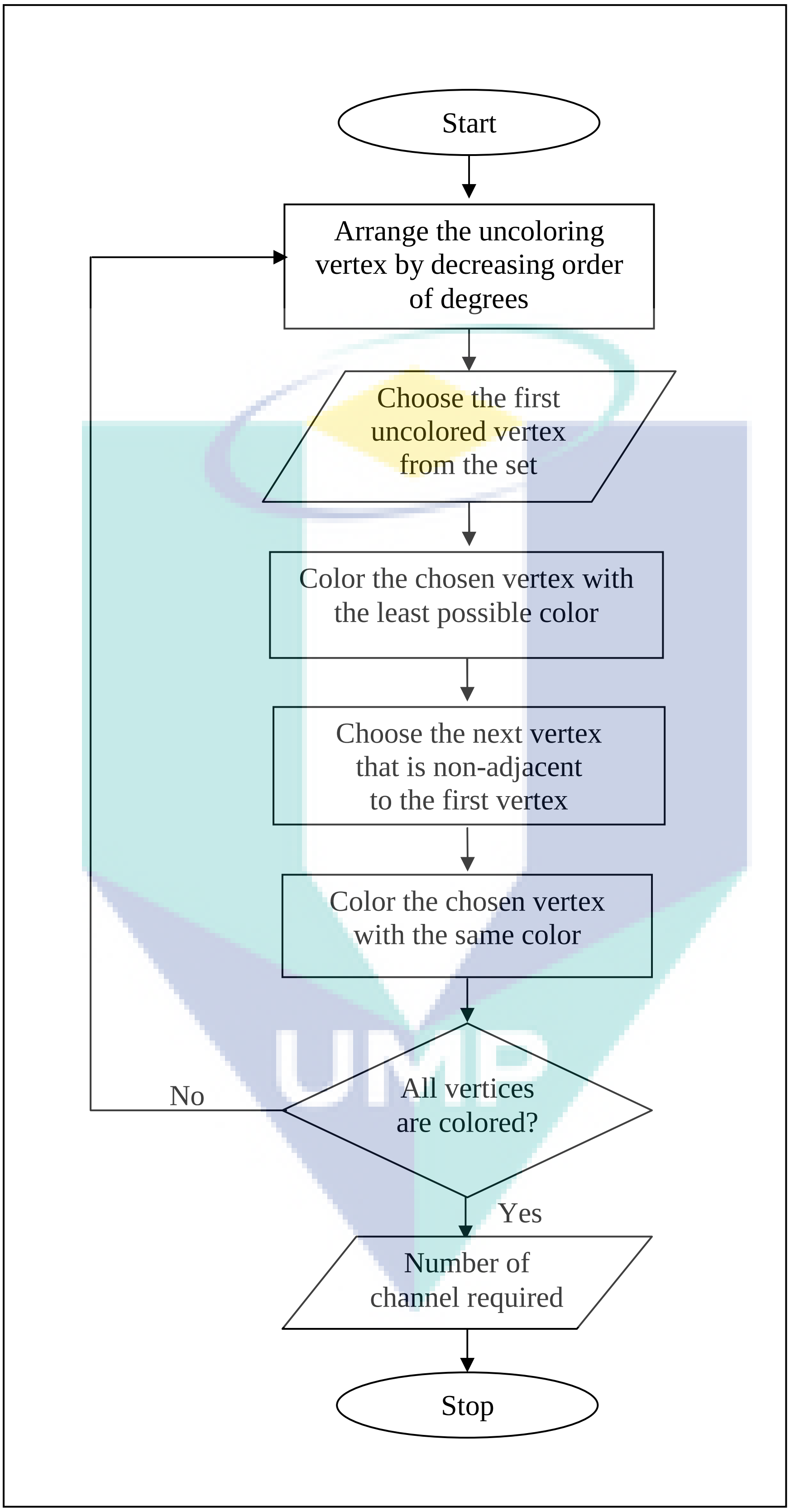


Figure 4.1: Flowchart of Welsh Powell algorithm

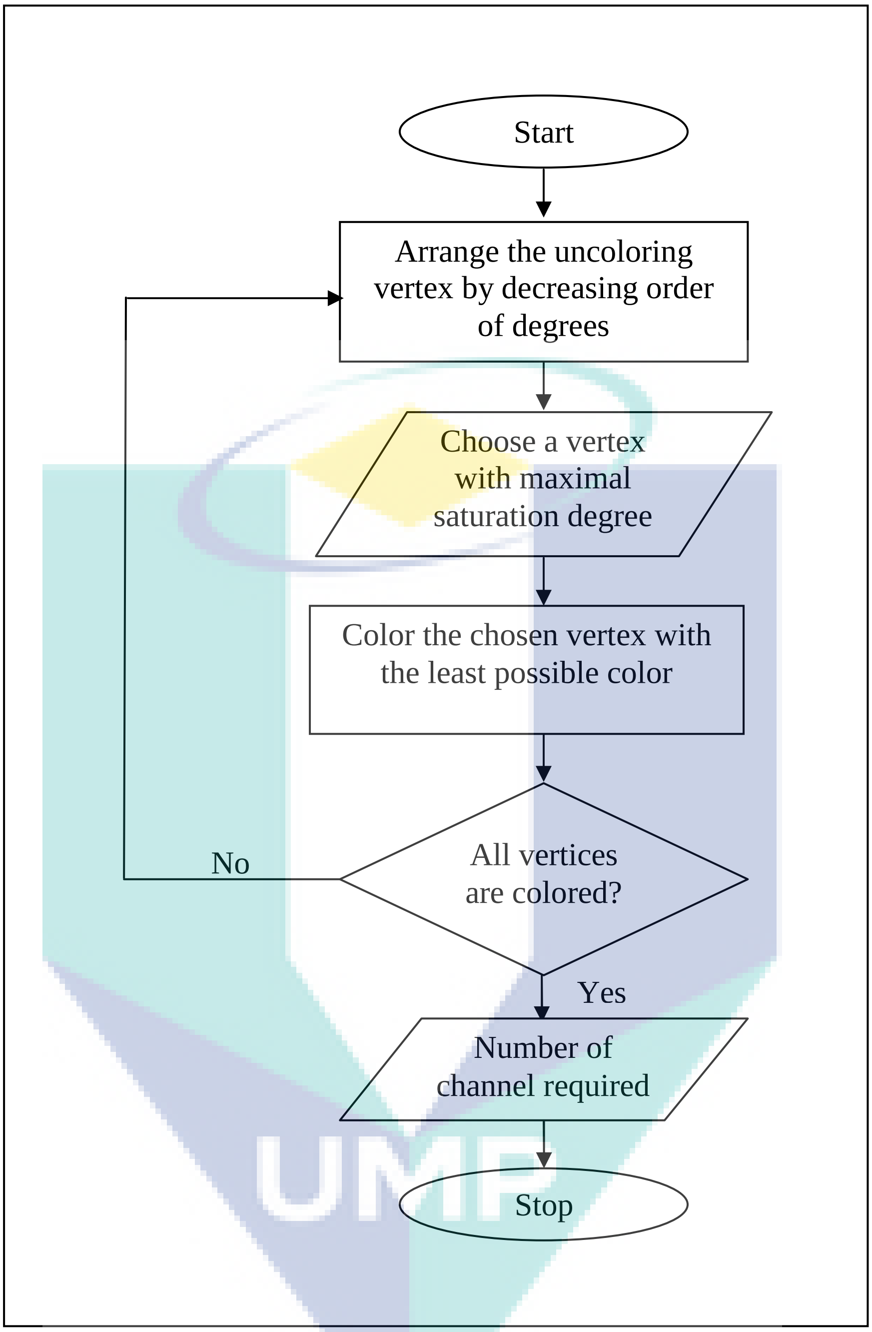


Figure 4.2: Flowchart of Dsatur algorithm

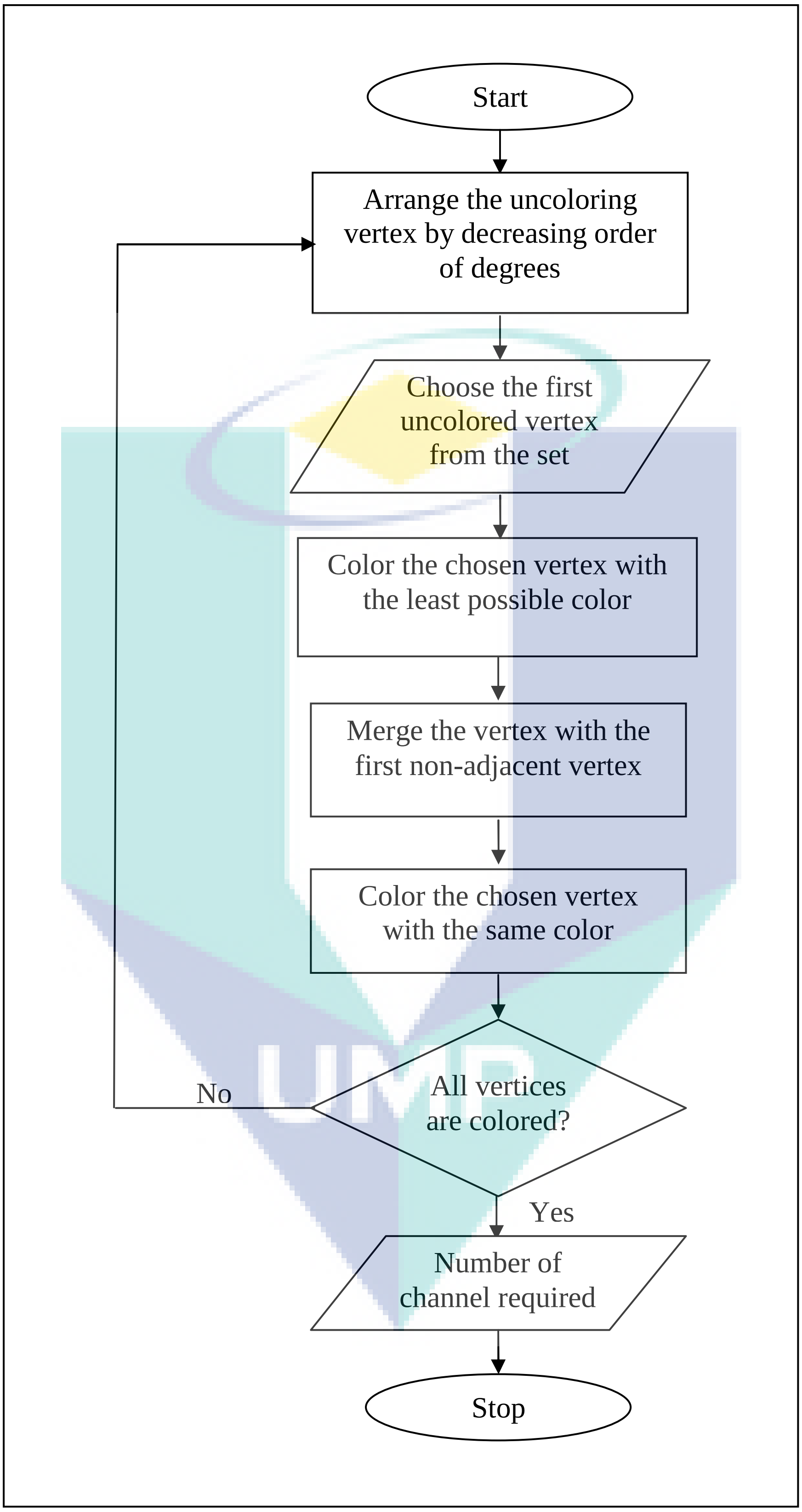


Figure 4.3: Flowchart the proposed algorithm

4.3.1 Coloring of Channel

In this section, coloring of channel will be described. Determining the color of each channel is very important, because every channel should have a different color. All the colors in this simulator uses RGB color model (Wikipedia, 2010). The RGB color model is an additive color model in which red, green, and blue light are added together in various ways to reproduce a broad array of colors. The name the model comes from the initials the three additive primary colors, red, green, and blue. Table 4.3 shows the color used for each channel. Meanwhile, estimate the required color for the entire channel is shown in Table 4.4.

Tabel 4.3: Required of color to channel

Allocation of Channel			
Channel	**Color**	**Color Code**	**Name of Color**
1		(255, 0, 0)	Red
2		(255, 255, 0)	Yellow
3		(255, 0, 255)	Magenta
4		(0, 255, 255)	Turquoise
5		(128, 128, 0)	Dark Yellow
6		(0, 255, 0)	Bright Green
7		(51, 51, 153)	Indigo
8		(153, 204, 255)	Pale Blue
9		(153, 51, 0)	Brown
10		(128, 0, 128)	Violet
11		(0, 0, 255)	Blue
12		(128, 0, 0)	Dark Red
13		(255, 102, 0)	Orange

Table 4.4: Estimation of color to channel

Number of Color	Allocation of Channel												
	1	2	3	4	5	6	7	8	9	10	11	12	13
1	●												
2	●					●							
3	●					●					●		
4	●				●				●				●
5	●			●			●			●			●
6	●			●			●		●		●		●
7	●		●		●		●		●		●		●
8	●		●		●		●		●		●	●	●
9	●		●		●		●		●	●	●	●	●
10	●		●		●		●	●	●	●	●	●	●
11	●		●		●	●	●	●	●	●	●	●	●
12	●		●	●	●	●	●	●	●	●	●	●	●
13	●	●	●	●	●	●	●	●	●	●	●	●	●

4.3.2 Create VMA, Dsatur and Welsh Powell Script

The */home* directory contains the users' home directories. By separating home directories to users own directory tree, the backup process becomes easier (Lars, 2010). VMA script is created under a */home/handrizal/program/vma/* directory. Dsatur script is created under a */home/handrizal/program/dsatur/* directory. Meanwhile, Welsh Powell script is created under a */home/handrizal/program/welshpowell/* directory. In this experiment, a particular directory */home/handrizal/program/vma/* is ordered to collect any input and output VMA simulation. A particular directory */home/handrizal/program/dsatur/* is ordered to collect any input and output Dsatur simulation. Meanwhile, directory */home/handrizal/program/welshpowell/* is ordered to collect any input and output Welsh Powell simulation during the experiments. The entire simulator has been created under PHP programming.

In particular, the user privilege is needed to create or modify VMA script. Figure 4.4 shows VMA script that has been created in */home/handrizal/program/vma/* directory.

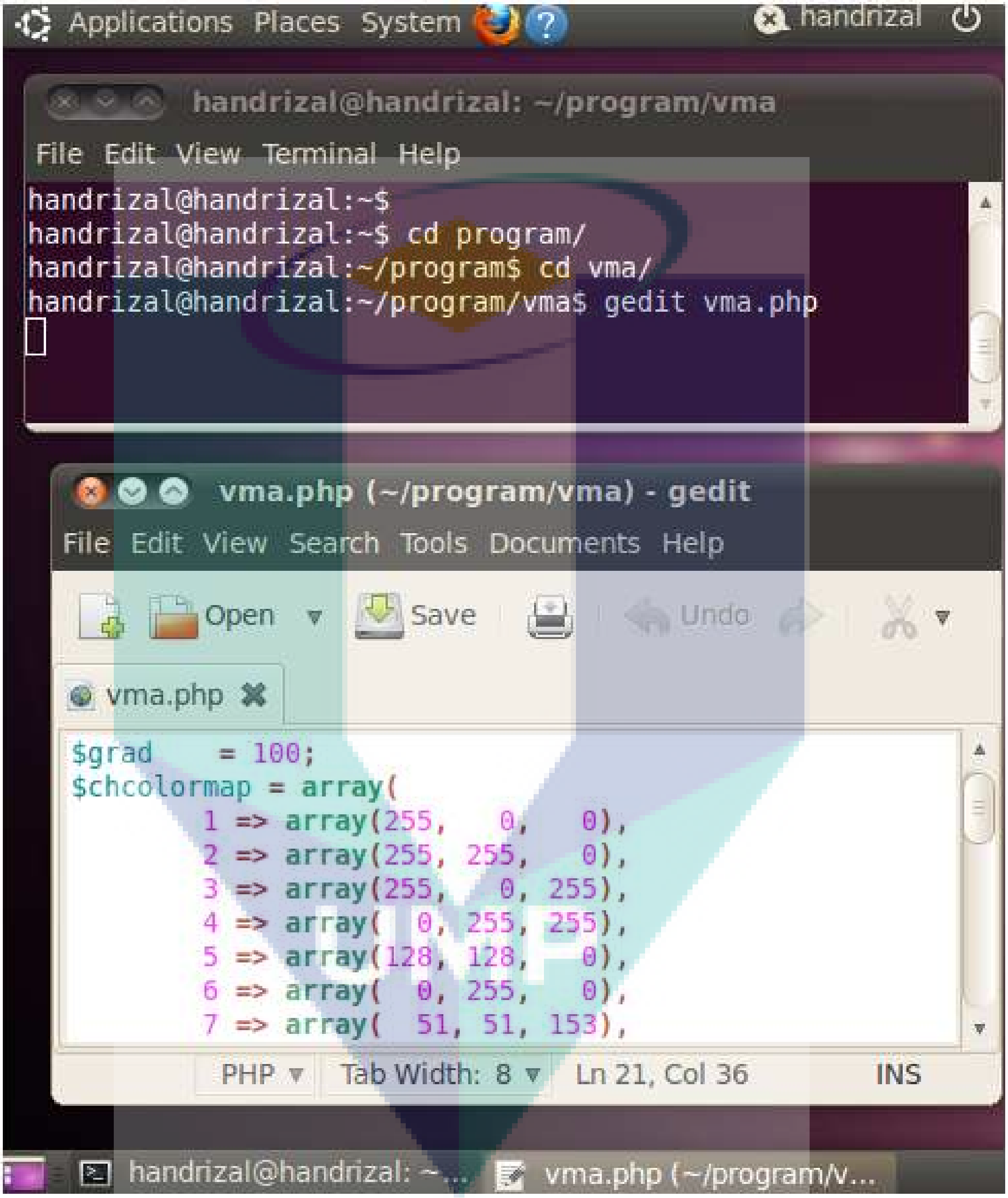


Figure 4.4: VMA simulator script

4.3.3. A Simulation Example

In previous chapters, the related woks associated with channel allocation have been discussed. VMA provides the minimum number of required channel. To give a better intuition of how to allocate the channel through VMA, VMA simulator, Dsatur simulator and Welsh Powell simulator is developed. A simulator is defined as an attempt to model a real-life or hypothetical situation on a computer so that it can be studied to see how the system works. By changing variables, predictions may be made about the behavior the system (Wikipedia, 2011). Usually, it provides some services either for the system as a whole or for the user application (The American Dictionaries, 2010). VMA Simulator, Dsatur Simulator and Welsh Powell simulator are started (and stopped) when a user requires this simulator. VMA Simulator, Dsatur simulator and Welsh Powell simulator run with the user privilege. VMA script is creating under a */home/handrizal/program/vma/* directory. Dsatur script is creating under a */home/handrizal/program/dsatur/* directory. Meanwhile, Welsh Powell script is creating under a */home/handrizal/program/welshpowell/* directory. VMA simulator, Dsatur simulator and Welsh Powell simulator script is created in order to determine the number of channel allocation required.

Several experiments have been conducted to shows that VMA simulator can be getting good results. Next, the simulator scripts the experiments that have conducted and the results will be described in the following sections. Figure 4.5 shows that VMA simulator status is ready to run.

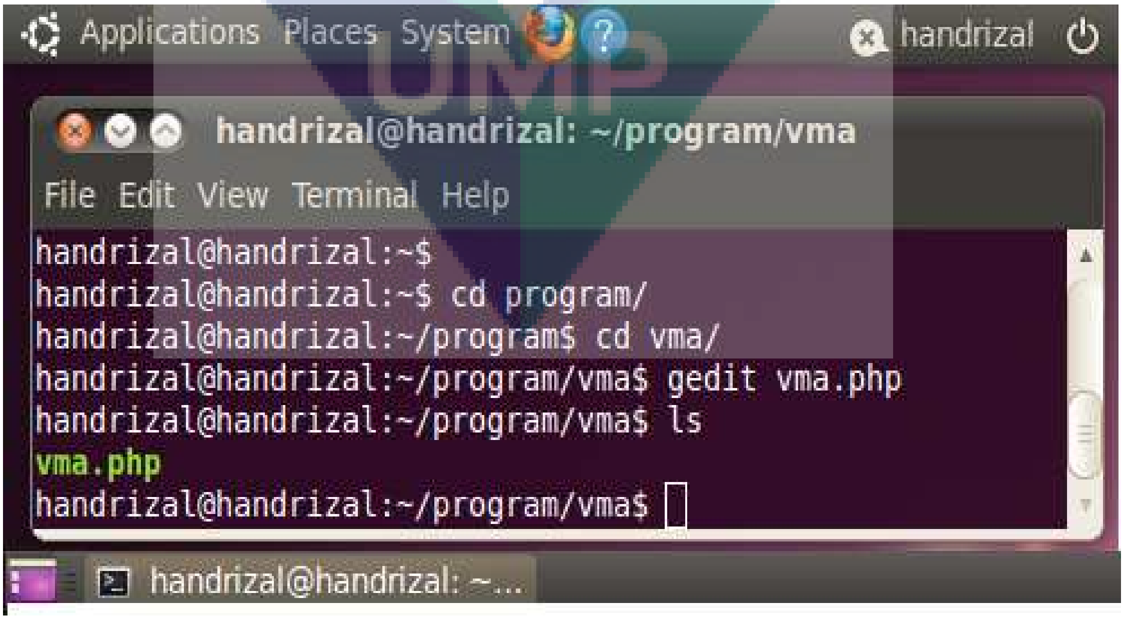


Figure 4.5: VMA simulator ready to run

In this simulation example, the VMA simulator is given input four of access points (AP) as depicted in Figure 4.6.

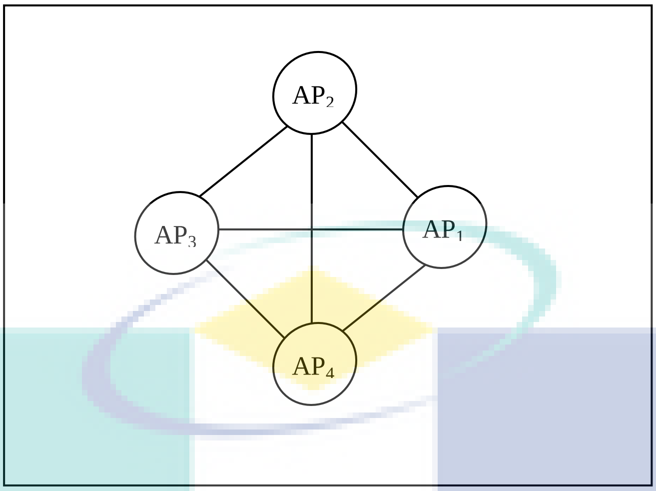


Figure 4.6: An example graph to input VMA simulator

In Figure 4.6 there are four access points that adjacent to each other namely AP_1, AP_2, AP_3, and AP_4. In this example, the graph is converted to the adjacent matrix as depicted in Table 4.5.

Table 4.5: Adjacent matrix of example graph

	AP_1	AP_2	AP_3	AP_4
AP_1	0	1	1	1
AP_2	1	0	1	1
AP_3	1	1	0	1
AP_4	1	1	1	0

Figure 4.7 shows the *gedit* application to input the adjacent matrix in VMA simulator. Meanwhile, Figure 4.8 shows the adjacent matrix has been input in *gedit* application.

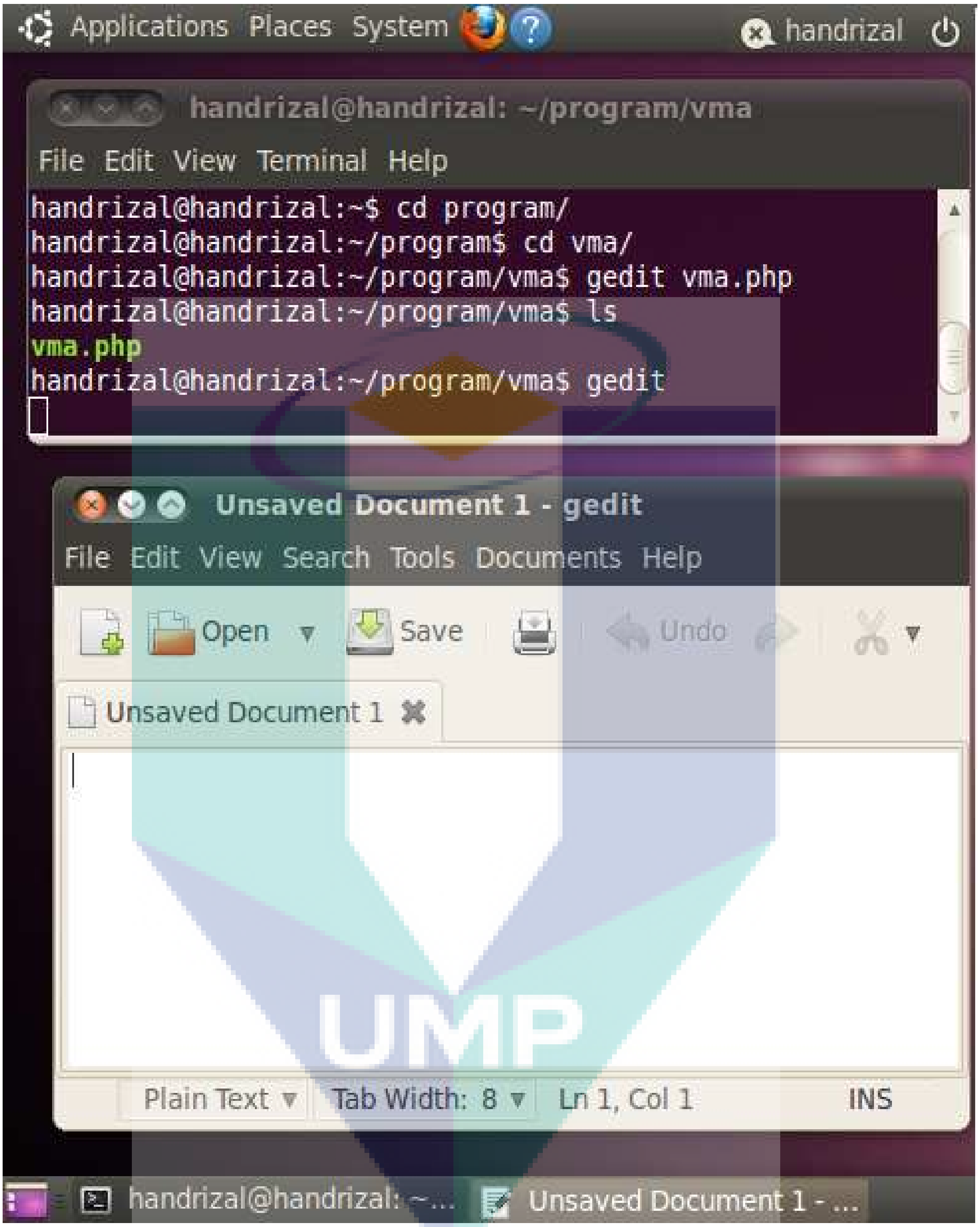


Figure 4.7: Open *gedit* to input data

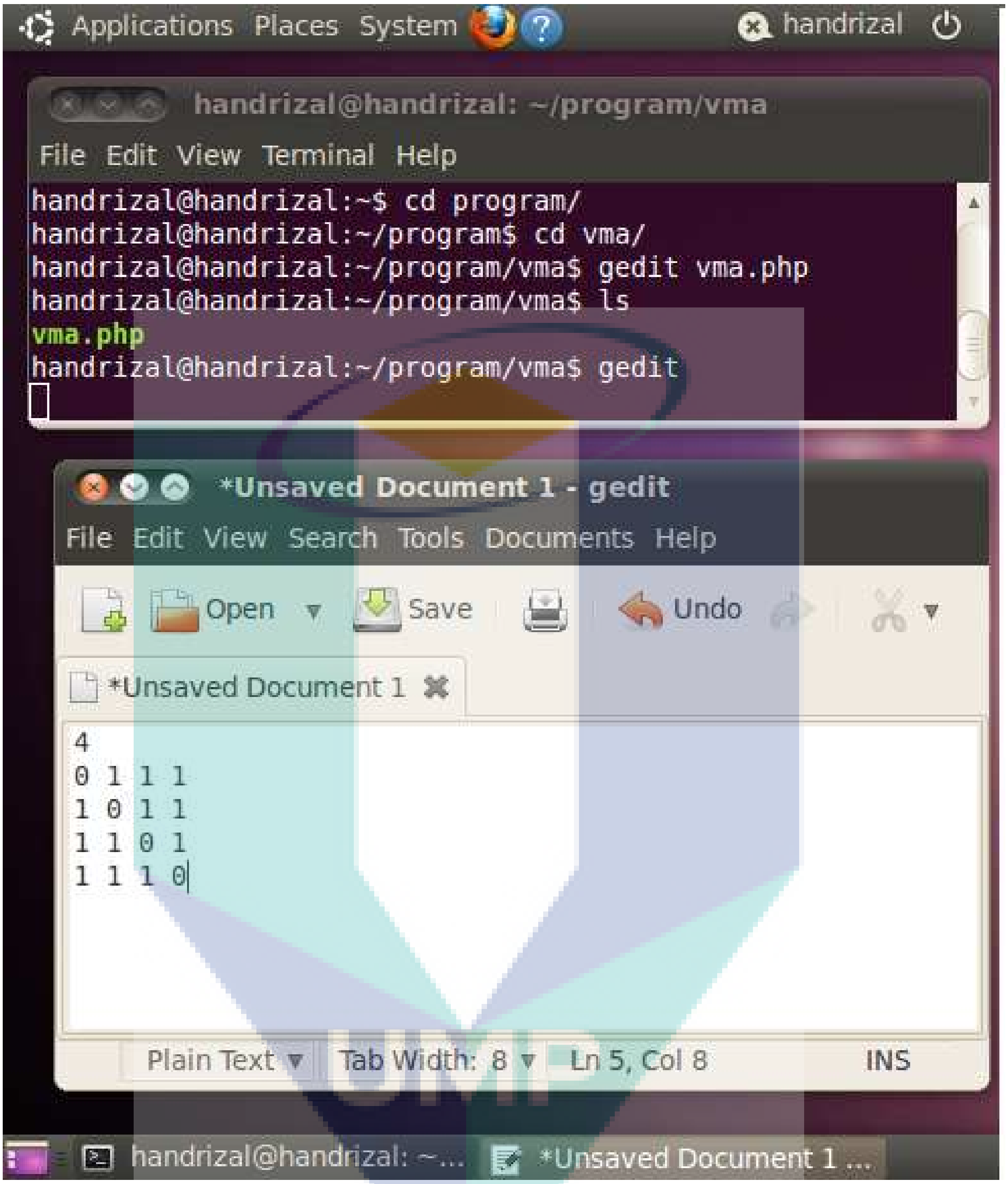


Figure 4.8: Input adjacent matrix to *gedit*

In Figure 4.8, number 4 in the first line of *gedit* application is total the access points. After all data the adjacent matrix has been input to *gedit* application, then save the data in *txt* format on */home/handriza/program/vma/* directory. In this example, *text.txt* is named of data as depicted in Figure 4.9 and Figure 4.10.

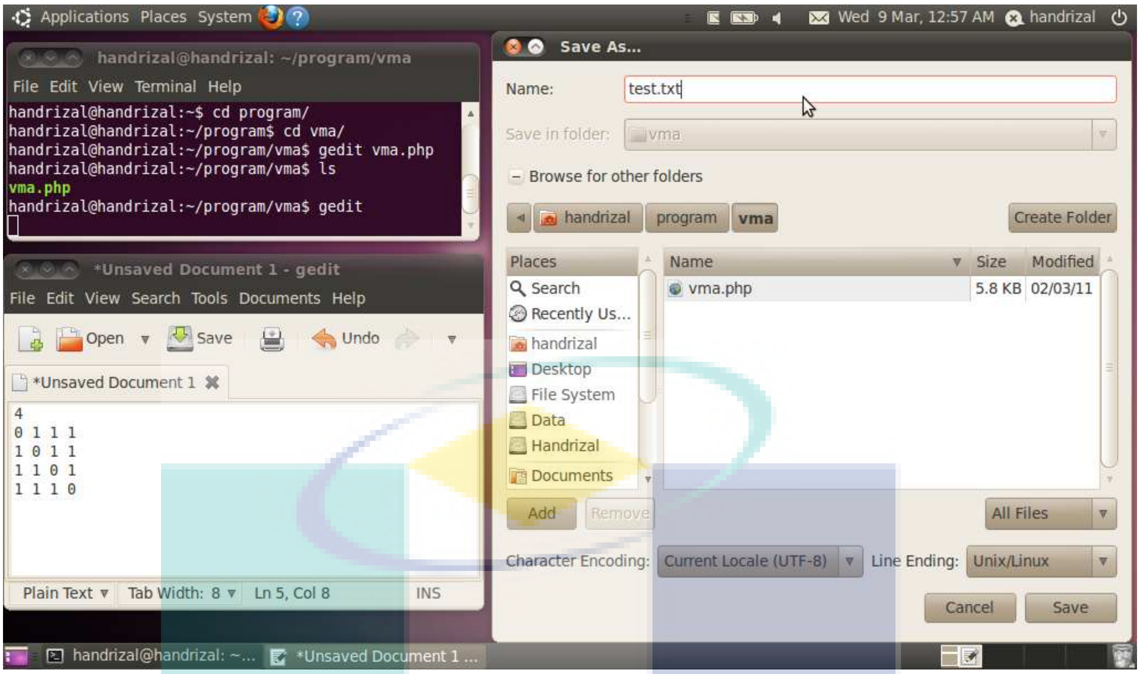


Figure 4.9: Save file input of VMA simulation with *txt* format

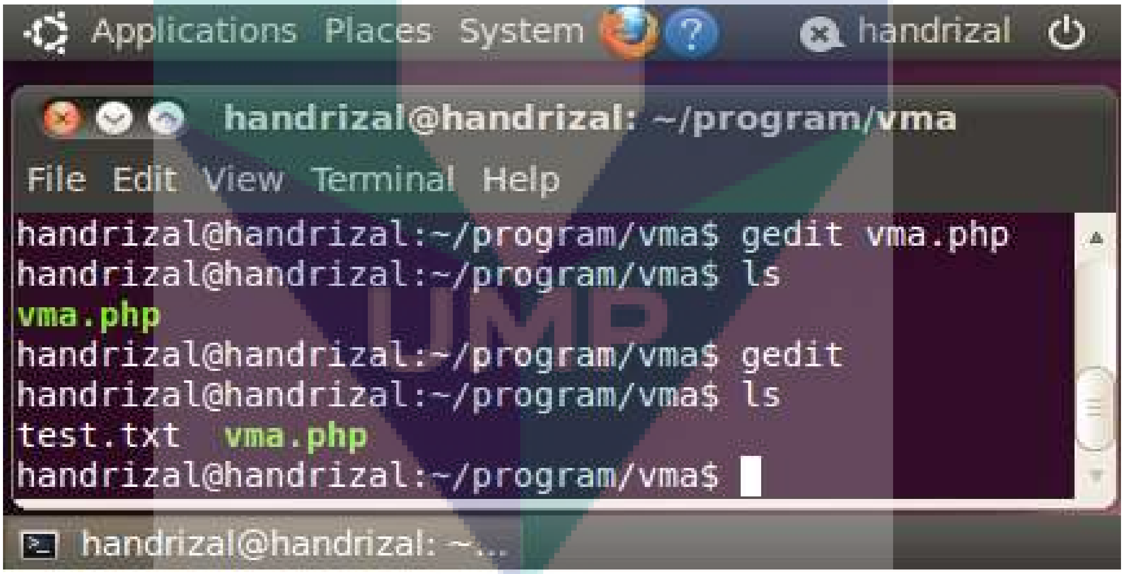


Figure 4.10: Input file of VMA simulation has been saved

In Figure 4.10 the input data (*test.txt*) already in */home/handrizal/program/vma/* directory. This data is used as input to the VMA simulator. The syntax used to enter data on the VMA simulator is command in the following text. *php(spacebar)name_of_simulator(spacebar)name_of_input_data*. An example, *php*

VMA.php test.txt as depicted in Figure 4.11. Next, the user presses *enter* to execute the command.

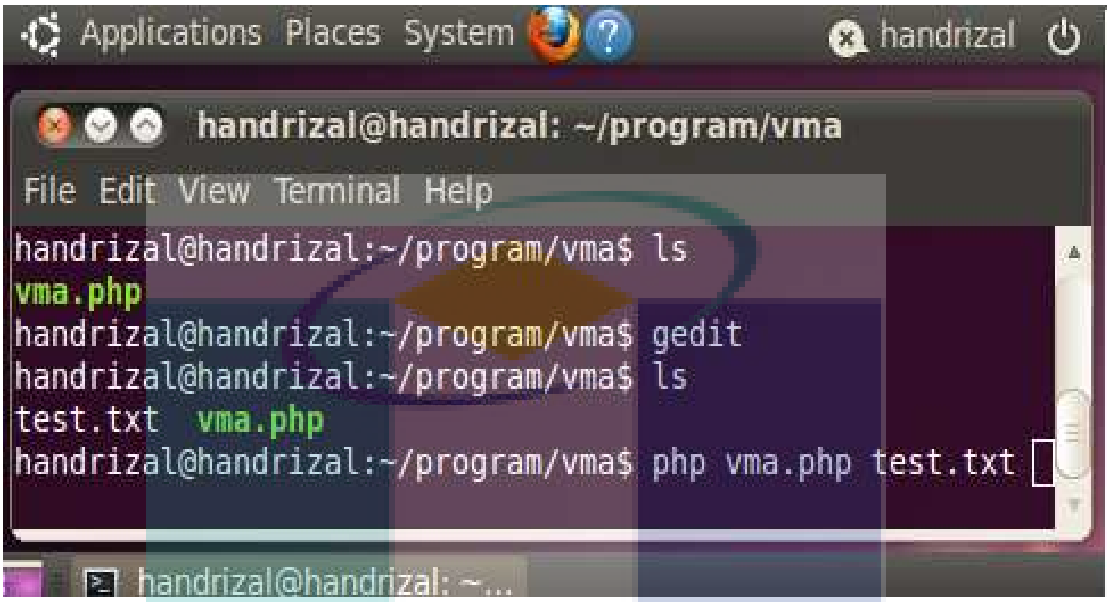


Figure 4.11: Test input file to VMA simulation

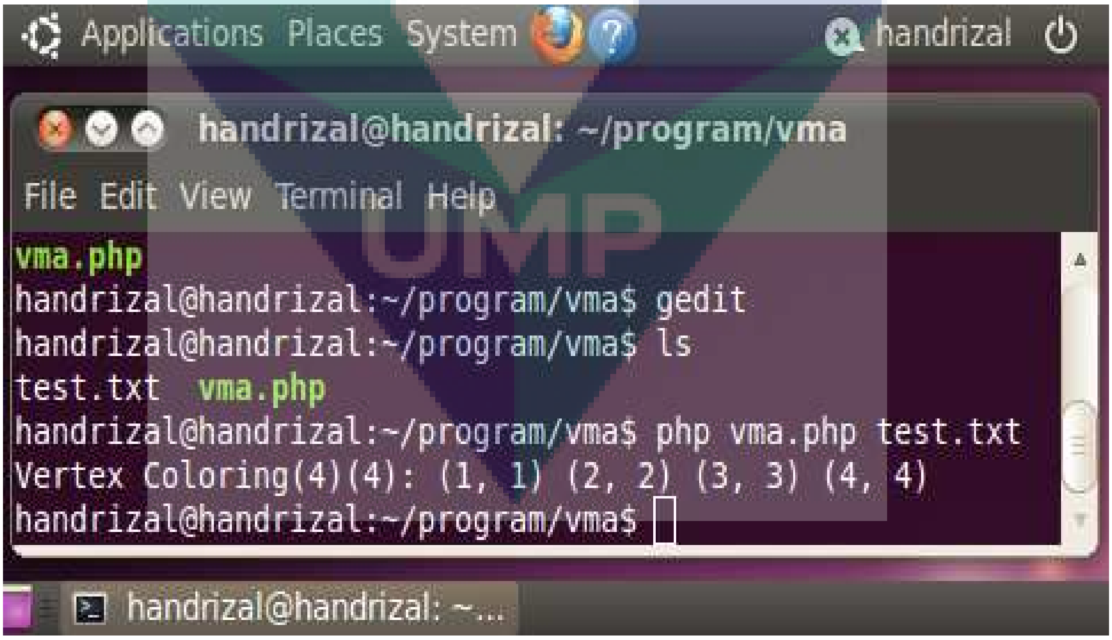


Figure 4.12: Output of VMA simulation in text format

Figure 4.12 shows output of VMA simulation in the text format that is,

Vertex Coloring (4) (4): (1, 1) (2, 2), (3, 3) (4, 4)

The output can be shows that:

1. *Vertex Coloring (4)*, which means number 4, is total of an access point.
2. *Vertex Coloring (4) (4)*, which mean number 4 first are total of an access point; number 4 second is total of channel required. In this step, VMA simulator chose column 4 in Table 4.4 as depicted in Table 4.6.

Table 4.6: Estimation of color to four required channels

Number of Color	**Allocation of Channel**												
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
4													

3. *(1, 1) (2, 2), (3, 3) (4, 4)*, which means number 1 first is AP_1; number 1 second is the number of channel allocation, etc. Table 4.6 shows:

   a. AP_1 colored , which mean that AP_1 is allocated to channel 1.
   b. AP_2 colored , which mean that AP_2 is allocated to channel 5.
   c. AP_3 colored , which mean that AP_3 is allocated to channel 9.
   d. AP_4 colored , which mean that AP_4 is allocated to channel 13.

After input data to VMA simulator, then on */home/handrizal/program/vma/* directory there are new files that are *input.png* and *output.png* as depicted in Figure 4.13. The files *input.png* and *output.png* are output of VMA simulator in image format. *PNG* extension is abbreviation of Portable Network Graphic. PNG is an open, extensible image format with lossless compression (Portable Network Graphic, 2010).

The *input.png* and *output.png* files contain an image the graph. The *input.png* and *output.png* generated from data input the adjacent matrix. The image of *input.png* is the uncolored graph. Meanwhile, image of *output.png* is the graph has been coloring.

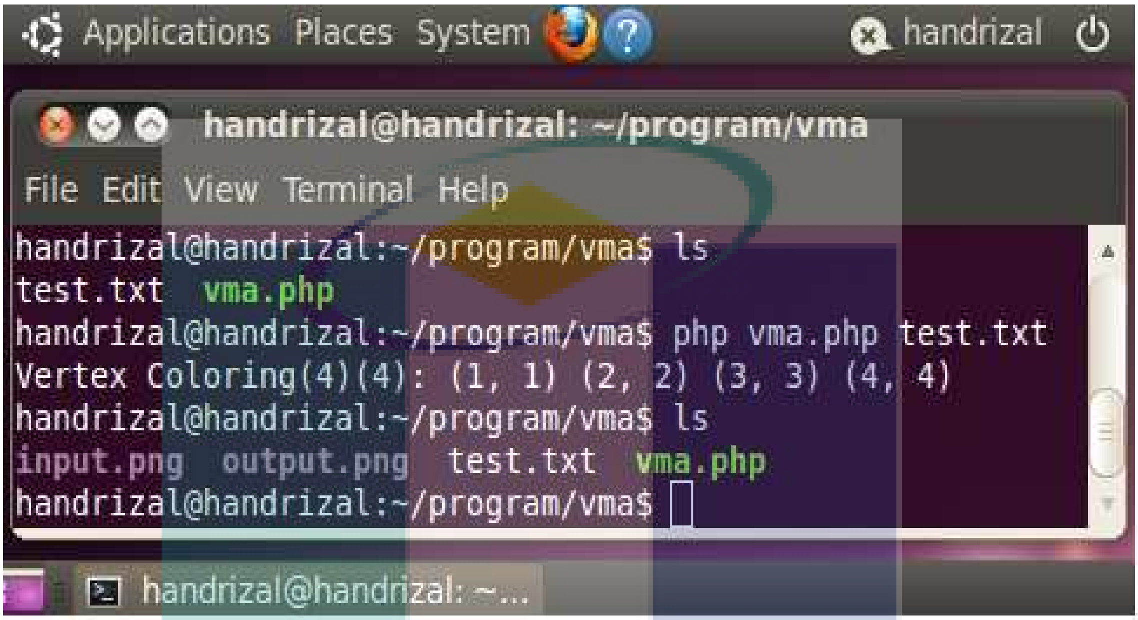


Figure 4.13: Input and Output have been included in the folder of VMA

The image *input.png* and *output.png* can be shows using *Eye of GNOME* application. The syntax used to viewer *input.png* and *output.png* is command the following text. *eog(spacebar)name_of_file.* An example, *eog input.png* and *eog output.png* as depicted in Figure 4.14 and Figure 4.15. Next, the user presses *enter* to execute the command.

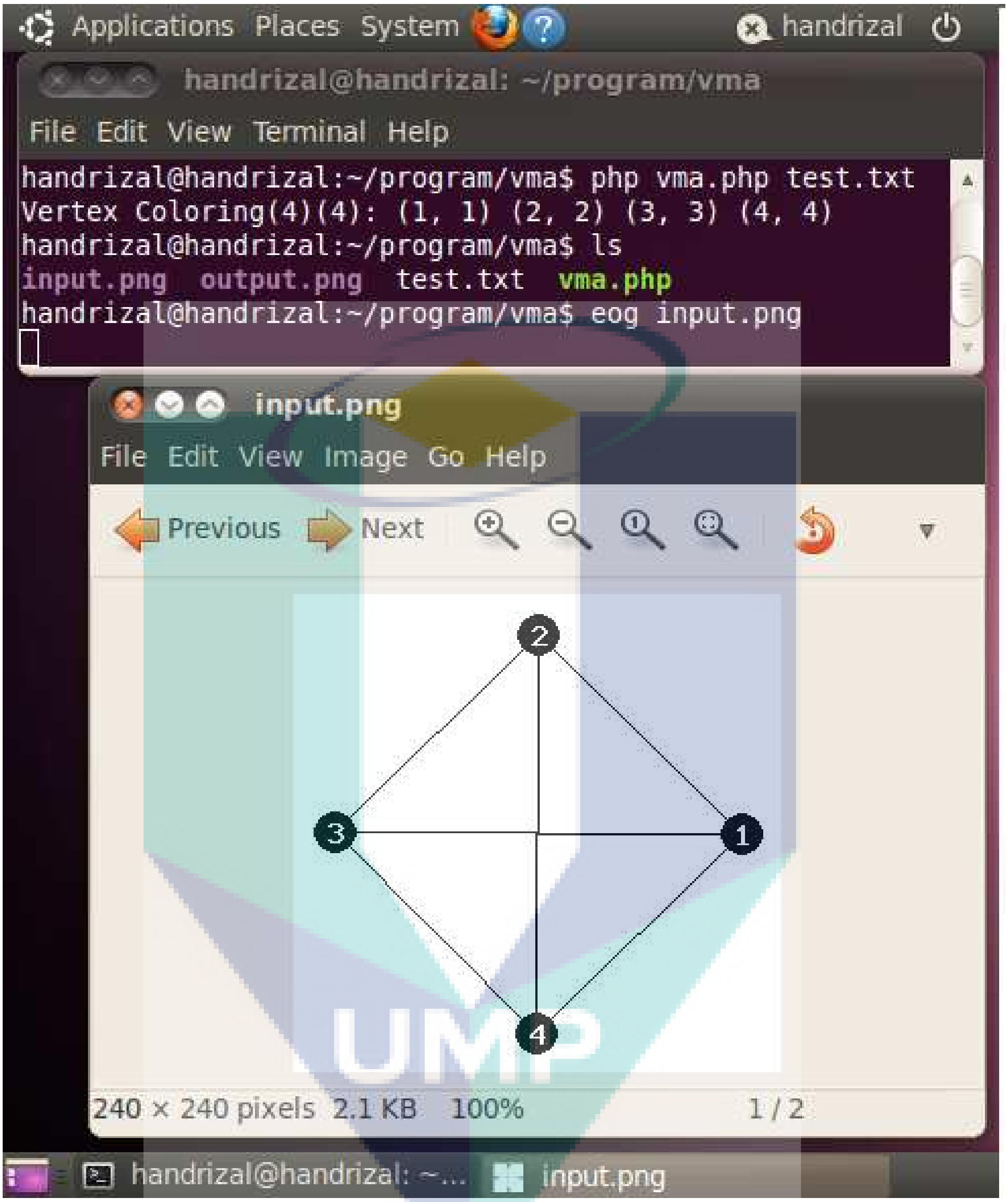


Figure 4.14: Output of VMA simulation in the graph model

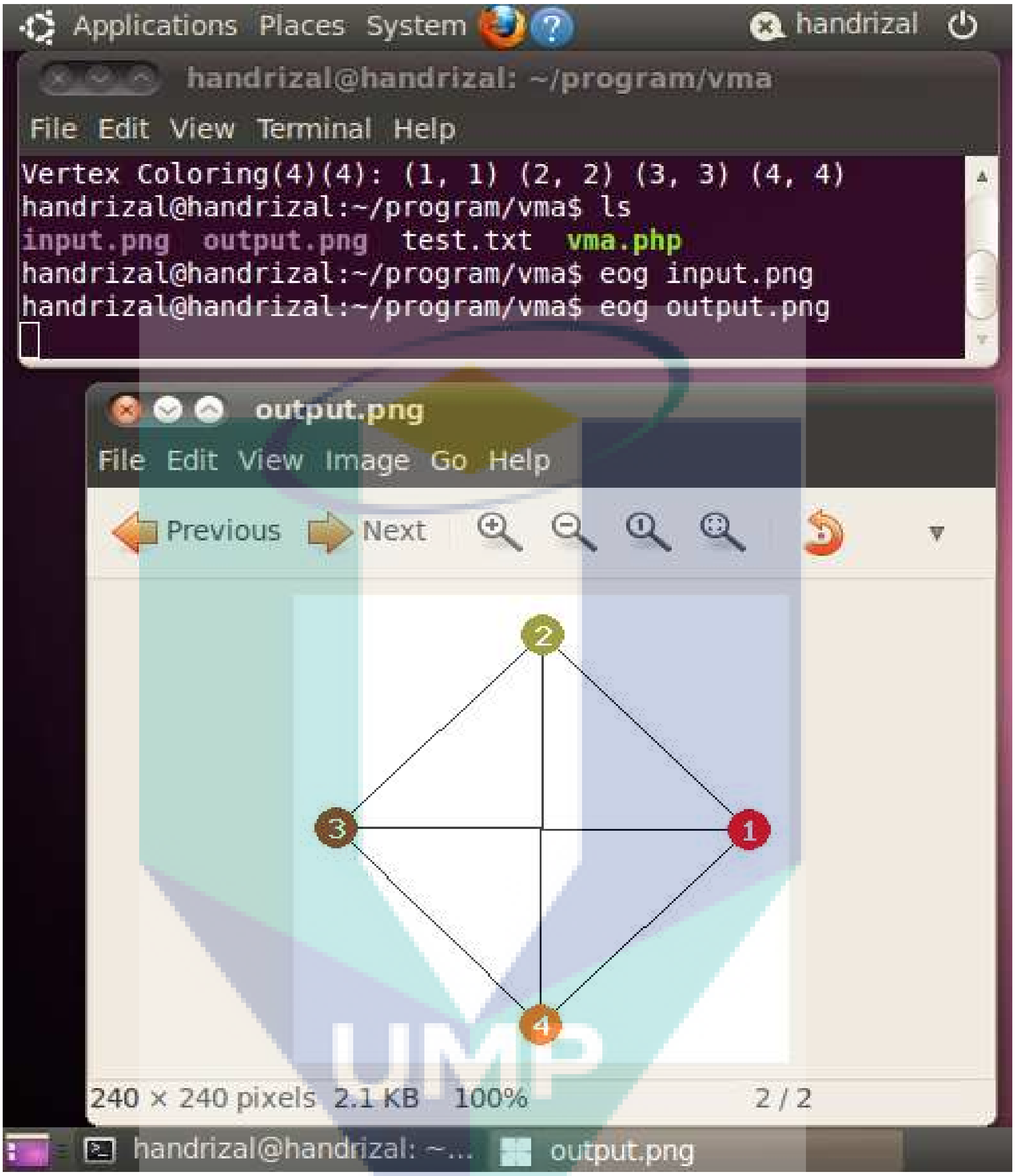


Figure 4.15: Output of VMA simulation in the graph coloring format

4. 4 RESULTS AND DISCUSSION

In this section, the experiment for VMA is compared with Welsh Powell and Dsatur algorithms are described. The experiment involves the phases that have described in Chapter 3 section 3.2. The different cases are considered in the experiment. In the first case, considers a simple graph, conversely, the complex graph is considered in the second case. In each case, the simulator finds a minimum number of channel required equal to the chromatic number χ(G). All results in the table output experiment based on output the simulator. The experiment used three simulators that are:

- *welshpowell.php* is the main source code to compile and running Welsh Powell simulator.
- *dsatur.php* is main the source code to compile and running Degree of Saturation (Dsatur) simulator.
- *vma.php* is main the source code to compile and running Vertex Merge Algorithm (VMA) simulator.

4.4.1 Case 1: Simple Graph

In this section, three simple graphs will be experimented, that is the Octahedron, the Cube and the Kuratowski bipartite graph K3,3.

1. The Octahedron

The Octahedron is a graph which has six vertices (Ashay, 2010). Table 4.7 shows an input of the octahedron graph and its adjacent matrix.

Table 4.7: Input of the Octahedron

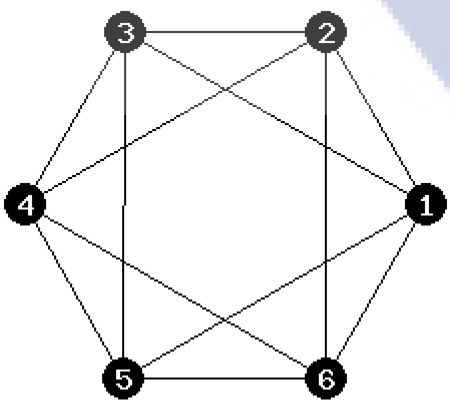
The graph	Adjacent matrix
	0 1 1 0 1 1 1 0 1 1 0 1 1 1 0 1 1 0 0 1 1 0 1 1 1 0 1 1 0 1 1 1 0 1 1 0

Table 4.8 shows output of the octahedron graph and number of channel required.

Table 4.8: Output of the Octahedron

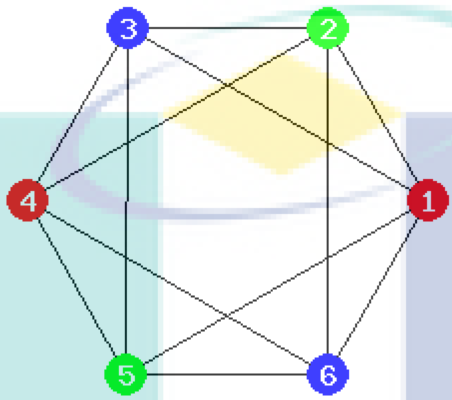
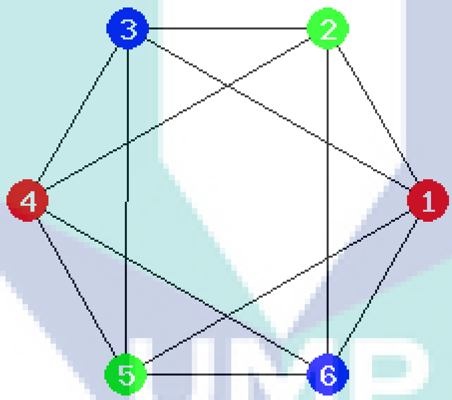
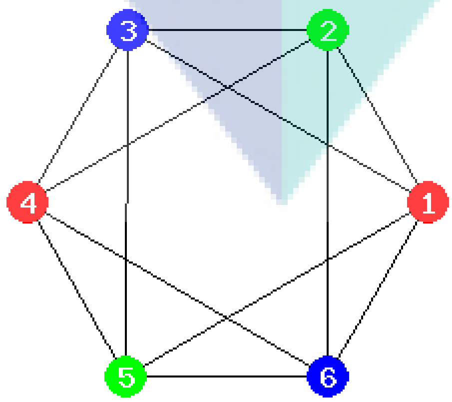
Simulator	**Output**	**Number of channel** $\chi(G)$
Welsh Powell		$\chi(G) = 3$ C(1,4)=channel 1, C(2,5)=channel 6, C(3,6)=channel 11.
Dsatur		$\chi(G) = 3$ C(1,4)=channel 1, C(2,5)=channel 6, C(3,6)=channel 11.
VMA		$\chi(G) = 3$ C(1,4)=channel 1, C(2,5)=channel 6, C(3, 6)=channel 11.

Table 4.8 shows that output the experiment given same result for all simulators, which are required three colors for coloring the graph. Its means required three channels in DSSS for this case, that is:

- channel 1 for AP_1 and AP_4
- channel 6 for AP_2 and AP_5
- channel 11 for AP_3 and AP_6

The channels that given to the access point are channel 1, 6 and 11 are non-overlapping channel. In this case, the entire of algorithm provides the channel with no interference between access points.

2. The Cube

The cube is a graph which has eight vertices (Ashay, 2010). Table 4.9 shows an input of the cube graph and adjacent matrix.

Table 4.9: Input of the Cube

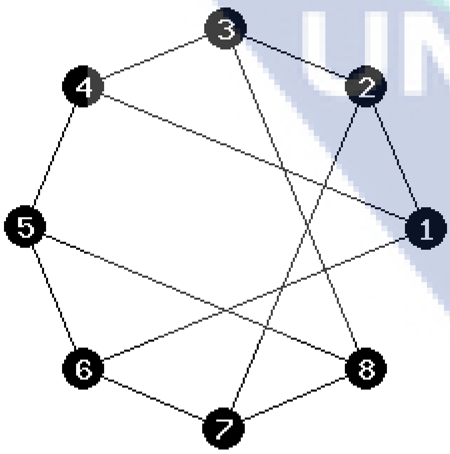
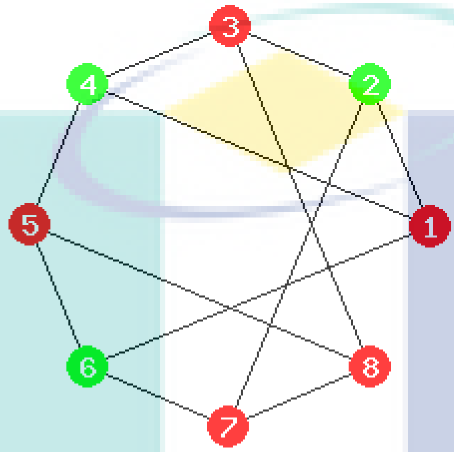
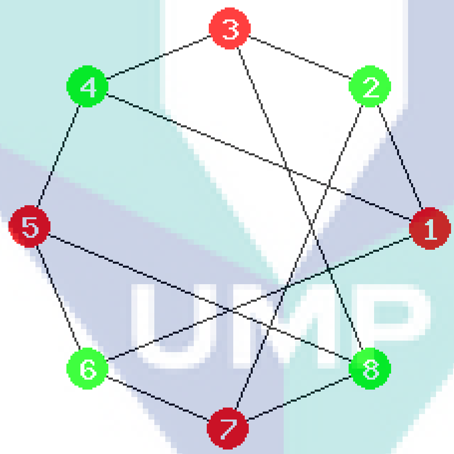
The graph	Adjacent matrix
	0 1 0 1 0 1 0 0 1 0 1 0 0 0 1 0 0 1 0 1 0 0 0 1 1 0 1 0 1 0 0 0 0 0 0 1 0 1 0 1 1 0 0 0 1 0 1 0 0 1 0 0 0 1 0 1 0 0 1 0 1 0 1 0

Table 4.0 shows output of the cube graph and number of channel required.

Table 4.10: Output of the Cube

Simulator	**Output**	**Number of channel** ***χ(G)***
Welsh Powell		*χ(G) = 2* *C*(1,3,5,7,8)=channel 1, *C*(2,4,6)=channel 6.
Dsatur		*χ(G) = 2* *C*(1,3,5,7)=channel 1, *C*(2,4,6,8)=channel 6.

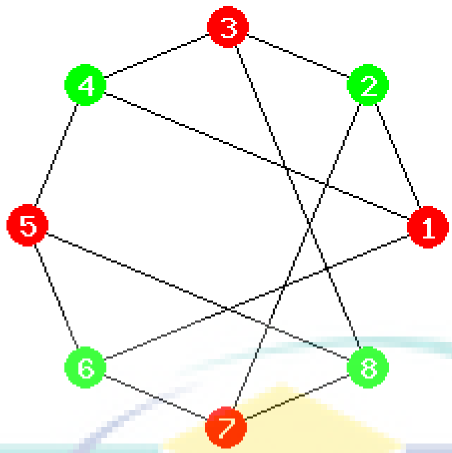
VMA		$\chi(G) = 2$ C(1,3,5,7)=channel 1, C(2,4,6,8)=channel 6.

Table 4.10 shows that output of the experiment given same result for all algorithms, which are required two colors for coloring the graph. Its means required two channels in DSSS for this case, that is:

1. Channel allocation for Welsh Powell algorithm:

   - channel 1 for AP_1, AP_3, AP_5, AP_7 and AP_8
   - channel 6 for AP_2, AP_4 and AP_6

   For this case, the Welsh Powell algorithm does not work properly. The AP_3 adjacent with AP_8, and AP_7 adjacent with AP_8, but has same color. This is the weakness of Welsh Powell algorithm in terms of not fulfilling the graph coloring concept; every adjacent vertex must be given a difference color. In this case, Welsh Powell algorithm provides the channel that interference between access points.

2. Channel allocation for Dsatur algorithm and VMA:

   - channel 1 for AP_1, AP_3, AP_5, and AP_7
   - channel 6 for AP_2, AP_4, AP_6, AP_8

   The channels that given to the access point are channel 1 and 6 are non-overlapping channel. In this case, Dsatur algorithm and VMA provides the channel with no interference between access points.

3. The Kuratowski bipartite graph K3,3

The Kuratowski is a graph which has six vertices (Ashay, 2010). Table 4.11 shows an input of the kuratowski bipartite graph K3,3 and adjacent matrix.

Table 4.11: Input of the Kuratowski bipartite graph K3, 3

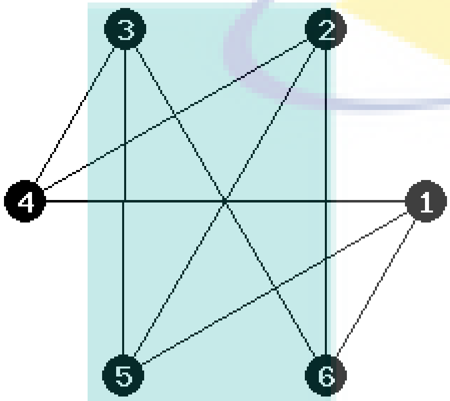
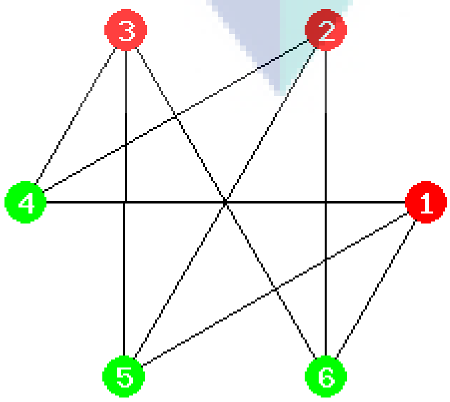
The graph	Adjacent matrix
	0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 1 1 1 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0

Table 4.12 shows output of the Kuratowski bipartite graph K3,3 and number of channel required.

Table 4.12: Output of the Kuratowski bipartite graph K3, 3

Simulator	Output	Number of channel $\chi(G)$
Welsh Powell		$\chi(G) = 2$ C(1,2,3)=channel 1, C(4,5,6)=channel 6.

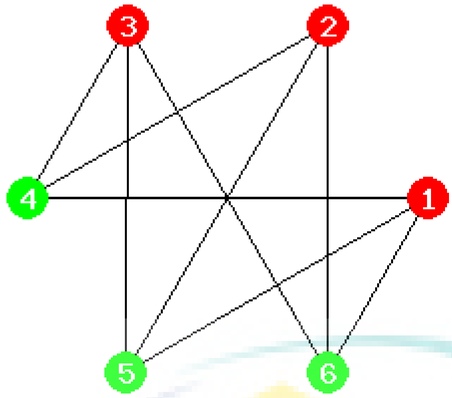
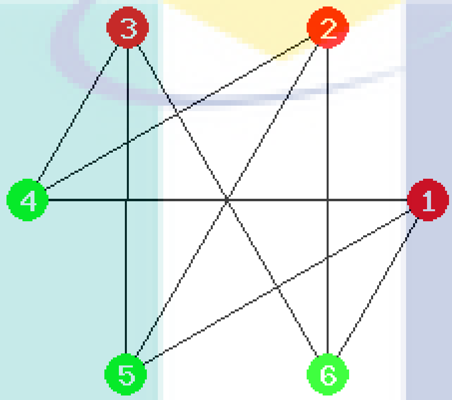
Dsatur		$\chi(G) = 2$ C(1,2,3)=channel 1, C(4,5,6)=channel 6.
VMA		$\chi(G) = 2$ C(1,2,3)=channel 1, C(4,5,6)=channel 6.

Table 4.12 shows that output of the experiment given same result for all algorithms, which are required two colors for coloring the graph. Its means required two channels in DSSS for this case, that is:

- channel 1 for AP_1, AP_2, and AP_3
- channel 6 for AP_4, AP_5 and AP_6

Simulators result the three examples the simple graph, can be shows that the entire simulator to work properly. The number of required channels in each case is the same for all simulators. The channels that given to the access point are channel 1, 6 and 11 are non-overlapping channel. In this case, the entire of algorithm provides the channel with no interference between access points. Figure 4.16 shows the comparison between VMA, Dsatur and Welsh Powell simulator in the simple graph.

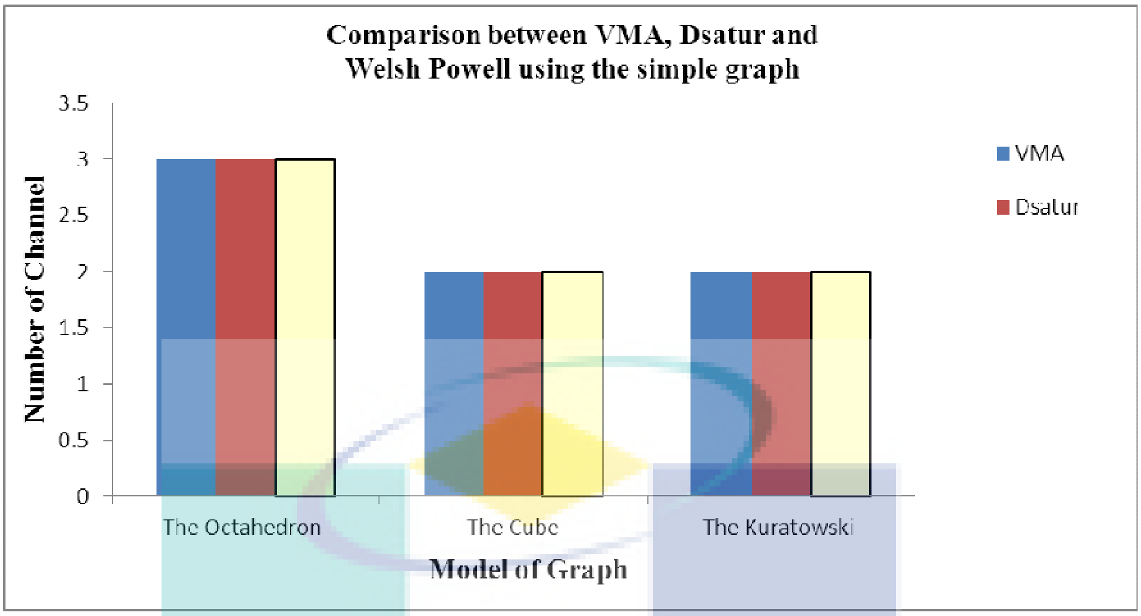


Figure 4.16: Comparison between VMA, Dsatur and Welsh Powell simulator in the simple graph

Figure 4.16 shows that simulation results from the simple graph. The number of required channels is same for all simulators. In the case Octahedron required three channels respectively. Meanwhile, in the Cube and Kuratowski required two channels respectively. The percentage reduction the number of required channel among Welsh Powell algorithm, Dsatur Algorithm and VMA in the simple graph is equivalent to 0.0%.

4.4.2 Case 2: Complex Graph

In this section, three complex graphs will be experiment, which is the complement the Petersen graph, complement the Bondy murty graph and the Ramsey graph K4, 4.

1. **The complement the Petersen graph**

The complement the Petersen graph is a graph which has ten vertices (Ashay, 2010). Table 4.13 shows input the complement the Petersen graph and adjacent matrix.

Table 4.13: Input of the complement the Petersen graph

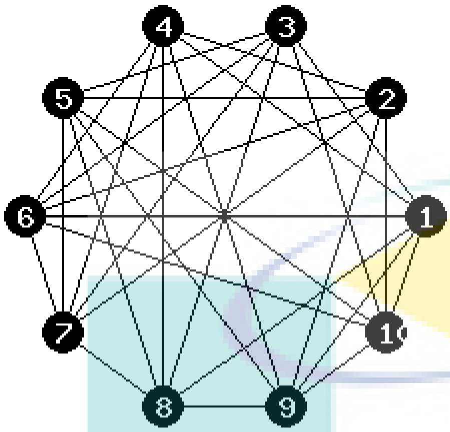
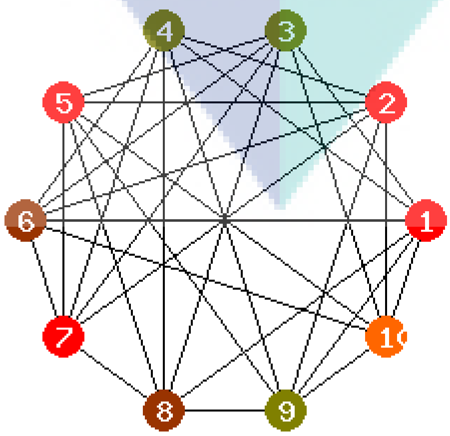
The graph	Adjacent matrix
	0 0 1 1 0 1 0 1 1 1 0 0 0 1 1 1 1 0 1 1 1 0 0 0 1 1 1 1 0 1 1 1 0 0 0 1 1 1 1 0 0 1 1 0 0 0 1 1 1 1 1 1 1 1 0 0 1 0 0 1 0 1 1 1 1 1 0 1 0 0 1 0 1 1 1 0 1 0 1 0 1 1 0 1 1 0 0 1 0 1 1 1 1 0 1 1 0 0 1 0

Table 4.14 shows output of the complement the Petersen graph and number of channel required.

Table 4.14: Output of the complement of Petersen graph

Simulator	**Output**	**Number of channel** ***χ(G)***
Welsh Powell		*χ(G) = 4* *C*(1,2,5,7)= channel 1, *C*(3,4,9)=channel 5, *C*(6,8)=channel 9, *C*(10)=channel 13.

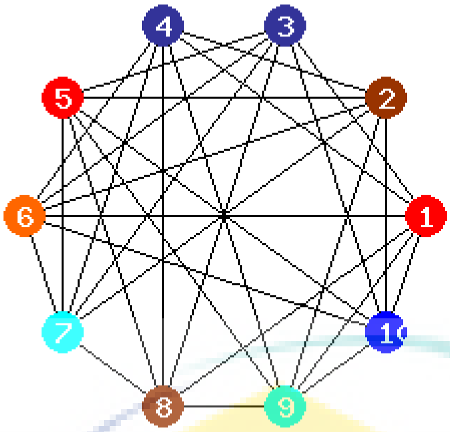
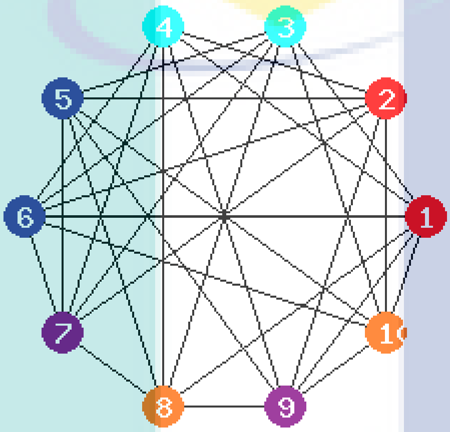
Dsatur		$\chi(G) = 6$ C(1,5)=channel 1, C(7,9)=channel 4, C(3,4)=channel 7, C(2,8)=channel 9, C(10)=channel 11, C(6)=channel 13.
VMA		$\chi(G) = 5$ C(1,2)= channel 1, C(3,4)=channel 4. C(5,6)=channel 7 C(7,9)=channel 10, C(8,10)=channel 13.

Table 4.14 shows that output of the experiment given difference result for all simulators, which are required four colors for Welsh Powell simulator, six colors for Dsatur simulator and five colors for VMA. Its means required four channels in DSSS for Welsh Powell simulator. Meanwhile, for Dsatur simulator is six channels and five channels for VMA, that is:

1. Welsh Powell simulator:

- channel 1 for AP_1, AP_2, AP_5, and AP_7
- channel 5 for AP_3, AP_4 and AP_9
- channel 9 for AP_6 and AP_8
- channel 13 for AP_{10}

For this case, the Welsh Powell simulator does not work properly. AP_2, AP_5 adjacent with AP_7, but has same color. This is the weakness of Welsh Powell algorithm in terms of not fulfilling the graph coloring concept; every adjacent vertex must be given a difference color. In this case, Welsh Powell algorithm provides the channel that interference between access points.

2. Dsatur simulator:

- channel 1 for AP_1 and AP_5
- channel 4 for AP_7 and AP_9
- channel 7 for AP_3 and AP_4
- channel 9 for AP_2 and AP_8
- channel 11 for AP_{10}
- channel 13 for AP_6

The number of required channels using Dsatur is six channels. Dsatur algorithm provides the number of required channels more than VMA.

3. VMA simulator:

- channel 1 for AP_1 and AP_2
- channel 4 for AP_3 and AP_4
- channel 7 for AP_5 and AP_6
- channel 10 for AP_7 and AP_9
- channel 13 for AP_8 and AP_{10}

The number of required channels using VMA is five channels. VMA provides the minimum number of required channel for access points compared with Dsatur algorithm.

2. The complement the Bondy murty graph

The complement the Bondy murty graph is a graph which has fourteen vertices (Ashay, 2010). Table 4.15 shows an input the Bondy murty graph and adjacent matrix.

Table 4.15: Input of the complement the Bondy murty graph

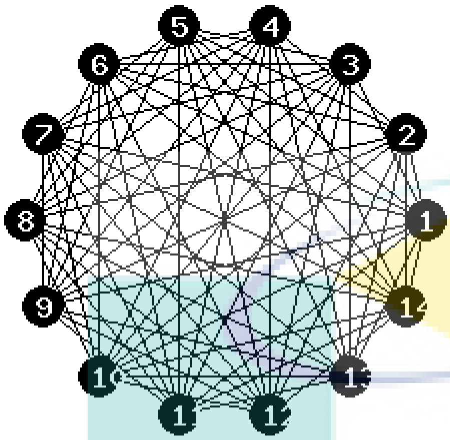
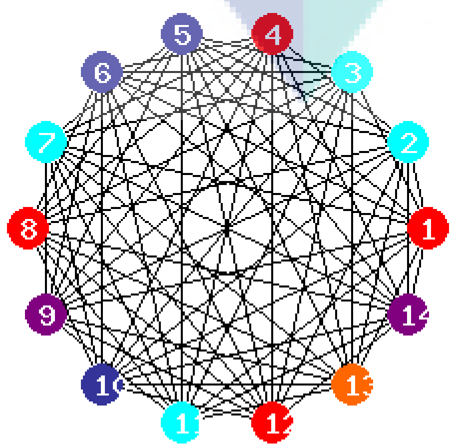
The graph	Adjacent matrix
	0 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 0 1 1 1 0 1 1 1 0 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 0 0 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 0 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 0 1 1 0 1 1 1 0 0 0 1 1 1 1 1 1 0 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 0 1 1 1 1 0 1 1 1 0 0 0 1 1 1 1 0 1 1 1 1 1 1 1 0 0 0 1 1 0 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 0 1 1 1 1 1 0 0 0 1 1 0 1 1 1 1 1 0 1 1 1 0 0

Table 4.16 shows output of the Bondy murty graph and number of channel required.

Table 4.16: Output of the complement the Bondy murty graph

Simulator	**Output**	**Number of channel** ***χ(G)***
Welsh Powell		$\chi(G) = 5$ C(1,4,8,12)=channel 1, C(2,3,7,11)=channel 4, C(5,6,10)=channel 7, C(9,14)=channel 10, C(13)=channel 13.

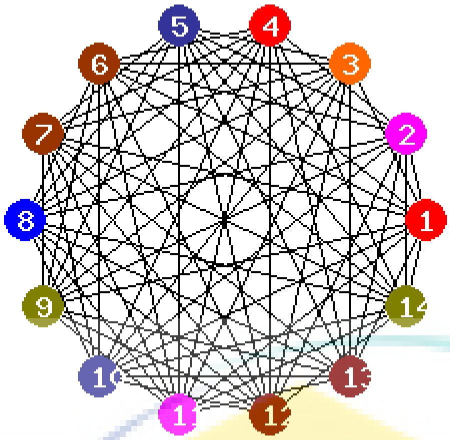
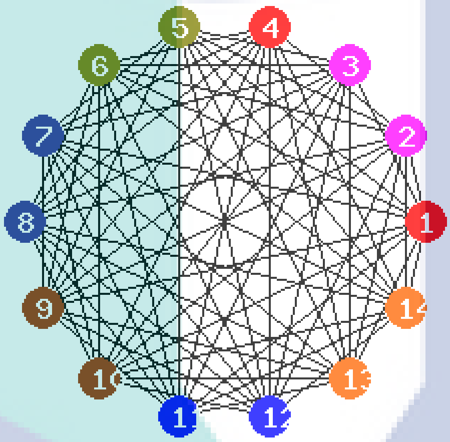
Dsatur		$\chi(G) = 8$ C(1,4)=channel 1, C(2,11)=channel 3, C(9,14)=channel 5, C(5,10)=channel 7, C(6,7)=channel 9, C(8)=channel 11, C(12,13)=channel 12, C(3)=channel 13.
VMA		$\chi(G) = 7$ C(1,4)=channel 1, C(2,3)=channel 3. C(5,6)=channel 5 C(7,8)=channel 7, C(9,10)=channel 9 C(11,12)=channel 11, C(13,14)=channel 13.

Table 4.16 shows that output of the experiment given difference result for all simulators, which are required five colors for Welsh Powell simulator, eight colors for Dsatur simulator and seven colors for VMA. Its means required five channels in DSSS for Welsh Powell simulator. Meanwhile, for Dsatur simulator is eight channels and seven channels for VMA, that is:

1. Welsh Powell simulator:
   - channel 1 for AP_1, AP_4, AP_8, and AP_{12}
   - channel 4 for AP_2, AP_3, AP_7 and AP_{11}
   - channel 7 for $AP_{5,}$ AP_6 and AP_{10}
   - channel 10 for AP_9 and AP_{14}
   - channel 13 for AP_{13}

For this case, the Welsh Powell simulator does not work properly. AP_4, AP_8 adjacent with AP_{12}. AP_3, AP_7 adjacent with AP_{11}. AP_6 adjacent with AP_{10}, but has same color. This is the weakness of Welsh Powell algorithm in terms of not fulfilling the graph coloring concept; every adjacent vertex must be given a difference color. In this case, Welsh Powell algorithm provides the channel that interference between access points.

2. Dsatur simulator:
   - channel 1 for AP_1 and AP_4
   - channel 3 for AP_2 and AP_{11}
   - channel 5 for AP_9 and AP_{14}
   - channel 7 for AP_5 and AP_{10}
   - channel 9 for AP_6 and AP_7
   - channel 11 for AP_8
   - channel 12 for AP_{12} and AP_{13}
   - channel 13 for AP_3

   The number of required channels using Dsatur is eight channels. Dsatur algorithm provides the number of required channels more than VMA.

3. VMA simulator:
   - channel 1 for AP_1 and AP_4
   - channel 3 for AP_2 and AP_3
   - channel 5 for AP_5 and AP_6
   - channel 7 for AP_7 and AP_8
   - channel 9 for AP_9 and AP_{10}
   - channel 11 for AP_{11} and AP_{12}
   - channel 13 for AP_{13} and AP_{14}

   The number of required channels using VMA is seven channels. VMA provides the minimum number of required channel for access points compared with Dsatur algorithm.

3. The Ramsey graph K4, 4

The Ramsey graph K4, 4 is a graph which has seventeen vertices (Ashay, 2010). Table 4.17 shows an input of the Ramsey graph K4, 4 and adjacent matrix.

Table 4.17: Input of the Ramsey graph K4, 4

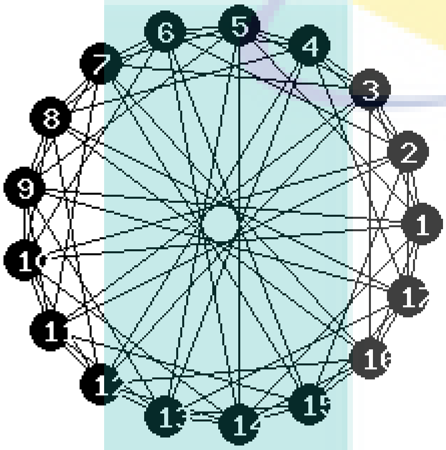
The graph	Adjacent matrix
	0 1 1 0 1 0 0 0 1 1 0 0 0 1 0 1 1 1 0 1 1 0 1 0 0 0 1 1 0 0 0 1 0 1 1 1 0 1 1 0 1 0 0 0 1 1 0 0 0 1 0 0 1 1 0 1 1 0 1 0 0 0 1 1 0 0 0 1 1 0 1 1 0 1 1 0 1 0 0 0 1 1 0 0 0 0 1 0 1 1 0 1 1 0 1 0 0 0 1 1 0 0 0 0 1 0 1 1 0 1 1 0 1 0 0 0 1 1 0 0 0 0 1 0 1 1 0 1 1 0 1 0 0 0 1 1 1 0 0 0 1 0 1 1 0 1 1 0 1 0 0 0 1 1 1 0 0 0 1 0 1 1 0 1 1 0 1 0 0 0 0 1 1 0 0 0 1 0 1 1 0 1 1 0 1 0 0 0 0 1 1 0 0 0 1 0 1 1 0 1 1 0 1 0 0 0 0 1 1 0 0 0 1 0 1 1 0 1 1 0 1 1 0 0 0 1 1 0 0 0 1 0 1 1 0 1 1 0 0 1 0 0 0 1 1 0 0 0 1 0 1 1 0 1 1 1 0 1 0 0 0 1 1 0 0 0 1 0 1 1 0 1 1 1 0 1 0 0 0 1 1 0 0 0 1 0 1 1 0

Table 4.18 shows output of the Ramsey graph K4, 4 and number of channel required.

Table 4.18: Output of the Ramsey graph K4, 4

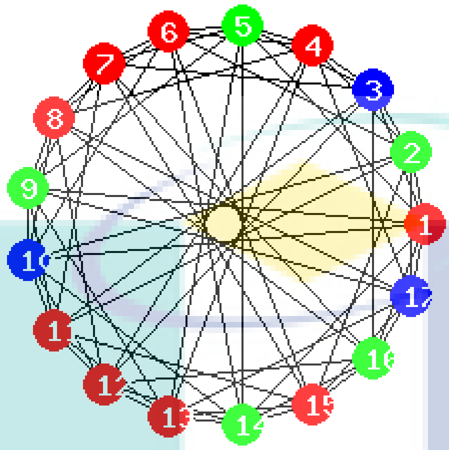
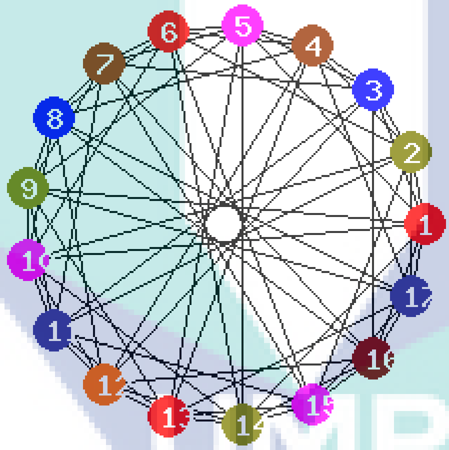
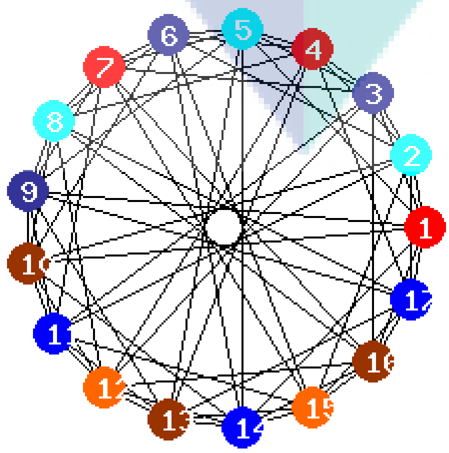
Simulator	**Output**	**Number of channel** $\chi(G)$
Welsh Powell		$\chi(G) = 3$ C(1,4,6,7,8,11,12,13,15)=channel1, C(2,5,9,14,16)=channel 6, C(3, 10,17)=channel 11.
Dsatur		$\chi(G) = 8$ C(1,6,13)=channel 1, C(5,10,15)=channel 3, C(2,9,14)=channel 5, C(11,17)=channel 7, C(4,7)=channel 9, C(3,8)=channel 11, C(16)=channel 12, C(12)=channel 13.
VMA		$\chi(G) = 6$ C(1,4,7)=channel 1, C(2,5,8)=channel 4, C(3,6,9)=channel 7 C(10,13,16)=channel 9, C(11,14,17)=channel 11, C(12,15)=channel 13.

Table 4.18 shows that output the experiment given difference result for all simulators, which are required three colors for Welsh Powell simulator, eight colors for Dsatur simulator and six colors for VMA. Its means required three channels in DSSS for Welsh Powell simulator. Meanwhile, for Dsatur simulator eight channels and six channels for VMA, that is:

1. Welsh Powell simulator:
   - channel 1 for AP_1, AP_4, AP_6, AP_7, AP_8, AP_{11}, AP_{12}, AP_{13} and AP_{15}
   - channel 6 for AP_2, AP_5, AP_9, AP_{14} and AP_{16}
   - channel 11 for AP_3, AP_{10} and AP_{17}

   For this case, the Welsh Powell simulator does not work properly. AP_4, AP_8 adjacent with AP_{12}. AP_6, AP_7 adjacent with AP_{15}. AP_5 adjacent with AP_9, but has same color. This is the weakness of Welsh Powell algorithm in terms of not fulfilling the graph coloring concept; every adjacent vertex must be given a difference color. In this case, Welsh Powell algorithm provides the channel that interference between access points.

2. Dsatur simulator:
   - channel 1 for AP_1, AP_6 and AP_{13}
   - channel 3 for AP_5, AP_{10} and AP_{15}
   - channel 5 for AP_2, AP_9 and AP_{14}
   - channel 7 for AP_{11} and AP_{17}
   - channel 9 for AP_4 and AP_7
   - channel 11 for AP_3 and AP_8
   - channel 12 for AP_{16}
   - channel 13 for AP_{12}

   The number of required channels using Dsatur is eight channels. Dsatur algorithm provides the number of required channels more than VMA.

3. VMA simulator:
   - channel 1 for AP_1, AP_4 and AP_7
   - channel 4 for AP_2, AP_5 and AP_8

- channel 7 for AP_3, AP_6 and AP_9
- channel 9 for AP_{10}, AP_{13} and AP_{16}
- channel 11 for AP_{11}, AP_{14} and AP_{17}
- channel 13 for AP_{12} and AP_{15}

The number of required channels using VMA is six channels. VMA provides the minimum number of required channel for access points compared with Dsatur algorithm.

Simulators result from the three examples the complex graph can be shows that the number of required channels in each case is different for all simulators. In the case complement the Petersen graph, the VMA required five channels. The Dsatur required six channels and Welsh Powell required four channels respectively. In the case complement the Bondy murti graph, the VMA required seven channels. The Dsatur required eight channels and Welsh Powell required five channels respectively. In the case Ramsey graph, the VMA required six channels. The Dsatur required eight channels and Welsh Powell required three channels respectively. In this case, Welsh Powell simulator is not compared in the complex graph since its drawback in terms of not fulfills the graph coloring concept. This is because there is two adjacent vertices have the same color. Figure 4.17 shows the comparison between VMA, and Dsatur simulator.

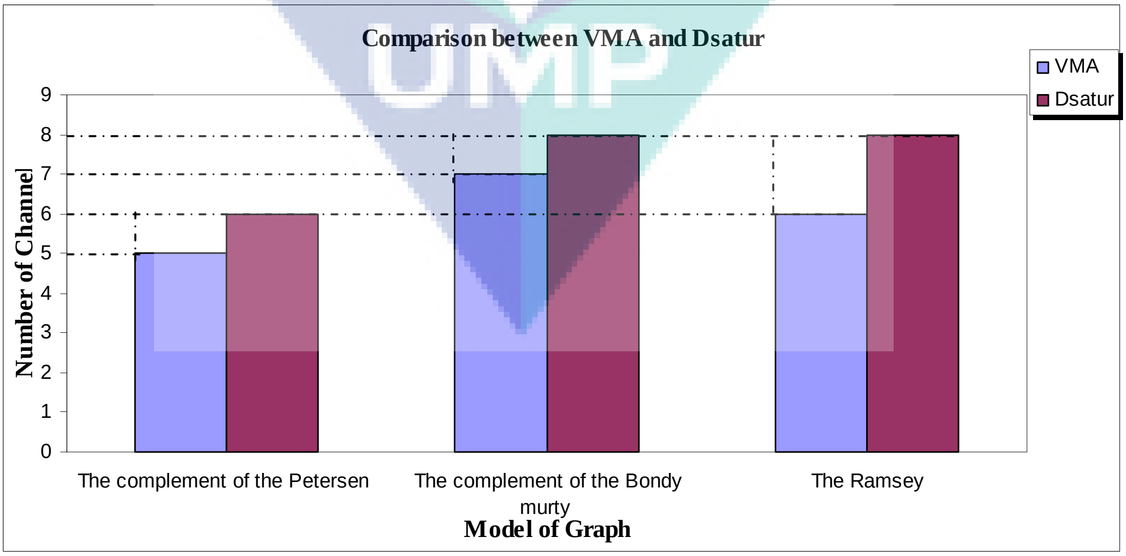


Figure 4.17: Comparison between VMA and Dsatur simulator in the complex graph

Figure 4.17 shows that the average number of required channels from VMA is smaller than Dsatur simulators. Greatest of reduction in the Ramsey graph, that is two channels. Meanwhile, in the complement the Petersen graph and the complement the Bondy murti graph, reduction one channel respectively. The percentage reduction the number of required channel in the complex graph that is 16.7% in the complement the Petersen graph, 12.5% in the complement the Bondy murti graph and 25% in the Ramsey graph respectively. Thus, the total percentage reduction in the number of required channels between the VMA simulator and Dsatur simulator in the complex graph is equivalent to 18.1%.

4.5 CONCLUSION

In this chapter, the simulation of VMA, Dsatur algorithm and Welsh Powell algorithm has been elaborated. A series of an experiment was carried out by using two model graphs that are simple graph and complex graphs. In case of simple graph used three simple graphs that is the Octahedron, the Cube and the Kuratowski bipartite graph K3, 3. The Octahedron has six vertices that show input in Table 4.6. The Cube has eight vertices that show input in Table 4.8. The Kuratowski bipartite graph K3, 3 has six vertices that show input in Table 4.10. In case of complex graph used three complex graphs which is the complement the Petersen graph, the complement the Bondy murty graph and the Ramsey graph K4, 4. The complement the Petersen graph has ten vertices that show input in Table 4.13. The complement the Bondy murty graph has fourteen vertices that shown input in Table 4.15. The Ramsey graph K4, 4 has seventeen vertices that show in Table 4.16. The experiment of VMA simulator, Dsatur simulator and Welsh Powell simulator was done in PHP programming and Eye of GNOME integrated with Ubuntu. PHP is selected because the command-line editing facilities and object oriented capabilities. Meanwhile, an automated Eye of GNOME is used for open and saved image. In this experiment, Linux Ubuntu 9.04 is used as a platform to computer (Ubuntu, 2010). All applications for computer are available to these particular Linux platforms. The applications include *gedit*, *vi* and *vim* editor. The experimental results showed that the proposed algorithm finds the minimum number of channel required in the channel allocation on Direct Sequence Spread Spectrum (DSSS). The average percentage reduction in the number of required channels among the VMA simulator and Dsatur algorithm and Welsh Powell algorithm in the simple graph is equivalent to 0.0%. Meanwhile, between the VMA and Dsatur algorithm in the complex graph is equivalent to 18.1%. However, VMA and Welsh Powell algorithm is not compared in complex graph since its drawback in terms of not fulfill the graph coloring concept. This is because there is two adjacent vertices have the same color. Overall, even there is no reduction for number of required channel among VMA, Dsatur algorithm and Welsh Powell algorithm in the simple graph, but the outstanding significant contribution of VMA since it has reduction in the complex graph.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1. CONCLUSIONS

In this research, the algorithm based on the work by many researchers have been discussed (refer Chapter 2). In particular, a new algorithm called Vertex Merge Algorithm (VMA) is proposed for channel allocation (refer Chapter 3). VMA considers only channel allocation for direct sequence spread spectrum (DSSS). The access point based on the logical structure of vertex in order to a coloring the graph. Each vertex on the graph will be arranged based on decreasing number of degree. The vertex in the first place on the set will be given a color, and then this vertex is merged with non-adjacent vertex. This process will be continued to repeat until all vertices is given color. In the remainder of this thesis, vertex represents of the access point.

In Chapter 4, an analysis of VMA is presented in terms of the algorithm effectiveness and minimum number of required channel. The novelty of this research is new VMA to solve the problem channel allocation in the DSSS. VMA has been same performance compared to Dsatur and Welsh Powell in the cases of the simple graph. However, VMA has better performance compared to Dsatur in the cases of the complex graph. Therefore, this algorithm is particularly useful for the large number of access points. Welsh Powell simulator is not compared in the complex graph since their drawbacks in terms of not fulfill the graph coloring concept. This is because there are two adjacent vertices in the Welsh Powell have the same color. The total percentage reduction in the number of required channels between the VMA simulator and Dsatur simulator the simple graph is equivalent to 0.0%. Meanwhile, in the complex graph is equivalent to 18.1%. Overall, even there is no reduction for number of required channel between VMA, Dsatur and Welsh Powell simulator in the simple graph, but the outstanding significant contribution of VMA since it has reduction in the complex graph.

This research does provide the following novel contributions: The first major contribution of the research is the new algorithm for channel allocation on the Direct Sequence Spread Spectrum (DSSS) called Vertex Merge Algorithm (VMA), which provides the minimum number of required channel compared to Dsatur algorithm. The second contribution is the VMA simulator, Dsatur simulator and Welsh Powell simulator tools that show the performance of the algorithm.

5.2 RECOMMENDATIONS FOR THE FUTURE RESEARCH

Simulation results show the number of channels required in a network can still be minimized. In addition to minimizing the number of channels required in a network, a network administrator must calculate the total of interference between access points in the network (Yoo, 2008). In the future, VMA will take this challenge by handling the calculation of total interference among access points. Furthermore, the performance of the network can be seen from the number of throughput in the network (Li, 2005). In the future, VMA will take this challenge by handling the calculation of the total throughput in the network.

In the network that has the large number of access points; significant a reliable management of channel allocation is required. VMA can help network administrators to determine the allocation of channels automatically. In particular, VMA can be applied in the hardware of the access point. Embeded in the access points will be considered in the future work.

LIST OF PUBLICATIONS

1. Handrizal, Noraziah Ahmad, Ahmed N. Abd. Alla. "*Spread Spectrum Process using Direct Sequence Spread Spectrum (DSSS) and Frequency Hopping Spread Spectrum (FHSS)* ", Proceeding National Conference on NCON-PGR 2009.
2. Handrizal, Noraziah Ahmad, Ahmed N. Abd. Alla. "*Comparison between Direct Sequence Spread Spectrum (DSSS) and Frequency Hopping Spread Spectrum (FHSS)*", Proceeding on International Conference on ICSECS 2009.
3. Handrizal, Noraziah Ahmad, Ahmed N. Abd. Alla. "*Channel Allocation for Direct Sequence Spread Spectrum using Vertex Merge Algorithm*", International Journal of Computer Science & Information Technology (IJCSIT), Vol. 3, No.1, pp. 139-150. 2011.
4. Handrizal, Noraziah Ahmad, Ahmed N. Abd. Alla. "*Comparison between Vertex Merge Algorithm and Dsatur Algorithm*". Journal of Computer Science. Vol. 7, Issue 5, pp. 664-670. 2011

REFERENCES

Ahmad, A. 2003. *Data Communication Principles for Fixed and Wireless Networks*, Kluwer Academic Publisher.

Al Mamun, K.M.A. et al. 2009. An Efficient Variable Channel Allocation Technique for WLAN IEEE 802.11 Standard, *Proceedings of Conference on Circuits, Communication and System*, pp.92-95.

Ashay, D. 2010. *The Vertex Coloring*. http://www.dharwadker.org/vertex_coloring/. Accessed on 20 December 2010.

Bacak, G. 2004. *Vertex Coloring of A Graph*. Master Thesis. Izmir Institute of Technology. Turkey.

Bensky, A. 2008. *Wireless Positioning Technologies and Aplications*, Artech House, Inc.

Bolobas, B. 1998. *Modern Graph Theory*. Springer-Verlag. New York.

Brelaz, D. 1979. New Methods to Color the Vertices of a graph, *Communication of the ACM*, no.**22**, pp.251-256.

Carpenter, T. 2008. *Certified Wireless Network Administrator*, Fourth Edition. Mc-Grow-Hill.

Chen, J. et al. 2008. A Fast Channel Allocation Scheme Using Simulated Annealing in Scalable WLANs. *IEEE International Conference on Broadband Communications, Networks and System*, pp.205-211.

Dandashi, A. and Mouhamed, M. 2010. Graph Coloring for Class Scheduling. *International Conference on Computer Systems and Applications.*

Diestel, R. 2010. *Graph Theory*. Springer-Verlag.

Duan, Z. et al. 2010. Optimal Channel Assignment for Wireless Networks Modelled as Hexagonal and Square Grids. *International Conference on Networks Security Wireless Communications and Trusted Computing.*

Drieberg, M. et al. 2009. Effectiveness of Asynchronous Channel Assignment Scheme in Heterogeneous WLANs. *International Symposium on Wireless Pervasive Computing.*

Eslamnour, B. et al. 2009. Dynamic Channel Allocation in Wireless Networks using Adaptive Learning Automata. *IEEE on Wireless Communications and Networking Conference.*

Gracia, V. et al. 2005. Implementation of A Distributed Dynamic Channel Assignment Mechanism For IEEE 802.11 Networks, *IEEE 16th International*

Symposium on Personal, Indoor and Mobile Radio Communications, pp 1458 – 1462.

Gracia, V. et al. 2005. New Algorithm for Distributed Frequency Assignment in IEEE 802.11 Wireless Networks, *Proceedings of the 11th European Wireless Conference – Next Generation Wireless and Mobile Communications and Services.*

Goldsmith, A. 2005. *Wireless Communications*, Cambridge University Press.

Guizani, M. 2004. *Wireless Communications System and Networks*, Kluwer Academic Publishers.

IEEE Standards Department, 2007. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE standard 802.11.

James, B.C. 2008. *CCNA Wireless Official Exam Certification Guide*, Cisco Press.

Koljanen, J.M. et al. 2010. Distributed Generalized Graph Coloring. *IEEE International Conference on Self-Adaptive and Self-Organizing Systems.*

Koster, A.M.C.A. and X. Mu˜noz, 2010 Graphs and Algorithms in Communication Networks, *Texts in Theoretical Computer Science. An EATCS Series*, DOI 10.1007/978-3-642-02250-0 1, Springer-Verlag Berlin Heidelberg.

Li, Y. et al. 2005, "Impact of Lossy Links on Performance of Multihop Wireless Networks", IEEE, *Proceedings of the 14th International Conference on Computer Communications and Networks*, pp. 303-308.

Mahonen, P. et al. 2004. Automatic Channel Allocation for Small Wireless Local Area Networks Using Graph Colouring Approach. *15th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*. Vol.1, pp.536-539.

Malone, D. et al. 2007. Experimental Implementation of Optimal WLAN Channel Selection without Communication. *2nd IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp.316-319.

Lars, W. et al. 2010. *The Linux System Administrators Guide*. http://www.faqs.org/docs/linux_admin/dir-tree-overview.html, Access on 20 November 2010.

Portable Network Graphic. 2010. http://www.libpng.org/pub/png/. Accessed on 20 December 2010.

Perez, R. 1998. *Wireless Communications Design Handbook Aspects of Noise, Interference, and Environmental Concerns*, Academic Press.

Rackley, S. 2007. *Wireless Networking Technology: From Principles to Successful Implementation*, Elsevier, Inc.

Raj, S. B.E. and Jacob, J.S. 2006. Optimal Channel Allocation in Wireless LAN, *IEEE International on Wireless and Optical Communication Networks.*

Riihijarvi, J. et al. 2005. Frequency allocation for WLANs Using Graph Colouring techniques, *Proceedings of second annual conference of WONS'05.*

Riihijarvi, J. et al. 2006. Performance Evaluation of Automatic Channel Assignment Mechanism for IEEE 802.11 Base on Graph Colouring, *Proceedings The 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Communications*, pp.1-5.

Rohit, G. et al. 2008. A Two Stage Heuristic Algorithm for Solving the Server Consolidation Problem with Item-Item and Bin-Item Incompatibility Constraints. *IEEE International Conference on Services Computing*, pp.39-46.

Sridevi, A and Sumathi, V. 2009. Improved Fault tolerant Model for Channel allocation in wireless Communication. *International Conference on Control, Automation, Communication and Energy Conservation*.

Stallings, W. 2003. *Data and Computer Communications*, Fifth edition. Prentice Hall.

Stavros, D.N. and Leonidas, P. 2006. Recognizing HH-free, HHD-free and Welsh Powell Opposition Graphs. *Discrete Mathematics and Theoretical Computer Science*. Vol. 8. pp 65-82.

The American Dictionaries. 2010. *Simulation*. http://www.answers.com/topic/simulation, Accessed on 8 September 2010.

Theodore, S.R. 2001. *Wireless Communications Principles and Practice*, Second edition, Prentice Hall.

Ubuntu, http://www.ubuntu.com/, Access on 29 February 2010.

Welsh, D.J.A. and Powell, M.B. 1967. An Upper Bound for the Chromatic Number of a Graph and its Application to Timetabling Problems. *The Computer Journal*. **10**: 85-86.

Wikipedia. 2010, *Computer Simulation*. http://en.wikipedia.org/wiki/Simulator#Computer_simulation, Access on 8 September 2010.

Wikipedia, 2010. *RGB color model*. http://en.wikipedia.org/wiki/RGB_color_model, Access on 10 September 2010.

Yoo, K. and Kim, C. 2008. A Channel Management Scheme for Reducing Interference in Ubiquitous Wireless LANs Environment. *International Conference on Multimedia and Ubiquitous Engineering*. pp.276-281.

Yue, X. et al. 2010. A Distributed Channel Assignment Algorithm for Uncoordinated WLANs. *IEEE on Consumer Communications and Networking Conference.*

Yuqing, L. et al. 2010. An Improved Channal Allocation Algorithm Based on List-coloring. *International Conference on Wireless Communications Networking and Mobile Computing*.

Zandstra, M. 2010. *PHP Objects, Pattern, and Practice*. Springer-Verlag. New York.

Zhuang, X. et.al 2010. Channel Assignment in Multi-Radio Wireless Networks Based on PSO Algorithm. *International Conference on Future Information Technology.*