

BAB I PENDAHULUAN

A. Latar Belakang

Dalam perkembangan teknologi saat ini banyak sekali bermunculan macam-macam permainan dalam media komputer. Tidak sedikit orang yang menyukai permainan di komputer. Ada banyak jenis aplikasi permainan, diantaranya permainan pertandingan bola, *game house*, dan lain-lain.

Pertama kalinya mencuat tentang *computer game* adalah pada tahun 1958 oleh seseorang bernama Willian HiginBotham yang mendirikan Brookhaven Nation laboratory's atau NBL's Instrumentation Division. Ketika itu Brookhaven akan mengadakan pameran yang dikunjungi oleh ribuan orang yang akan datang ke gimnasium dan labnya.

Berdasarkan pengalaman dari pameran sebelumnya, para pengunjung tidak terlalu tertarik dengan pameran biasa, kemudian Willian mencetuskan ide untuk membuat sebuah *video tennis game*. *Game* ini menggunakan komputer analog yang dipasangkan dengan satu unit osiloskop.

Seorang matematikawan asal Swiss yang bernama Leonhard Euler disebut-sebut sebagai pencetus teori graf. Teori graf merupakan salah satu cabang ilmu matematika dan banyak permasalahan dalam kehidupan sehari-hari dapat direpresentasikan dengan teori graf.

Graf merupakan diagram yang terdiri dari dua komponen yaitu *vertex* dan *edge*. *Vertex* merupakan titik dari graf, dan *edge* adalah garis dari graf. Berdasarkan salah satu topik dalam teori graf yaitu pewarnaan pada graf, dapat dibuat aplikasi permainan. Pewarnaan dari suatu graf merupakan pemberian warna pada semua *vertex* pada suatu graf sedemikian sehingga dua *vertex* yang saling bertetangga tidak mempunyai warna yang sama.

Dalam penelitian ini n dibuat aplikasi *Game list colouring* menggunakan jenis graf *bipartite* dan graf *caterpillar*. *Game* ini terdiri dari dua pemain yaitu Alice dan Bob. *User* akan berperan sebagai Alice dan sistem pada aplikasi ini akan berperan sebagai Bob.

Dasar dari pembuatan *game list colouring* ini adalah menentukan strategi Bob untuk menghalangi langkah Alice dalam mewarnai seluruh *vertex*. Tetapi tidak menutup kemungkinan untuk membuat Alice memenangkan permainan. Dalam permainan ini yang perlu diperhatikan adalah strategi dalam memilih warna dan *vertex* mana yang akan diwarnai.

B. Rumusan Masalah

Dalam Penelitian ini masalah difokuskan pada salah satu topik dari teori graf yaitu *List Colouring Graf* atau pewarnaan graf untuk mewarnai semua *vertex* yang terhubung dengan garis pada graf.

C. Batasan Masalah

Dalam penelitian ini graf yang digunakan dalam permainan adalah graf *caterpillar* dengan jumlah *vertex* batang maksimal 8, dan graf *bipartite* $K(m,n)$ dengan jumlah $m = n$ maksimal 8.

D. Tujuan

Tujuan dilaksanakan penelitian ini adalah :

- a. Mengimplementasikan teori pewarnaan graf ke dalam bahasa pemrograman Visual Basic.
- b. Memanfaatkan graf *bipartite* dan graf *caterpillar* dalam mengimplementasikan ke dalam permainan pewarnaan graf.
- c. Menentukan beberapa strategi untuk menghalangi kemenangan Alice dalam mewarnai seluruh *vertex* dalam graf.

E. Manfaat

Penelitian ini dilakukan untuk memberikan manfaat nyata dalam hal sebagai berikut :

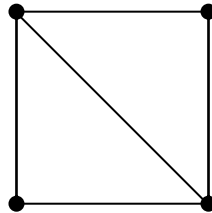
- a. Menambah pengetahuan tentang teori graf dan memanfaatkan dalam pengembangan aplikasi permainan dalam teknologi komputer.
- b. Menambah pengetahuan dan wawasan tentang pewarnaan graf dan mengimplementasikan ke dalam bahasa pemrograman Visual Basic.
- c. Memberikan permainan baru dari pengembangan ilmu matematika yaitu tentang graf.

BAB II

TINJAUAN PUSTAKA

A. Graf

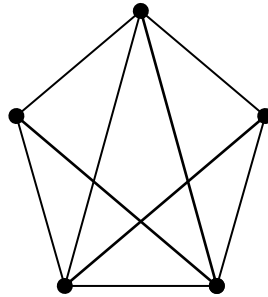
Graf G adalah suatu struktur (V, E) dimana V merupakan himpunan tak kosong dengan elemen-elemennya disebut dengan *vertex* (titik/node), sedangkan E yang mungkin kosong merupakan himpunan pasangan tak urut dari elemen-elemen di V . Anggota dari E disebut *edge* (sisi). (Deo, 1989)



Gambar 2.1. Graf dengan 4 *vertex* dan 5 *edge*

Definisi 2.1 Graf Lengkap (K_n)

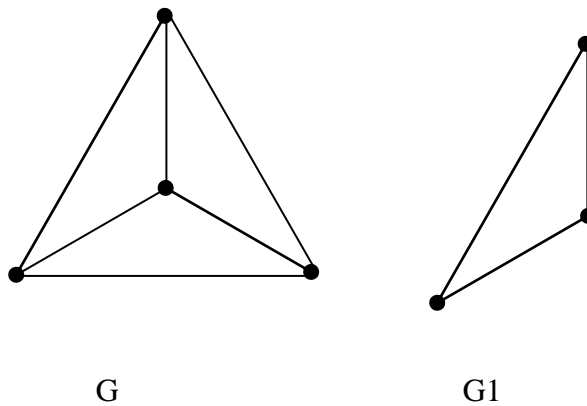
Graf Lengkap adalah graf yang setiap titiknya terhubung dengan semua titik yang lain dengan hanya satu sisi. (Siang, 2004)



Gambar 2.2 Graf lengkap K5

Definisi 2.2. Sub Graf

Jika $G = (V, E)$ adalah suatu graf (berarah atau tak berarah) maka $G_1 = (V_1, E_1)$ disebut subgraf dari G jika $V_1 \neq \emptyset$ dan $E_1 \subseteq E$, $V_1 \subseteq V$, dimana setiap *edge* pada E_1 menempel dengan *vertex* pada V_1 . (Grimaldi, 1999)



Gambar 2.3 G1 adalah sub graf G

Definisi 2.3 Walk

Suatu *walk* dari v ke w adalah barisan titik yang terhubung dan garis secara berselang-seling, diawali dari titik v dan diakhiri pada titik w .

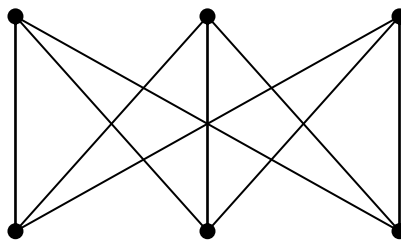
Walk dengan panjang n dari v ke w dituliskan sebagai berikut : $v_0 e_1 v_1 e_2 v_2 \dots v_{n-1} e_n v_n$ dengan $v_0 = v$; $v_n = w$; v_{i-1} dan v_i adalah titik-titik ujung garis e_i . (Wibisono, 2004)

Definisi 2.4 Path

Path dengan panjang n dari v ke w adalah *walk* dari v ke w yang semua garisnya berbeda. *Path* dari v ke w dituliskan sebagai $v = v_0 e_1 v_1 e_2 v_2 \dots v_{n-1} e_n v_n = w$ dengan $e_i \neq e_j$ untuk $i \neq j$. (Wibisono, 2004)

Definisi 2.5 Graf Bipartite

Suatu graf $G = (V,E)$ dikatakan graf *bipartite* jika himpunan *vertex*nya dapat dibagi menjadi dua himpunan V_1 dan V_2 yang *disjoint* ($V_1 \cap V_2 = \emptyset$, $V_1 \cup V_2 = V$), sedemikian sehingga tiap *edge* pada graf tersebut menghubungkan suatu *vertex* di V_1 dan V_2 . (Deo, 1989)

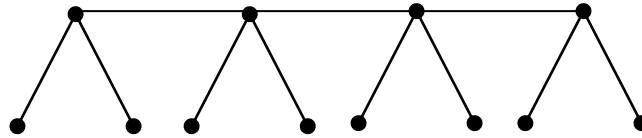


Gambar 2.4 *Bipartite* lengkap

Definisi 2.6 Graf Caterpillar

Caterpillar adalah sebuah *tree* sedemikian sehingga jika semua *vertex* yang berada pada bagian daun dan *edge-edgenya* yang menempel dihilangkan, graf

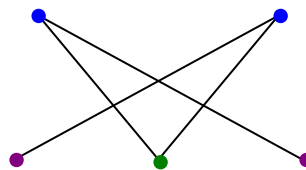
yang tersisa membentuk suatu lintasan. Suatu *tree* adalah *caterpillar* jika dan hanya jika semua *vertex* yang berderajat ≥ 3 dikelilingi paling banyak dua *vertex* yang berderajat dua atau lebih. (Weisstein, 2009)



Gambar 2.5 Graf *Caterpillar*

Definisi 2.7 Pewarnaan *Vertex*

Pewarnaan *vertex* adalah memberi warna pada *vertex* dengan syarat *vertex* yang bertetangga tidak memiliki warna yang sama. (Gross & Yellen, 1999)



Gambar 2.6. Pewarnaan *vertex* graf *bipartite*

Algoritma pewarnaan *vertex* secara berurutan :

L1. Input : Graf G dengan urutan *vertex* V_1, V_2, \dots, V_p

L2. Output : *Vertex* yang sudah diwarnai (f) : $V_G \rightarrow (1, 2, \dots)$

L3. For $i = 1, \dots, p$

$f(V_i) :=$ warna yang tidak digunakan pada *vertex* tetangga v_i .

(Gross & Yellen, 1999)

Definisi 2.8 Derajat *Vertex*

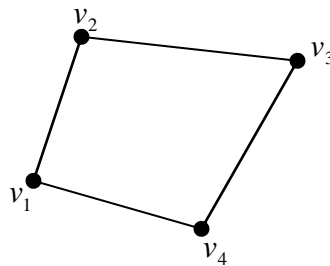
Derajat (*degree*) dari suatu *vertex* v dari graf G adalah jumlah *edge* pada G yang *incident* dengan *vertex* v dan dinotasikan dengan $d(v)$. (Grimaldi, 1999)

Definisi 2.9 *Cycle*

Cycle adalah suatu *walk* yang mempunyai *vertex* awal dan akhir yang sama, dengan tidak terjadi pengulangan *vertex*. (Deo, 1989)

Definisi 2.10 *Adjacent* dan *Incident*

Jika suatu *vertex* v_1 adalah *vertex* ujung dari suatu *edge* e_1 , maka dikatakan *vertex* tersebut *incident* (menempel) dengan *edge* e_1 . Dua *edge* dikatakan *adjacent* (bertetangga) jika *incident* terhadap *vertex* yang sama, atau dua *vertex* dikatakan *adjacent* jika *incident* pada *edge* yang sama. (Deo, 1989)



Gambar 2.7. Graf dengan $V(G) = \{v_1, v_2, v_3, v_4\}$ dan $edge = \{e_1, e_2, e_3, e_4\}$

Definisi 2.11 Graf Sederhana

Parallel edge merupakan dua atau lebih *edge* yang mempunyai *vertex* ujung yang sama, sedangkan *loop* adalah *edge* yang mempunyai *vertex* awal dan *vertex* akhir

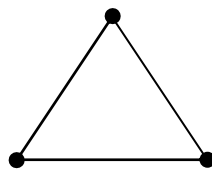
yang sama. Graf sederhana merupakan suatu graf yang tidak mempunyai *parallel edge* dan *loop*. (Wilson & Watkins, 1990)

Definisi 2.12 Graf Terhubung

Suatu graf dikatakan terhubung jika terdapat paling sedikit satu *path* antara setiap pasang *vertex* pada G. Jika sebaliknya, maka G tak terhubung. Graf tak terhubung akan terdiri dari dua atau lebih graf terhubung. (Deo, 1989)

Definisi 2.13 Graf Reguler

Graf reguler (*regular graph*) adalah suatu graf sederhana yang semua *vertex*nya mempunyai derajat yang sama. Jika graf G adalah graf sederhana dengan setiap *vertex*nya berderajat r, maka graf tersebut dinamakan graf reguler berderajat r. Setiap graf lengkap dengan n *vertex* merupakan graf reguler dengan derajat $(n - 1)$. (Weisstein, 2009)



Gambar 2.8. Graf reguler dengan 3 *vertex* dan berderajat 2

B. Permainan (*Game*)

Game erat hubungannya dengan permainan dimana ada aturan tertentu dalam permainan tersebut sehingga ada yang menang dan ada yang kalah, dan biasanya bersifat hiburan. (Dian, 2009)

Definisi 2.14 *Game Komputer*

Game Komputer adalah sebuah program *software* dimana satu atau lebih pemain berusaha untuk membuat keputusan lewat kontrol terhadap *object* dan *resource* guna memenuhi satu tujuan tertentu.

Computer game berbeda dengan jenis *game* yang lain karena tidak ada pergerakan secara fisik atau interaksi langsung dengan objek kecuali lewat perantaraan komputer. (Prayudi, 2008)

Definisi 2.15 *Strategy Game*

Strategy game berfokus pada *gameplay* di mana dibutuhkan pemikiran yang tepat agar dapat meraih kemenangan.

- a. *Real-time strategy and turn based strategy game*. Dalam *real-time* (RTS), *action* dilakukan dalam waktu yang bersamaan oleh masing-masing pihak, sedangkan *turn – based* (TBS), dimana *action* dimainkan per ronde atau bergiliran. Contoh dari genre RTS termasuk *Warcraft Series*. Contoh dari genre TBS termasuk *Heroes of Might and Magic Series*.
- b. *Tactical Game*. Dalam genre ini *player* harus menggunakan macam-macam taktik dan strategi untuk mencapai kemenangan. Contoh dari genre ini yang bersifat RTS termasuk *Warhammer : Dark Omen*. Contoh dari genre ini yang bersifat TBS termasuk *Jagged Alliance Series*.

- c. 4X *game*, yang berarti *eXplore* (penjelajahan), *eXpand* (meluas), *eXploit* (menjajah), dan *eXterminate* (memusnahkan). Contoh dari genre ini termasuk *Galactic Civilizations*, dan *Sid Meier's Alpha Centauri*.
(Anonim, 2008)

C. Metode Waterfall Model

Model pengembangan dengan menggunakan metode waterfall adalah pengembangan sistem atau aplikasi yang dilakukan secara sistematis dan berurutan mulai dari identifikasi masalah sampai tahap *testing*. Metode *waterfall model* muncul pertama kali pada sekitar tahun 1970. Model ini sering disebut dengan “*classic life cycle*”.

BAB III

METODOLOGI PENELITIAN

A. Tempat dan Waktu Penelitian

Penelitian ini dilakukan di Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung. Waktu penelitian dilakukan selama semester genap tahun ajaran 2009-2010.

B. Bahan dan Alat

Bahan yang dibutuhkan untuk melaksanakan penelitian ini adalah teori-teori pendukung tentang graf dan pewarnaan graf dalam mengimplementasikan ke dalam bahasa pemrograman Visual Basic.

Untuk melakukan penelitian ini menggunakan alat berupa perangkat keras (*hardware*) dan perangkat lunak (*software*).

1. Perangkat Keras (*hardware*), dengan spesifikasi :

- a. *Processor* Intel Pentium IV
- b. *RAM* 256 MB
- c. *Hardisk* 40 GB

- d. Monitor dan VGA
- e. *Keyboard* dan *Mouse*

2. Perangkat Lunak (*software*) yang digunakan :

- a. Sistem Operasi : Windows XP SP1
- b. Bahasa Pemrograman : Visual Basic 6
- c. Aplikasi Tool : Microsoft Visual Basic, Adobe Photo Shop

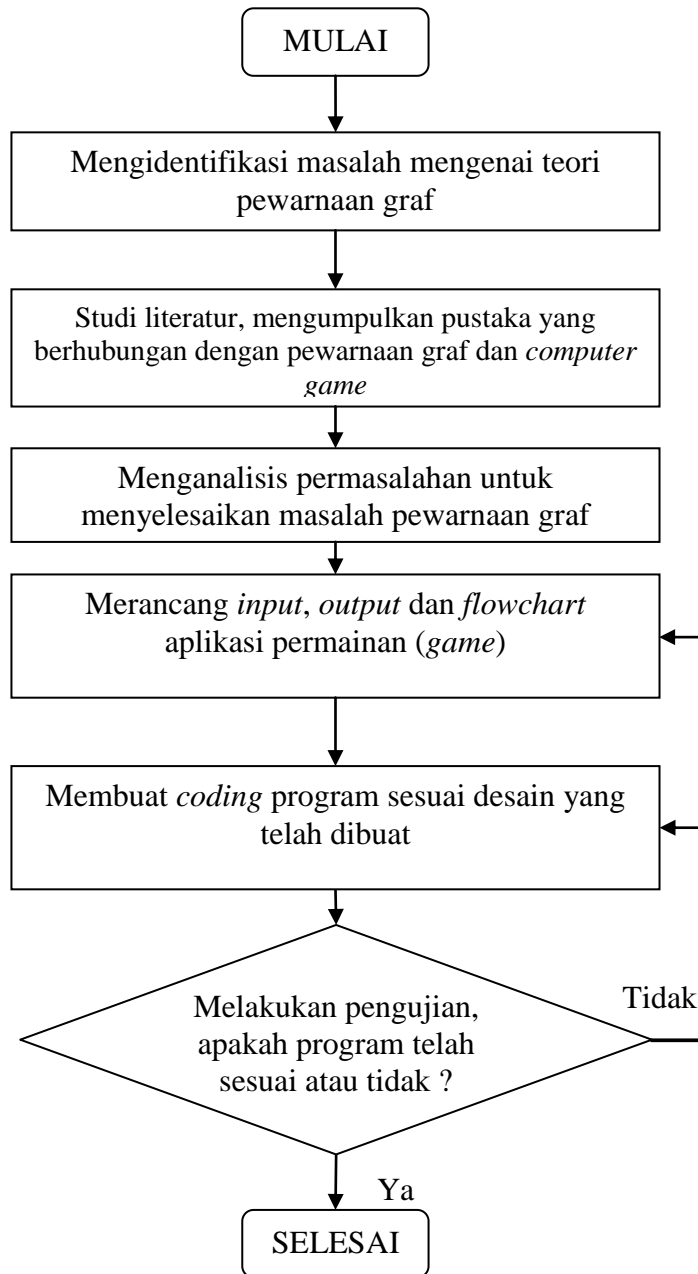
C. Tahapan Penelitian

Dalam pelaksanaan penelitian ini dilakukan beberapa tahapan penelitian sesuai dengan metode pengembangan aplikasi yang digunakan yaitu menggunakan metode *waterfall model* . Metode ini dimulai dengan mengidentifikasi masalah mengenai permainan pewarnaan graf, mengumpulkan studi literatur yaitu mengumpulkan informasi tentang istilah *game* atau permainan, teori graf, dan permainan pewarnaan graf.

Pada tahapan selanjutnya adalah menganalisis permasalahan dan membuat algoritma dari permainan pewarnaan graf . Setelah tahapan analisis selesai dilakukan, maka dilanjutkan dengan merancang desain *input*, *output* , *flowchart* dari permainan pewarnaan graf.

Tahapan berikutnya adalah tahapan implementasi yaitu membuat *coding* program sesuai dengan desain yang telah dirancanga pada tahap sebelumnya. Tahapan terakhir dari penelitian ini adalah melakukan *testing* atau pengujian dari aplikasi yang telah dibuat.

Tahapan pengujian ini dilakukan untuk memastikan apakah program yang dibuat sudah sesuai dengan fungsi kerja yang diharapkan. Jika program belum sesuai dengan fungsi kerja yang diharapkan, maka dilakuka perbaikan program sehingga program dapat berjalan sesuai dengan fungsi kerja yang diharapkan.



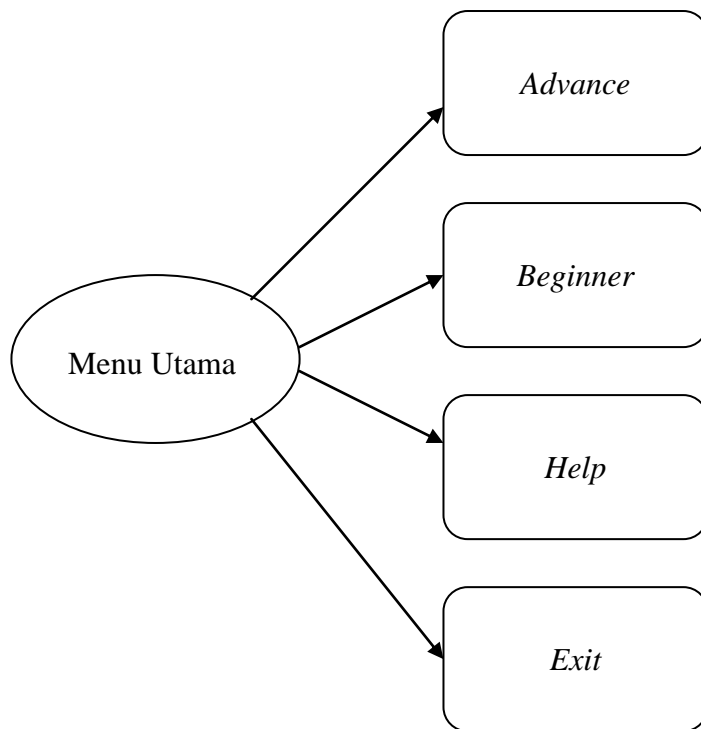
Gambar 3.1 *Flowchart* penelitian

D. Aplikasi *Game List Colouring*

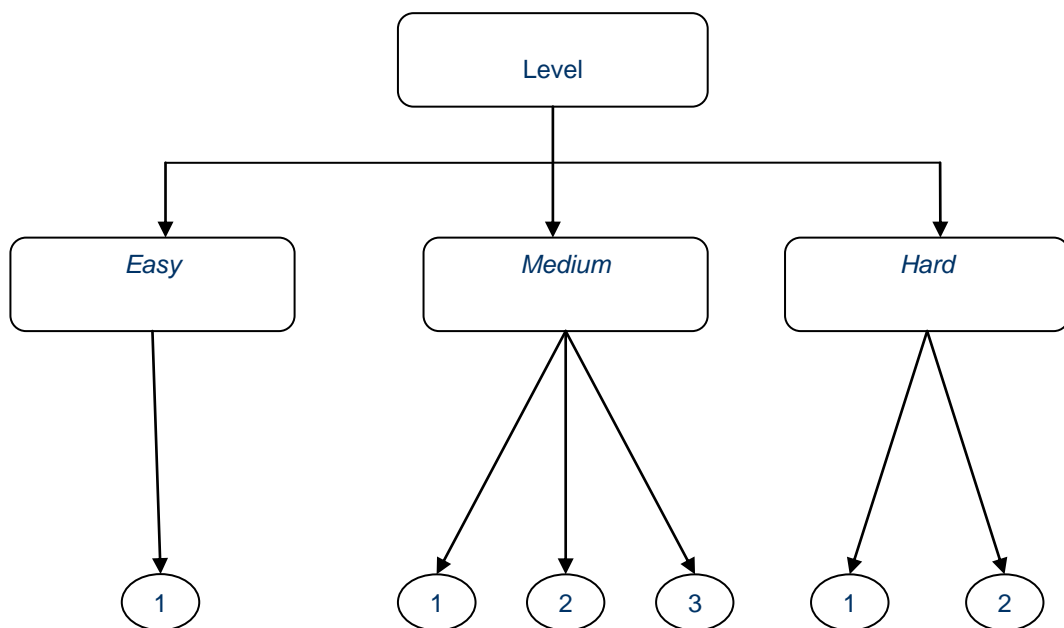
Aplikasi *game list colouring* ini dilakukan oleh seorang *user* yang bermain dengan komputer. *User* bermain sebagai Alice dan komputer akan bermain sebagai Bob. Pada permainan ini yang memulai permainan adalah *user* atau Alice dan dilanjutkan dengan Bob kembali.

Permainan list coloring ini dibagi menjadi dua jenis yaitu *advance* dan *beginner*. Perbedaan permainan *advance* dan *beginner* adalah dari pembatasan waktu bermain. Dalam permainan *advance* user diberi batas waktu dalam mewarnai *vertex*, sedangkan pada permainan *beginner* user tidak diberikan batas waktu dalam melakukan permainan.

Dalam permainan ini dibagi menjadi tiga level dalam permainan *advance* dan *beginner*, yaitu level *easy*, level *medium* dan level *hard*. Setiap level memiliki tingkat kesulitan yang berbeda-beda, dan semakin tinggi level permainan, akan semakin sulit Alice atau *user* untuk memenangkan permainan.



Gambar 3.2 Diagram menu utama game

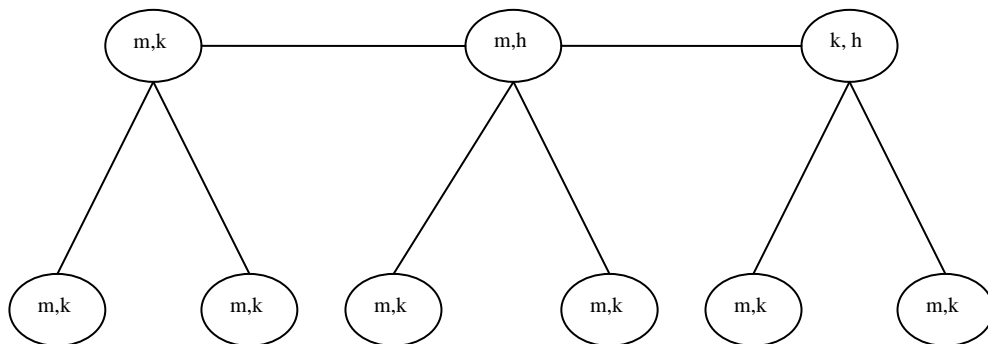


Gambar 3.3 Diagram tingkatan level

a. Level Easy

Pada level *easy* graf yang digunakan adalah graf *caterpillar*, dengan jumlah *vertex* batang (m) minimal 2 dan maksimal 3 ($2 \leq m \leq 3$), yang terdiri 2 pilihan warna dari 3 warna yaitu merah, kuning, dan hijau.

Aturan Pewarnaan dari level *easy* adalah untuk *vertex* batang hanya boleh diwarnai dengan pilihan 2 warna dari 3 warna secara berurutan yaitu (merah, kuning), (merah, hijau), dan (kuning, hijau). Sedangkan untuk *vertex* daun hanya ada dua pilihan warna yaitu merah dan kuning.



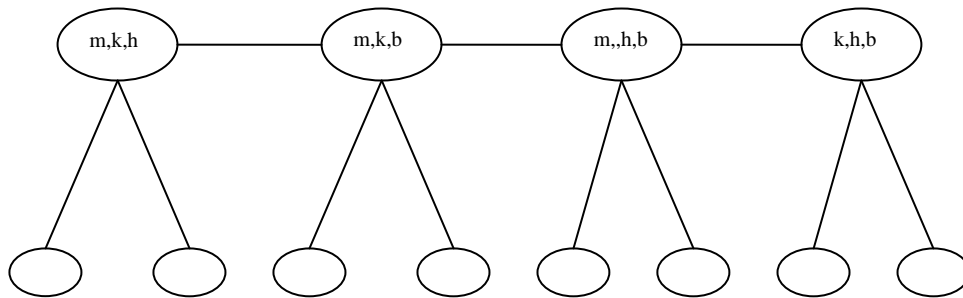
Gambar 3.4 Graf *caterpillar* dengan $m = 3$

b. Level Medium

Selanjutnya pada level *medium* graf yang digunakan masih sama dengan level *easy*, yaitu graf *caterpillar*. Pada level ini jumlah simpul batang adalah $4 \leq m \leq 8$. Ada dua tingkatan pada level *medium*, yaitu level *medium* tingkat 1 dan level

medium tingkat 2. Pada level medium tingkat 1 aturan permainan masih sama dengan level *easy*, hanya pada level ini akan ditambahkan strategi pada *source code* untuk mengalahkan Alice atau *user*.

Level *medium* tingkat 2 memiliki aturan yang sedikit berbeda dengan level sebelumnya. Ada 3 pilihan warna dari 4 warna untuk aturan mewarnai simpul batang, sedangkan untuk mewarnai simpul daun diberikan aturan 1 pilihan warna dari 2 warna. Pada simpul batang, warna yang digunakan yaitu merah, kuning, hijau, dan biru. Aturan pilihan warnanya adalah (merah, kuning, hijau), (merah, kuning, biru), (merah, hijau, biru), dan (kuning, hijau, biru).



Gambar 3.5 Graf caterpillar dengan $m = 4$

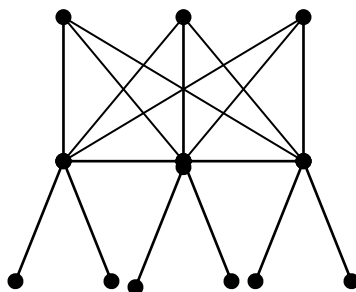
Untuk pewarnaan pada simpul daun diberikan aturan hanya boleh mewarnai dengan warna merah untuk simpul daun yang berindeks genap, dan warna kuning untuk simpul daun yang berindeks ganjil.

Pada level *medium* tingkat 3 aturan yang digunakan adalah *vertex* batang hanya boleh diwarnai dengan warna biru atau hijau, sedangkan untuk *vertex* daun aturan yang digunakan seperti pada level *medium* tingkat 2.

c. Level *Hard*

Level *hard* adalah level terakhir pada permainan. Level ini menggunakan graf *bipartite* dan graf gabungan antara *bipartite* dan *caterpillar*. Pada level *hard* tingkat 1 graf yang digunakan adalah graf *bipartite* lengkap dengan jumlah $m = n$ dan maksimal m adalah 8. Sedangkan pada level *hard* tingkat 2 graf yang digunakan adalah gabungan dari graf *bipartite* dan graf *caterpillar*.

Aturan permainan pada level *hard* tingkat 1 adalah semua *vertex* bebas diwarnai dengan warna merah, kuning, atau hijau. Pada level *hard* tingkat 2 aturan permainan tidak jauh berbeda dengan tingkat sebelumnya yaitu pada *vertex* (m, n) bebas diwarnai dengan warna merah, kuning, atau hijau. Sedangkan pada *vertex* daun hanya boleh diwarnai dengan warna merah atau kuning.



Gambar 3.6 Gabungan graf *bipartite* dan graf *caterpillar*