

**PENERAPAN ALGORITMA RUNUT BALIK (*BACKTRACKING*)  
DALAM PEWARNAAN GRAF (*GRAPH COLORING*)  
PADA PENYELESAIAN PERMAINAN *SUDOKU***

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
Mencapai derajat Sarjana S-1

Program Studi Matematika



Diajukan oleh:

**Arum Septya Ayu  
12610015**

Kepada  
**PROGRAM STUDI MATEMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA  
YOGYAKARTA  
2016**



**PENGESAHAN SKRIPSI/TUGAS AKHIR**

Nomor : B- 4331/Un.02/DST/PP.05.3/12/2016

Skripsi/Tugas Akhir dengan judul : Penerapan Algoritma Runut Balik (*Backtracking*) dalam Pewarnaan Graf (*Graph Coloring*) pada Penyelesaian Permainan *Sudoku*

Yang dipersiapkan dan disusun oleh :  
Nama : Arum Septya Ayu  
NIM : 12610015  
Telah dimunaqasyahkan pada : 9 November 2016  
Nilai Munaqasyah : A

Dan dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga

**TIM MUNAQASYAH :**

Ketua Sidang

Noor Saif Muh. Mussafi, M.Sc  
NIP. 19820617 200912 1 005

Penguji I

Much. Abrori, S.Si, M.Kom  
NIP.19720423 199903 1 003

Penguji II

Dr. Muhammad Wakhid Musthofa, M.Si  
NIP.19800402 200501 1 003

Yogyakarta, 2 Desember 2016  
UIN Sunan Kalijaga  
Fakultas Sains dan Teknologi  
Dekan



Dr. Murtopo, M.Si  
NIP.19691212 200003 1 001



## **SURAT PERSETUJUAN SKRIPSI/TUGAS AKHIR**

Hal : Persetujuan Skripsi

Lamp : 3 Eksemplar Skripsi

Kepada

Yth. Dekan Fakultas Sains dan Teknologi

UIN Sunan Kalijaga Yogyakarta

di Yogyakarta

*Assalamu'alaikum wr. wb.*

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka kami selaku pembimbing berpendapat bahwa skripsi Saudara:

Nama : Arum Septya Ayu

NIM : 12610015

Judul Skripsi : Penerapan Algoritma Runut Balik (*Backtracking*) dalam Pewarnaan Graf (*Graph Coloring*) pada Penyelesaian Permainan *Sudoku*

sudah dapat diajukan kembali kepada Program Studi Matematika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam bidang Matematika.

Dengan ini kami berharap agar skripsi/tugas akhir Saudara tersebut di atas dapat segera dimunaqsyahkan. Atas perhatiannya kami ucapkan terma kasih.

*Wassalamu'alaikum wr. wb.*

Yogyakarta, 25 Oktober 2016

Pembimbing

Noor Saif Muhammad Mussafi, M.Sc.

NIP. 19820617 200912 1 005

## SURAT PERNYATAAN KEASLIAN SKRIPSI

Yang bertanda tangan dibawah ini saya:

Nama : Arum Septya Ayu  
NIM : 12610015  
Prodi / Smt : Matematika / IX  
Fakultas : Sains dan Teknologi

Dengan ini saya menyatakan bahwa skripsi ini tidak terdapat karya serupa yang diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi lain, dan sepanjang pengetahuan saya juga belum terdapat karya yang pernah ditulis atau diterbitkan orang lain, kecuali yang secara tertulis dalam naskah ini dan disebutkan dalam daftar pustaka.

Yogyakarta, 27 Oktober 2016

Yang menyatakan,



Arum Septya Ayu

NIM: 12610015

## HALAMAN PERSEMBAHAN

*Karya ini saya persembahkan untuk:*

Papa, Mama, Kakak dan Adikku tercinta

Keluarga Besar ku

Sahabat-sahabatku

*Teman-teman Matematika 2012*

Beserta Almamater tercinta

Program Studi Matematika

Fakultas Sains dan Teknologi

UIN Sunan Kalijaga Yogyakarta

## HALAMAN MOTTO

*“Karena sesungguhnya sesudah kesulitan itu ada kemudahan. Sesungguhnya sesudah kesulitan itu ada kemudahan. Maka apabila kamu telah selesai (dari sesuatu urusan), kerjakanlah dengan sungguh-sungguh (urusan) yang lain. Dan hanya kepada Tuhan-mulah kamu berarap (QS. Al-Insyirah: 5-8)”*

*“Man Shabara Zhafira”*

*Barangsiapa yang bersabar akan beruntung*

## KATA PENGANTAR

*Assalamu'alaikum Wr. Wb.*

*Alhamdulillah* aahirabbil'aalamiin, puji syukur kepada Allah yang telah melimpahkan rahmat, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan penulisan skripsi yang berjudul “*Penerapan Algoritma Runut Balik (Backtracking) dalam Pewarnaan Graf (Graph Coloring) pada Penyelesaian Permainan Sudoku*” guna memenuhi syarat memperoleh gelar kesarjanaan di Program Studi Matematika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.

Shalawat serta salam senantiasa tercurahkan kepada Baginda Rasulullah Muhammad SAW, yang selalu menjadi suri tauladan bagi seluruh umat manusia dari zaman kegelapan menuju zaman terang seperti saat ini. Penulis menyadari bahwa proses penulisan skripsi ini tidak terlepas dari bantuan, dukungan, motivasi, dan bimbingan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terimakasih kepada:

1. Papa Triyono Supriaji dan Mama Radiyah tercinta, terimakasih atas segala doa, dukungan moril maupun materil dan motivasi sampai saat ini sehingga penulis selalu bersemangat untuk menjalani perkuliahan hingga penulisan skripsi dapat terselesaikan, serta ketiga Saudariku Mba Kiki, Rema, dan Ghina tersayang yang selalu membuat rindu untuk pulang ke rumah.

2. Alm. Mbah Puryatin dan istri, serta seluruh keluarga besar Sinemet dan Wirameja. Terimakasih selalu memberikan doa, kasih sayang dan perhatian, sehingga penulisan skripsi ini dapat terselesaikan.
3. Prof. Drs. Yudian Wahyudi, MA, Ph.D., selaku Rektor UIN Sunan Kalijaga Yogyakarta.
4. Dr. Murtono, M.Si., selaku Dekan Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
5. Bapak Dr. Muhammad Wakhid Musthofa, M.Si., selaku Ketua Program Studi Matematika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
6. Bapak Muchammad Abrori, M.Kom., selaku Dosen Penasihat Akademik mahasiswa Program Studi Matematika angkatan 2012 atas segala arahan dan bimbingan selama masa studi.
7. Bapak Noor Saif Muhammad Mussafi, M.Sc., selaku dosen pembimbing skripsi. Terimakasih selalu meluangkan waktu untuk membimbing, serta memberikan pengarahan sehingga skripsi ini dapat terselesaikan dengan baik.
8. Bapak/Ibu Dosen dan Staf Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta. Terimakasih atas ilmu, bimbingan dan pelayanan selama masa perkuliahan dan penyusunan skripsi.
9. Abang Sepupu Herjuno Aji Prayogo, S.Kom. Terimakasih telah banyak membantu dan memberikan motivasi dalam menyelesaikan skripsi ini.
10. Teman hidup, Ibnu Chaedir, S.S.. Terimakasih atas segala doa, dukungan dan semangat tiada henti kepada penulis untuk segera menyelesaikan penulisan skripsi.



11. Bulek Eti dan Mas Aji. Terimakasih telah membantu sejak awal di Jogja hingga saat penulis dapat menyelesaikan skripsi ini.
12. Sahabatku MARMOS (Malaikhah, Diani, Lisa dan Rindu). Terimakasih telah membuat hari ku lebih berwarna. Semoga persahabatan ini tetap ada sampai tua walaupun terpisah jarak nantinya.
13. Teman-teman matematika angkatan 2012. Terimakasih atas kebersamaan dan cerita yang telah kita buat bersama. Serta seluruh keluarga HM-PS Matematika. Terimakasih telah mengajarkan banyak hal tentang organisasi dan pengalaman yang penulis dapatkan.
14. In Imanatun, Nisa, Bu Nur, Duwi, Fatimah, Daim, Miftah, Iin, Hana, Nurul, Muthmainnah, Fidia, Juni dan teman lainnya. Terimakasih atas waktu bermain dan belajar bersama kalian.
15. Teman-teman KKN 164 Pejaten I (Rona, Endah, Arif, Joko, Galih, Na'im, Ifti, Win dan Rizki). Keluarga baru selama dua bulan KKN dan semoga seterusnya. Terimakasih selalmu memberikan semangat dan motivasi mengerjakan skripsi. Terimakasih juga untuk Bapak dan Ibu Tulus selaku orangtua baru, beserta masyarakat Pejaten I yang telah memberikan kesempatan kepada penulis untuk mempelajari apa yang penulis belum dapatkan sebelumnya dan kebersamaan yang tak mudah dilupakan.
16. Semua pihak yang tidak dapat penulis sebutkan satu per satu. Terimakasih telah membantu dan memberikan dukungan serta doa sehingga penulis dapat menyelesaikan skripsi ini.

Semoga Allah SWt memberikan balasan kepada mereka dengan kebaikan. Penulis menyadari bahwa penulisan skripsi ini masih jauh dari sempurna, sehingga diharapkan kritik dan saran yang bersifat membangun demi kesempurnaan skripsi ini. Semoga skripsi ini dapat memberikan manfaat untuk penulis dan pembaca.

*Wassalamu'alaikum Wr.Wb*

Yogyakarta, 27 Oktober 2016

Arum Septya Ayu

## DAFTAR ISI

<b>HALAMAN JUDUL</b> .....	<b>i</b>
<b>HALAMAN PENGESAHAN SKRIPSI</b> .....	<b>ii</b>
<b>HALAMAN PERSETUJUAN SKRIPSI</b> .....	<b>iii</b>
<b>HALAMAN PERNYATAAN KEASLIAN SKRIPSI</b> .....	<b>iv</b>
<b>HALAMAN PERSEMBAHAN</b> .....	<b>v</b>
<b>HALAMAN MOTTO</b> .....	<b>vi</b>
<b>KATA PENGANTAR</b> .....	<b>vii</b>
<b>DAFTAR ISI</b> .....	<b>xi</b>
<b>DAFTAR GAMBAR</b> .....	<b>xiv</b>
<b>DAFTAR TABEL</b> .....	<b>xix</b>
<b>DAFTAR LAMPIRAN</b> .....	<b>xxi</b>
<b>DAFTAR LAMBANG</b> .....	<b>xxii</b>
<b>ABSTRAK</b> .....	<b>xxiii</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	5
1.3 Tujuan Penelitian .....	5
1.4 Batasan Masalah .....	5
1.5 Manfaat Penelitian .....	6
1.6 Tinjauan Pustaka .....	6
1.7 Metode Penelitian .....	9

1.8 Sistematika Penulisan .....	11
<b>BAB II LANDASAN TEORI .....</b>	<b>12</b>
2.1 Teori Graf .....	12
2.1.1 Definisi .....	12
2.1.2 Terminologi Dasar .....	13
2.1.3 Jenis-jenis Graf .....	16
2.1.4 Pewarnaan Graf ( <i>Graph Coloring</i> ) .....	20
2.1.5 Pohon ( <i>Tree</i> ) .....	28
2.2 Algoritma Runut Balik ( <i>Backtracking</i> ) .....	30
2.3 Permainan Sudoku .....	31
2.3.1 Sejarah Permainan Sudoku .....	31
2.3.2 Aturan Permainan Sudoku .....	34
2.3.3 Graf <i>Sudoku</i> .....	35
2.4 Visual Basic.NET .....	37
<b>BAB III PEMBAHASAN .....</b>	<b>40</b>
3.1 Representasi Permainan Sudoku ke dalam Graf .....	40
3.1.1 Cara Kerja Representasi Permainan <i>Sudoku</i> ke dalam Graf .....	42
3.1.2 Pewarnaan Graf <i>Sudoku</i> $X_n$ .....	43
3.2 Konsep dan Cara Kerja Algoritma <i>Backtracking</i> .....	45
3.2.1 Konsep Algoritma <i>Backtracking</i> .....	45
3.2.2 Cara Kerja Algoritma <i>Backtracking</i> .....	48
3.3 Penerapan Algoritma <i>Backtracking</i> dalam Pewarnaan Graf pada Penyelesaian Permainan Sudoku .....	49

3.3.1	Pencarian Solusi Permainan <i>Sudoku</i> .....	54
a.	Solusi <i>Subgrid</i> Ke-1 ( $S_1$ ) .....	54
b.	Solusi <i>Subgrid</i> Ke-2 ( $S_2$ ) .....	56
c.	Solusi <i>Subgrid</i> Ke-3 ( $S_3$ ) .....	59
d.	Solusi <i>Subgrid</i> Ke-4 ( $S_4$ ) .....	62
e.	Solusi <i>Subgrid</i> Ke-5 ( $S_5$ ) .....	65
f.	Solusi <i>Subgrid</i> Ke-6 ( $S_6$ ) .....	73
g.	Solusi <i>Subgrid</i> Ke-7 ( $S_7$ ) .....	85
h.	Solusi <i>Subgrid</i> Ke-8 ( $S_8$ ) .....	88
i.	Solusi <i>Subgrid</i> Ke-9 ( $S_9$ ) .....	90
3.3.2	Rancang Bangun Penyelesaian Permainan <i>Sudoku</i> dengan Algoritma <i>Backtracking</i> .....	144
<b>BAB IV PENUTUP</b> .....		163
4.1	Kesimpulan .....	163
4.2	Saran .....	165
<b>DAFTAR PUSTAKA</b> .....		166
<b>LAMPIRAN</b> .....		168

## DAFTAR GAMBAR

<b>Gambar 1.1</b> Skema Langkah Penelitian .....	10
<b>Gambar 2.1</b> Graf $G$ .....	12
<b>Gambar 2.2</b> Graf $G_1$ dan $G_2$ .....	13
<b>Gambar 2.3</b> Tiga buah graf .....	17
<b>Gambar 2.4</b> Graf Berarah $G$ .....	18
<b>Gambar 2.5</b> Graf Lengkap .....	19
<b>Gambar 2.6</b> Graf Isomorfik .....	20
<b>Gambar 2.7</b> Ilustrasi Pembuktian Teorema 2.1 .....	22
<b>Gambar 2.8</b> Pewarnaan graf $G$ .....	26
<b>Gambar 2.9</b> Pohon Berakar .....	29
<b>Gambar 2.10</b> <i>Pre-sudoku</i> dari <i>La France</i> .....	32
<b>Gambar 2.11</b> Ilustrasi Ketetanggaan Simpul Graf <i>Sudoku</i> .....	36
<b>Gambar 2.12</b> Tampilan Awal <i>Form Design</i> .....	39
<b>Gambar 2.13</b> Tampilan Awal <i>Window Code</i> .....	39
<b>Gambar 3.1</b> <i>Flowchart</i> Representasi Permainan <i>Sudoku</i> ke Graf .....	42
<b>Gambar 3.2</b> Contoh Pohon Ruang Status Algoritma <i>Backtracking</i> .....	46
<b>Gambar 3.3</b> <i>Sudoku</i> $9 \times 9$ .....	49
<b>Gambar 3.4</b> Tabel Indeks <i>Sudoku</i> dan Matriks .....	50
<b>Gambar 3.5</b> Graf <i>Sudoku</i> dengan <i>Rank 3</i> ( $X_3$ ) .....	52
<b>Gambar 3.6</b> Permasalahan <i>Sudoku</i> (Jussien, 2007) .....	52
<b>Gambar 3.7</b> Pohon Solusi $S_1$ .....	55
<b>Gambar 3.8a</b> Proses <i>Backtracking</i> $S_2$ Simpul $V_{1,4}$ .....	57

<b>Gambar 3.8b</b> Proses <i>Backtracking</i> $S_2$ Simpul $V_{1,4}$ .....	58
<b>Gambar 3.9a</b> Proses <i>Backtracking</i> $S_3$ Simpul $V_{1,9}$ .....	60
<b>Gambar 3.9b</b> Proses <i>Backtracking</i> $S_3$ Simpul $V_{1,9}$ .....	61
<b>Gambar 3.10a</b> Pohon Solusi $S_{4(11)}$ .....	63
<b>Gambar 3.10b</b> Pohon Solusi $S_{4(11)}$ .....	64
<b>Gambar 3.11</b> Pohon Solusi $S_{5(12)}$ .....	66
<b>Gambar 3.12a</b> Pohon Solusi <i>Backtracking</i> $S_{4(18)}$ .....	67
<b>Gambar 3.12b</b> Pohon Solusi <i>Backtracking</i> $S_{4(18)}$ .....	68
<b>Gambar 3.13</b> Pohon Solusi $S_{5(19)}$ .....	69
<b>Gambar 3.14a</b> Pohon Solusi <i>Backtracking</i> $S_{4(21)}$ .....	70
<b>Gambar 3.14b</b> Pohon Solusi <i>Backtracking</i> $S_{4(21)}$ .....	71
<b>Gambar 3.14c</b> Pohon Solusi <i>Backtracking</i> $S_{4(21)}$ .....	72
<b>Gambar 3.15</b> Pohon Solusi $S_{5(22)}$ .....	73
<b>Gambar 3.16</b> Pohon Solusi $S_{6(23)}$ .....	74
<b>Gambar 3.17</b> Pohon Solusi $S_{5(24)}$ .....	75
<b>Gambar 3.18a</b> Pohon Solusi <i>Backtracking</i> $S_{4(29)}$ .....	76
<b>Gambar 3.18b</b> Pohon Solusi <i>Backtracking</i> $S_{4(29)}$ .....	77
<b>Gambar 3.19</b> Pohon Solusi $S_{5(30)}$ .....	78
<b>Gambar 3.20a</b> Pohon Solusi <i>Backtracking</i> $S_{4(31)}$ .....	80
<b>Gambar 3.20b</b> Pohon Solusi <i>Backtracking</i> $S_{4(31)}$ .....	81
<b>Gambar 3.21</b> Pohon Solusi $S_{5(32)}$ .....	82
<b>Gambar 3.22a</b> Pohon Solusi $S_{6(33)}$ .....	83
<b>Gambar 3.22b</b> Pohon Solusi $S_{6(33)}$ .....	84

<b>Gambar 3.23a</b> Pohon Solusi $S_{7(34)}$ .....	86
<b>Gambar 3.23b</b> Pohon Solusi $S_{7(34)}$ .....	87
<b>Gambar 3.24a</b> Pohon Solusi $S_{8(35)}$ .....	89
<b>Gambar 3.24b</b> Pohon Solusi $S_{8(35)}$ .....	90
<b>Gambar 3.25</b> Pohon Solusi $S_{9(36)}$ .....	91
<b>Gambar 3.26</b> Pohon Solusi <i>Backtracking</i> $S_{8(37)}$ .....	93
<b>Gambar 3.27a</b> Pohon Solusi <i>Backtracking</i> $S_{7(39)}$ .....	95
<b>Gambar 3.27b</b> Pohon Solusi <i>Backtracking</i> $S_{7(39)}$ .....	96
<b>Gambar 3.28</b> Pohon Solusi $S_{8(40)}$ .....	98
<b>Gambar 3.29a</b> Pohon Solusi <i>Backtracking</i> $S_{7(42)}$ .....	100
<b>Gambar 3.29b</b> Pohon Solusi <i>Backtracking</i> $S_{7(42)}$ .....	101
<b>Gambar 3.30</b> Pohon Solusi $S_{8(43)}$ .....	103
<b>Gambar 3.31a</b> Pohon Solusi <i>Backtracking</i> $S_{7(44)}$ .....	105
<b>Gambar 3.31b</b> Pohon Solusi <i>Backtracking</i> $S_{7(44)}$ .....	106
<b>Gambar 3.32a</b> Pohon Solusi <i>Backtracking</i> $S_{8(45)}$ .....	108
<b>Gambar 3.32b</b> Pohon Solusi <i>Backtracking</i> $S_{8(45)}$ .....	109
<b>Gambar 3.33</b> Pohon Solusi $S_{9(46)}$ .....	110
<b>Gambar 3.34</b> Pohon Solusi <i>Backtracking</i> $S_{8(48)}$ .....	112
<b>Gambar 3.35</b> Pohon Solusi <i>Backtracking</i> $S_{7(58)}$ .....	114
<b>Gambar 3.36a</b> Pohon Solusi <i>Backtracking</i> $S_{6(59)}$ .....	116
<b>Gambar 3.36b</b> Pohon Solusi <i>Backtracking</i> $S_{6(59)}$ .....	117
<b>Gambar 3.37a</b> Pohon Solusi $S_{7(60)}$ .....	118
<b>Gambar 3.37a</b> Pohon Solusi $S_{7(60)}$ .....	119



<b>Gambar 3.38a</b> Pohon Solusi $S_{8(61)}$ .....	120
<b>Gambar 3.38b</b> Pohon Solusi $S_{8(61)}$ .....	121
<b>Gambar 3.39</b> Pohon Solusi $S_{9(62)}$ .....	123
<b>Gambar 3.40</b> Pohon Solusi <i>Backtracking</i> $S_{8(63)}$ .....	124
<b>Gambar 3.41a</b> Pohon Solusi <i>Backtracking</i> $S_{7(65)}$ .....	126
<b>Gambar 3.41b</b> Pohon Solusi <i>Backtracking</i> $S_{7(65)}$ .....	127
<b>Gambar 3.42</b> Pohon Solusi $S_{8(66)}$ .....	129
<b>Gambar 3.43a</b> Pohon Solusi <i>Backtracking</i> $S_{7(68)}$ .....	131
<b>Gambar 3.43b</b> Pohon Solusi <i>Backtracking</i> $S_{7(68)}$ .....	132
<b>Gambar 3.44</b> Pohon Solusi $S_{8(69)}$ .....	134
<b>Gambar 3.45a</b> Pohon Solusi <i>Backtracking</i> $S_{7(70)}$ .....	136
<b>Gambar 3.45b</b> Pohon Solusi <i>Backtracking</i> $S_{7(70)}$ .....	137
<b>Gambar 3.46a</b> Pohon Solusi $S_{8(71)}$ .....	139
<b>Gambar 3.46b</b> Pohon Solusi $S_{8(71)}$ .....	140
<b>Gambar 3.47a</b> Pohon Solusi $S_{9(72)}$ .....	141
<b>Gambar 3.47b</b> Pohon Solusi $S_{9(72)}$ .....	142
<b>Gambar 3.48</b> Representasi Pewarnaan Graf dalam Permainan <i>Sudoku</i> .....	144
<b>Gambar 3.49</b> <i>Flowchart</i> Penerapan Algoritma <i>Backtracking</i> dalam Pewarnaan Graf untuk Penyelesaian <i>Sudoku</i> .....	145
<b>Gambar 3.50</b> Kode Fungsi Pembuat <i>Grid</i> .....	146
<b>Gambar 3.51</b> Kode Input Permasalahan .....	148
<b>Gambar 3.52</b> Kode Batas Nilai Input .....	148

<b>Gambar 3.53</b> Kode Indeks Simpul Kosong .....	149
<b>Gambar 3.54</b> Kode Penentuan Simpul Kosong dan Fungsi Pembatas .....	151
<b>Gambar 3.55</b> Kode Fungsi Cek .....	152
<b>Gambar 3.56</b> Kode Fungsi Cek Baris dan Kolom .....	153
<b>Gambar 3.57</b> Kode Cek <i>Subgrid</i> .....	154
<b>Gambar 3.58</b> Kode Fungsi Cek Posisi .....	155
<b>Gambar 3.59</b> Kode Fungsi Cek <i>Subgrid</i> .....	155
<b>Gambar 3.60</b> <i>User Interface</i> Aplikasi Penyelesaian Permasalahan <i>Sudoku</i> ..	157
<b>Gambar 3.61</b> <i>User Interface</i> Aplikasi Penyelesaian Permasalahan <i>Sudoku</i> dengan $n = 3$ .....	158
<b>Gambar 3.62</b> Permasalahan <i>Sudoku</i> yang Diinput oleh <i>User</i> .....	159
<b>Gambar 3.63</b> Solusi Penyelesaian Permainan <i>Sudoku</i> .....	160
<b>Gambar 3.61</b> Tampilan <i>Window</i> setelah Bersihkan Angka .....	161

## DAFTAR TABEL

<b>Tabel 1.1</b> Perbedaan Dan Persamaan Penelitian .....	8
<b>Tabel 3.1</b> Indeks Simpul <i>Sudoku</i> .....	50
<b>Tabel 3.2</b> Status Ketetangaan <i>Sudoku</i> $9 \times 9$ .....	51
<b>Tabel 3.3</b> Solusi <i>Subgrid</i> Ke-1 ( $S_1$ ) .....	54
<b>Tabel 3.4</b> Solusi <i>Subgrid</i> Ke-2 ( $S_2$ ) .....	56
<b>Tabel 3.5</b> Solusi <i>Subgrid</i> Ke-3 ( $S_3$ ) .....	59
<b>Tabel 3.6</b> Solusi <i>Subgrid</i> Ke-4 ( $S_4$ ) .....	62
<b>Tabel 3.7</b> Solusi <i>Subgrid</i> Ke-5 ( $S_5$ ) .....	65
<b>Tabel 3.8</b> Solusi <i>Backtracking</i> $S_{4(18)}$ .....	67
<b>Tabel 3.9</b> Solusi $S_{5(19)}$ .....	69
<b>Tabel 3.10</b> Solusi <i>Backtracking</i> $S_{4(21)}$ .....	70
<b>Tabel 3.11</b> Solusi $S_{5(21)}$ .....	72
<b>Tabel 3.12</b> Solusi $S_{6(23)}$ .....	74
<b>Tabel 3.13</b> Solusi <i>Backtracking</i> $S_{5(24)}$ .....	75
<b>Tabel 3.14</b> Solusi <i>Backtracking</i> $S_{4(29)}$ .....	76
<b>Tabel 3.15</b> Solusi <i>Backtracking</i> $S_{5(30)}$ .....	78
<b>Tabel 3.16</b> Solusi <i>Backtracking</i> $S_{4(31)}$ .....	79
<b>Tabel 3.17</b> Solusi $S_{5(32)}$ .....	82
<b>Tabel 3.18</b> Solusi $S_{5(32)}$ .....	83
<b>Tabel 3.19</b> Solusi $S_{7(34)}$ .....	85
<b>Tabel 3.20</b> Solusi $S_{8(35)}$ .....	88
<b>Tabel 3.21</b> Solusi $S_{9(36)}$ .....	91

<b>Tabel 3.22</b> Solusi <i>Backtracking</i> $S_{8(37)}$ .....	92
<b>Tabel 3.23</b> Solusi <i>Backtracking</i> $S_{7(39)}$ .....	94
<b>Tabel 3.24</b> Solusi $S_{8(40)}$ .....	97
<b>Tabel 3.25</b> Solusi <i>Backtracking</i> $S_{7(42)}$ .....	99
<b>Tabel 3.26</b> Solusi $S_{8(43)}$ .....	102
<b>Tabel 3.27</b> Solusi <i>Backtracking</i> $S_{7(44)}$ .....	104
<b>Tabel 3.28</b> Solusi $S_{8(45)}$ .....	107
<b>Tabel 3.29</b> Solusi $S_{9(46)}$ .....	109
<b>Tabel 3.30</b> Solusi <i>Backtracking</i> $S_{8(48)}$ .....	111
<b>Tabel 3.31</b> Solusi <i>Backtracking</i> $S_{7(58)}$ .....	113
<b>Tabel 3.32</b> Solusi <i>Backtracking</i> $S_{6(59)}$ .....	115
<b>Tabel 3.33</b> Solusi $S_{7(60)}$ .....	117
<b>Tabel 3.34</b> Solusi $S_{8(61)}$ .....	120
<b>Tabel 3.35</b> Solusi $S_{9(62)}$ .....	122
<b>Tabel 3.36</b> Solusi <i>Backtracking</i> $S_{8(63)}$ .....	124
<b>Tabel 3.37</b> Solusi <i>Backtracking</i> $S_{7(65)}$ .....	125
<b>Tabel 3.38</b> Solusi $S_{8(66)}$ .....	128
<b>Tabel 3.39</b> Solusi <i>Backtracking</i> $S_{7(68)}$ .....	130
<b>Tabel 3.40</b> Solusi $S_{8(69)}$ .....	133
<b>Tabel 3.41</b> Solusi <i>Backtracking</i> $S_{7(70)}$ .....	135
<b>Tabel 3.42</b> Solusi $S_{8(71)}$ .....	138
<b>Tabel 3.43</b> Solusi $S_{9(72)}$ .....	140
<b>Tabel 3.44</b> Solusi Permainan <i>Sudoku</i> .....	143

## DAFTAR LAMPIRAN

Lampiran 1 Status Ketetanggaan pada Permainan <i>Sudoku</i> $9 \times 9$ .....	168
Lampiran 2 .....	173
1. Gambar 3.10 Pohon Solusi $S_{4(11)}$ .....	173
2. Gambar 3.12a Pohon Solusi <i>Backtracking</i> $S_{4(16)}$ .....	176
3. Gambar 3.12b Pohon Solusi <i>Backtracking</i> $S_{4(18)}$ .....	179
4. Gambar 3.14 Pohon Solusi <i>Backtracking</i> $S_{4(21)}$ .....	181
5. Gambar 3.18 Pohon Solusi <i>Backtracking</i> $S_{4(29)}$ .....	184
6. Gambar 3.27 Pohon Solusi <i>Backtracking</i> $S_{7(39)}$ .....	187
7. Gambar 3.29b Pohon Solusi <i>Backtracking</i> $S_{7(42)}$ .....	191
8. Tabel Solusi <i>Backtracking</i> $S_{7(58)}$ .....	194
9. Gambar 3.35 Pohon Solusi <i>Backtracking</i> $S_{7(58)}$ .....	195
10. Gambar 3.41 Pohon Solusi <i>Backtracking</i> $S_{7(65)}$ .....	200
11. Gambar 3.43 Pohon Solusi <i>Backtracking</i> $S_{7(68)}$ .....	204
12. Tabel 3.44 Solusi Permainan <i>Sudoku</i> .....	208
Lampiran 3 Source Code VB.Net .....	209
Lampiran 4 Output Aplikasi “Pecahkan <i>Sudoku</i> ” .....	218

## DAFTAR LAMBANG

$V(G)$	: Himpunan simpul
$E(G)$	: Himpunan sisi
$d(v)$	: Derajat simpul $v$
$K_n$	: Graf lengkap
$\chi(G)$	: Bilangan kromatik
$\Delta(G)$	: Derajat maksimum dari graf $G$
$\omega(G)$	: Bilangan kelompok ( <i>clique number</i> )
$V_{i,j}$	: Simpul dari baris ke- $i$ dan kolom ke- $j$
$X_n$	: Graf <i>sudoku</i>
$S_i$	: Solusi <i>subgrid</i> ke- $i$
$\parallel$	: atau

## ABSTRAK

# PENERAPAN ALGORITMA RUNUT BALIK (*BACKTRACKING*) DALAM PEWARNAAN GRAF (*GRAPH COLORING*) PADA PENYELESAIAN *SUDOKU*

Oleh  
**Arum Septya Ayu**  
**12610015**

Pewarnaan simpul adalah pemberian warna pada setiap simpul, sehingga tidak ada simpul bertetangga memiliki warna yang sama. Pewarnaan graf khususnya pewarnaan simpul banyak diterapkan dalam menyelesaikan permasalahan. Salah satunya yaitu menyelesaikan permasalahan permainan *sudoku*. Permainan teka-teki angka berbasis logika ini bertujuan untuk mengisikan angka dengan aturan dalam setiap baris, kolom dan *subgrid* tidak ada angka yang sama. Penelitian ini menentukan solusi permainan *sudoku* internasional yang berbentuk bujur sangkar ( $n^2 \times n^2$ ) dengan  $n = 3$  atau ukuran *grid*  $9 \times 9$ .

Diawali dengan merepresentasikan sel menjadi simpul dan relasi ketetanggaan sel-sel menjadi sisi sehingga terbentuk graf *sudoku* ( $X_n$ ). Graf tersebut berhubungan dengan pewarnaan parsial dari graf. Simpul-simpul pada graf *sudoku* diwarnai sesuai dengan kaidah bilangan kromatik. Dalam mewarnai graf *sudoku* terdapat banyak kemungkinan solusi penyelesaian sehingga digunakan algoritma runut balik (*backtracking*). Algoritma ini berfungsi untuk memecahkan suatu permasalahan dengan banyak kemungkinan solusi secara bertahap. Tahap pencarian solusi dibuat dalam pohon berakar yang berbasis pada DFS (*Depth First Search*). Cara kerja algoritma *backtracking* dimulai dengan menentukan fungsi pembatas kemudian membangkitkan solusi. *Backtracking* dilakukan jika terjadi *dead node*, sehingga proses pencarian akan kembali ke simpul hidup terdekat. Kemudian simpul tersebut membangkitkan simpul anak dengan solusi yang baru. Proses pencarian solusi dilakukan secara manual dan rancang bangun menggunakan bahasa pemrograman VB.Net 2015.

Berdasarkan hasil penelitian, diperoleh representasi permainan *sudoku* yaitu graf *sudoku* dengan rank 3 ( $X_3$ ). Graf *sudoku* tersebut diselesaikan dengan pewarnaan graf yang memiliki bilangan kromatik yaitu  $n^2$  serta menerapkan algoritma *backtracking* yang dilakukan secara manual sehingga diperoleh solusi penyelesaian tunggal. Graf *sudoku* ini juga diselesaikan melalui rancang bangun berupa aplikasi pemecahan permainan *sudoku* dengan  $2 \leq n \leq 5$  dan diperoleh solusi penyelesaian tunggal.

Kata kunci: permainan *sudoku*, pewarnaan graf, algoritma runut balik, VB.Net

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Matematika diskrit adalah cabang matematika yang mengkaji tentang objek-objek diskrit. Ilmu ini tidak hanya penting dalam matematika saja namun juga penting pada penerapan ilmu lain seperti dalam teknik informatika dan ilmu komputer. Cabang matematika tersebut selanjutnya dijadikan sebagai landasan matematis terapan ilmu komputer dan informatika. Seiring dengan perkembangan teknologi yang semakin canggih pada saat ini, matematika diskrit juga terus mengalami perkembangan yang sangat pesat. Hal ini diperkuat dengan fakta bahwa komputer digital bekerja secara diskrit dengan menyimpan serta memanipulasi informasi- informasi dalam bentuk diskrit. Seperti dalam perancangan suatu algoritma dan bahasa pemrograman. Kedua hal tersebut menggunakan logika serta bilangan bulat (*integer*) yang bekerja secara diskrit.

Dalam matematika diskrit, teori graf merupakan salah satu pokok bahasan yang populer serta sangat banyak peranannya untuk memecahkan permasalahan dalam kehidupan sehari-hari. Teori-teori yang dikaji dapat digunakan untuk merepresentasikan objek-objek diskrit beserta hubungannya dalam bentuk graf, yaitu objek dinyatakan sebagai noktah, bulatan, titik atau simpul (*node / vertex*) dan hubungan antar objek dinyatakan dengan garis atau sisi (*edge*). Menurut catatan sejarah, masalah Jembatan Königsberg adalah permasalahan awal yang



menggunakan teori graf. Pada tahun 1736, seorang matematikawan Swiss bernama Leonhard Euler berhasil menyelesaikan permasalahan tersebut dengan pembuktian yang sederhana dengan menyatakan daratan sebagai titik/simpul dan jembatan sebagai garis/sisi. Euler memikirkan bahwa permasalahan tersebut dapat diselesaikan berdasarkan derajat dari simpul-simpulnya. Beliau dapat menunjukkan penyelesaian permasalahan dengan sirkuit yang tidak terdiri dari simpul yang berderajat negatif. Sirkuit ini disebut juga sebagai sirkuit Euler. Walaupun pada awalnya teori graf digunakan untuk menyelesaikan masalah Jembatan Königsberg, tetapi saat ini teori tersebut telah mengalami perkembangan yang sangat luas. Salah satu topik menarik yang merupakan perkembangan graf yaitu pewarnaan graf (*graph coloring*).

Pewarnaan graf biasanya berupa pemberian warna pada graf, namun juga dapat dinotasikan dengan bilangan bulat yaitu 1, 2, 3, ... . Ada tiga macam persoalan pewarnaan graf, yaitu pewarnaan simpul atau titik, pewarnaan sisi dan pewarnaan wilayah (*region*). Beberapa permasalahan yang bisa diselesaikan dengan pewarnaan graf, seperti mewarnai peta (*coloring of map*), penentuan jadwal kuliah atau ujian, serta bisa juga diaplikasikan dalam permainan (*game*).

Menurut Daniel Berlyne dalam (Santrock, 2006 : 273), permainan (*game*) sebagai suatu kegiatan yang menyenangkan karena hal tersebut dapat memuaskan dorongan penjelajahan, seperti dorongan rasa keingintahuan akan informasi tentang sesuatu yang baru atau tidak biasa. Kegiatan tersebut dapat menjadi media bagi pemain untuk mendapatkan kemungkinan-kemungkinan baru, kompleksitas atau kerumitan, kejutan dan keanehan. Selain itu, juga dapat menjadi media edukasi

untuk para pemainnya. Permainan edukatif menurut (Adams, 1975), adalah semua bentuk permainan yang dirancang untuk memberikan pengalaman pendidikan atau pengalaman belajar kepada para pemainnya, termasuk permainan tradisional dan nontradisional (*modern*) yang diberi muatan pendidikan dan pengajaran. Salah satu contoh dari permainan edukatif yaitu permainan puzzle. Permainan ini sangat banyak variasinya salah satunya adalah teka-teki angka.

*Sudoku* merupakan permainan teka-teki angka berbasis logika yang sangat populer. Teka-teki angka ini diperkenalkan oleh Howard Gams dan pertama kali diterbitkan oleh Dell Magazine pada tahun 1979. Kemudian menjadi terkenal di Jepang pada tahun 1986. Nama "*Sudoku*" berasal dari bahasa Jepang, "*Suuji wa dokushin ni kagiru*" yang berarti angka-angkanya harus tetap tunggal. Banyak di antara para pelajar bahkan karyawan memainkan *sudoku* di waktu luang untuk mengasah logika. Di Jepang, *sudoku* banyak dimainkan oleh penumpang kereta yang sedang menunggu dan saat dalam perjalanan. Rasa penasaran untuk menyelesaikan permainan ini membuat orang menjadi ketagihan untuk memainkannya. Permainan ini dapat dimainkan oleh berbagai kalangan usia, jenis kelamin maupun jenis pekerjaan. Hal ini dikarenakan *sudoku* merupakan permainan yang sederhana dan semakin tinggi tingkat kesulitannya maka akan semakin menantang para pemain untuk memainkannya.

Pada setiap permasalahan permainan *sudoku* akan ada beberapa angka yang sudah diberikan dan merupakan landasan awal pencarian solusi permasalahan. Jumlah angka-angka yang diberikan dan posisi menjadi penentu tingkat kesulitan dalam permainan ini. Pada umumnya, tujuan permainan ini adalah untuk

mengisikan angka dari 1 sampai 9 ke dalam kotak-kotak (*grid*) berukuran  $9 \times 9$  yang terbagi menjadi 9 kotak berukuran  $3 \times 3$  dan biasa disebut *minigrid* / *subgrid*. Aturan permainan yaitu dalam satu baris, satu kolom dan satu *subgrid* tidak ada angka yang berulang. Beberapa aplikasi permainan *sudoku* telah banyak dikembangkan oleh para programmer. Aplikasi yang dikembangkan yaitu programmer memasukkan aturan permainan *sudoku* ke dalam bahasa pemrograman dan kemudian diproses dengan algoritma tertentu sehingga mencapai solusi terbaik dari permainan tersebut. Salah satu algoritma yang dapat digunakan adalah algoritma runut balik (*backtracking*).

Algoritma runut balik (*backtracking*) adalah algoritma yang merupakan perbaikan dari algoritma *Brute Force* serta berbasis pada DFS (*Depth First Search*). Algoritma ini digunakan untuk memecahkan suatu permasalahan yang memiliki banyak kemungkinan solusi yang di uji secara bertahap, namun tidak perlu memeriksa semua kemungkinan solusi yang ada. Hanya pencarian yang mengarah pada solusi saja yang menjadi pertimbangan. Tahap pencarian solusinya direpresentasikan sebagai suatu pohon berakar.

Berdasarkan pemaparan di atas, maka permainan *sudoku* sangat mungkin untuk diselesaikan dengan pewarnaan graf dan algoritma *backtracking*. Sehingga akan dilakukan penelitian Skripsi yang berjudul **“Penerapan Algoritma Runut Balik (*Backtracking*) dalam Pewarnaan Graf (*Graph Coloring*) Pada Penyelesaian Permainan *Sudoku*”**.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan di atas, maka diperoleh rumusan masalah yang akan dibahas dalam skripsi ini sebagai berikut :

1. Bagaimana merepresentasikan permasalahan permainan *sudoku* ke dalam bentuk graf ?
2. Bagaimana konsep dan cara kerja Algoritma Runut Balik (*Backtracking*)?
3. Bagaimana penerapan Algoritma Runut Balik (*Backtracking*) dalam pewarnaan graf (*Graph Coloring*) pada penyelesaian permainan *sudoku* ?

## 1.3 Tujuan Penelitian

Adapun tujuan dari penulisan Skripsi ini sebagai berikut :

1. Merepresentasikan permasalahan permainan *sudoku* ke dalam bentuk graf, sehingga dapat dicari penyelesaiannya.
2. Menjelaskan konsep dan cara kerja Algoritma *Backtracking*.
3. Menerapkan Algoritma *Backtracking* dalam pewarnaan graf untuk menyelesaikan permainan *sudoku*.

## 1.4 Batasan Masalah

Dalam penulisan skripsi ini diperlukan batasan masalah sebagai berikut:

1. Hanya difokuskan pada Algoritma *Backtracking* dalam pewarnaan graf untuk menyelesaikan permainan *sudoku*.
2. Jenis *sudoku* yang akan dibahas yaitu *sudoku* berbentuk bujur sangkar dengan  $n = 3$  atau ukuran *grid*  $9 \times 9$  yang terdiri dari 9 buah *subgrid* berukuran  $3 \times 3$  dan 81 sel.

3. Pengambilan data untuk pencarian solusi dilakukan per *subgrid* yaitu dimulai dari *subgrid* pertama, kemudian kedua dan urutan selanjutnya sampai ke *subgrid* kesembilan untuk *sudoku* dengan ukuran *grid*  $9 \times 9$ .
4. Program (rancang bangun) pada penelitian ini menggunakan bahasa pemrograman yaitu *Visual Basic.Net* 2015 (VB.Net 2015) dengan batasan  $2 \leq n \leq 5$ .

### 1.5 Manfaat Penelitian

Penelitian ini diharapkan dapat bermanfaat sebagai berikut :

1. Memberikan tambahan wawasan pengetahuan dalam bidang matematika, khususnya mengenai penerapan algoritma *backtracking* dalam penyelesaian permainan *sudoku*.
2. Mengetahui penerapan algoritma *backtracking* dalam bahasa pemrograman sehingga mencapai solusi dengan waktu yang efektif dan optimal.
3. Menjadi referensi untuk penelitian lebih lanjut dalam mengembangkan dan mengaplikasikan ilmu matematika khususnya untuk penerapan algoritma *Backtracking* dan pewarnaan graf.

### 1.6 Tinjauan Pustaka

Tinjauan pustaka dalam penulisan Skripsi ini terinspirasi dari beberapa sumber sebagai berikut :

1. Prosiding yang ditulis oleh Fari Ardilla Adrianto, Yurika Permanasari dan Icih Sukarsih (Universitas Islam Bandung, 2015) dengan judul "*Penerapan Pewarnaan Graf sebagai Metode untuk Mencari Solusi Permainan Sudoku*". Dalam makalahnya, penulis mencari solusi dari permainan *sudoku* berukuran

$9 \times 9$  dengan menerapkan pewarnaan graf. Algoritma yang digunakan untuk mewarnai graf dari permainan *sudoku* adalah algoritma *Welch-Powell*. Pewarnaan graf dengan algoritma ini dimulai dari mewarnai simpul pertama yang memiliki derajat tertinggi.

2. Jurnal yang ditulis oleh Deasy Ramadiyan Sari, Wulan Widyasari, Eunice Sherta Ria (Institut Teknologi Bandung, 2005) dengan judul "*Penerapan Algoritma Backtracking pada Pewarnaan Graf*". Dalam jurnalnya dibahas mengenai analisis Algoritma *Backtracking* untuk pewarnaan graf secara umum serta kompleksitas waktu Algoritma *Backtracking*.
3. Jurnal yang ditulis oleh Vlastimil Chytry (Faculty of Natural Science, Costantine The Philosopher University in Nitra, 2004) dengan judul "*Sudoku Game Solution Based on Graph Theory and Suitable For School-Mathematics*" atau Solusi Permainan *Sudoku* berdasarkan Teori Graf dan sesuai untuk Jurusan Matematika. Jurnal ini difokuskan pada logika matematika untuk menyelesaikan permainan *sudoku* yang bersifat mendidik. Strategi untuk memperoleh kemenangan didasarkan pada teori graf. Dalam jurnal ini, penyelesaian permainan *sudoku*  $4 \times 4$  dianalisis melalui pewarnaan graf dengan menggunakan metode "*step after step*" yang didasarkan pada algoritma heuristik. Penulis menggunakan kombinasi pewarnaan simpul graf (*vertex graph coloring*) dengan pewarnaan sisi graf (*edge graph coloring*).

**Tabel 1.1** Perbedaan dan Persamaan Penelitian

No.	Nama Peneliti	Judul Penelitian	Perbedaan	Persamaan
1.	Fari Ardilla Adrianto, Yurika Permanasari dan Ichi Sukarsih (2015)	<i>Penerapan Pewarnaan Graf sebagai Metode untuk Mencari Solusi Permainan Sudoku</i>	Prosiding tersebut menerapkan pewarnaan graf dengan menggunakan algoritma <i>Welch-Powell</i> untuk mencari solusi permainan <i>sudoku</i> , sedangkan skripsi ini menerapkan algoritma <i>backtracking</i> dalam pewarnaan graf pada permainan <i>sudoku</i> .	Menggunakan pewarnaan graf dan permainan <i>sudoku</i> .
2.	Deasy Ramadiyan Sari, Wulan Widyasari dan Eunice Sherta Ria (2005)	<i>Penerapan Algoritma Backtracking pada Pewarnaan Graf</i>	Jurnal tersebut membahas analisis algoritma <i>backtracking</i> pada pewarnaan graf secara umum, sedangkan skripsi ini meneliti pada penyelesaian permainan <i>sudoku</i> .	Menggunakan algoritma <i>backtracking</i> pada pewarnaan graf.
3.	Vlastimil Chytry (2004)	<i>Sudoku Game Solution Based on Graph Theory and Suitable For School- Mathematics</i>	Jurnal tersebut menyelesaikan permainan <i>sudoku</i> $4 \times 4$ dengan pewarnaan graf menggunakan metode <i>step after step</i> , sedangkan skripsi ini menyelesaikan permainan <i>sudoku</i> $9 \times 9$ dengan algoritma <i>backtracking</i> dalam pewarnaan graf.	Menggunakan pewarnaan graf pada penyelesaian permainan <i>sudoku</i> .

Skripsi yang ditulis oleh Arum Septya Ayu (UIN Sunan Kalijaga, 2016) dengan judul “*Penerapan Algoritma Runut Balik (Backtracking) dalam Pewarnaan Graf (Graph Coloring) pada Penyelesaian Permainan Sudoku*” ini terinspirasi dari ketiga tinjauan pustaka diatas. Dalam penelitian ini akan dicari solusi dari permainan *sudoku* dengan ukuran *grid*  $9 \times 9$  menggunakan metode yang sama dengan ketiga tinjauan pustaka di atas, yaitu pewarnaan graf yang dicari satu per satu kemungkinan solusinya dengan menggunakan algoritma runut balik. Urutan langkah-langkah pencariannya akan direpresentasikan menggunakan pohon berakar dengan metode pencarian *Depth First Search (DFS)*. Kemudian akan dibuat program penyelesaian *sudoku* dengan menggunakan bahasa pemrograman *Visual Basic.Net*. Dengan demikian permainan ini dapat diselesaikan dengan optimal.

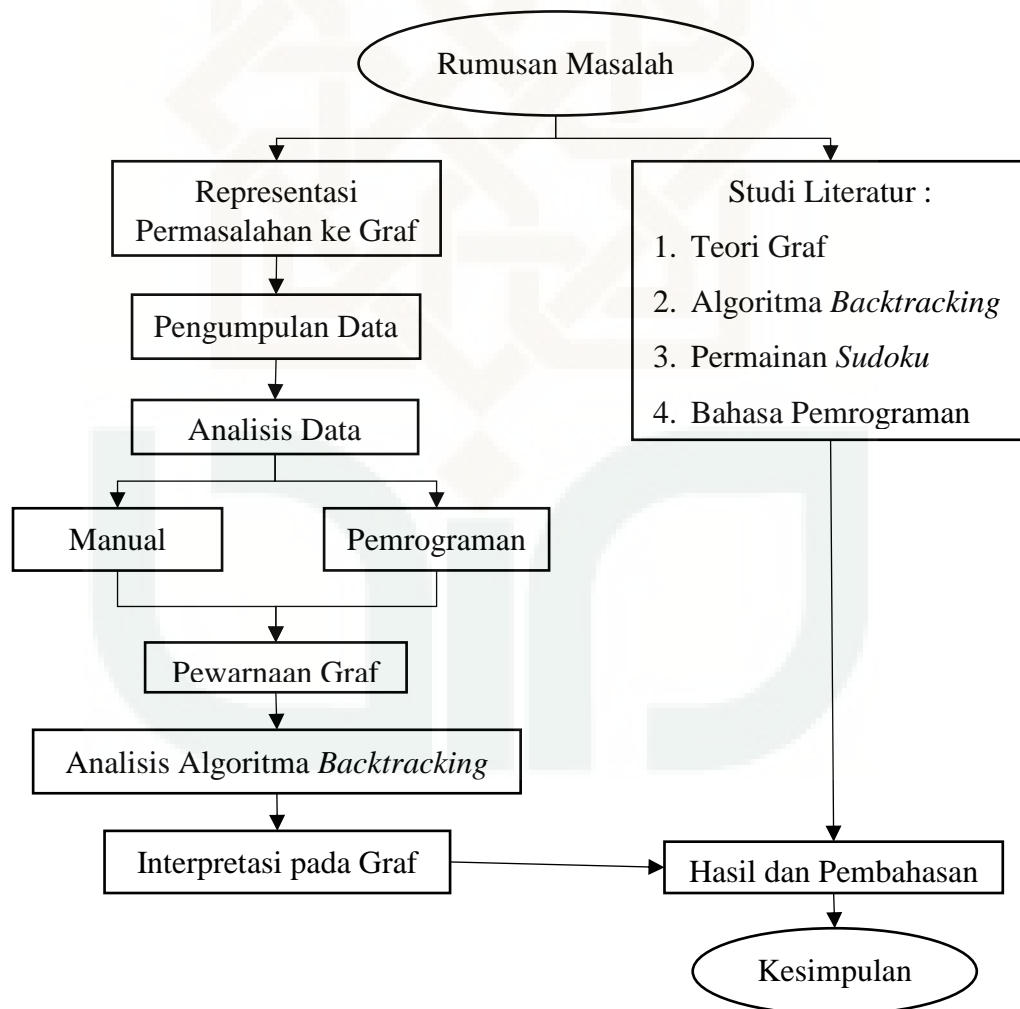
### **1.7 Metode Penelitian**

Berdasarkan tujuannya, penelitian ini termasuk dalam penelitian kualitatif, yaitu penelitian tentang riset yang bersifat deskriptif dan cenderung menggunakan analisis serta lebih menunjukkan proses dan makna. Penelitian ini bersifat deskriptif, yaitu penelitian dengan menggambarkan serta menginterpretasikan suatu objek sesuai dengan fakta-fakta yang ada. Objek penelitian ini adalah permainan *sudoku* dengan *grid* berukuran  $9 \times 9$ .

Metode penelitian yang digunakan dalam penulisan skripsi ini adalah metode penelitian kepustakaan (*library research*), yaitu dengan mengumpulkan data dan informasi dari berbagai materi yang terdapat di perpustakaan berbasis kualitatif melalui sumber-sumber buku, jurnal, skripsi, majalah, catatan dan artikel yang mendukung penyelesaian permasalahan.



Penelitian ini diawali dari merumuskan masalah mengenai penyelesaian *sudoku* dengan menerapkan algoritma *Backtracking* dalam pewarnaan graf. Kemudian, dilanjutkan dengan mengumpulkan data dan informasi melalui sumber-sumber buku, jurnal, skripsi, artikel, handout / diktat kuliah, dan lain-lain yang dapat menjadi referensi penelitian sehingga diperoleh penyelesaian hasil dan pembahasan serta kesimpulan. Langkah-langkah di atas dapat dilihat dalam skema langkah penelitian pada Gambar 1.1 sebagai berikut :



**Gambar 1.1** Skema Langkah Penelitian

## 1.8 Sistematika Penulisan

Sistematika penulisan yang akan diuraikan dalam skripsi ini terbagi menjadi empat bab, sebagai berikut :

### BAB I PENDAHULUAN

Pada bab ini akan dibahas kerangka dari penulisan skripsi yang terdiri dari latar belakang, rumusan masalah, tujuan penelitian, batasan masalah, manfaat penelitian, metode penelitian dan sistematika penulisan.

### BAB II LANDASAN TEORI

Pada bab ini akan dibahas teori-teori dasar yang mendukung penyelesaian rumusan masalah pada bab selanjutnya. Materi yang akan dijadikan landasan teori yaitu teori-teori yang berkaitan dengan matematika diskrit, termasuk pewarnaan graf dan Algoritma *Backtracking* serta berkaitan dengan permainan *sudoku*.

### BAB III PEMBAHASAN

Pada bab ini berisi pembahasan yang membahas konsep dan langkah-langkah penerapan algoritma *Backtracking* dalam pewarnaan graf untuk menyelesaikan permasalahan *sudoku* serta hasil penelitian sesuai dengan yang telah dirumuskan pada rumusan masalah.

### BAB IV PENUTUP

Pada bab ini berisi kesimpulan dan saran-saran yang membangun berdasarkan hasil penelitian sesuai dengan pembahasan yang telah diuraikan.

## BAB IV

### PENUTUP

#### 4.1 Kesimpulan

Berdasarkan hasil pembahasan mengenai Penerapan Algoritma Runut Balik (*Backtracking*) dalam Pewarnaan Graf (*Graph Coloring*) pada Penyelesaian Permainan *Sudoku* dapat disimpulkan sebagai berikut :

1. Dibuat representasi permasalahan permainan *sudoku* ke dalam bentuk graf dengan mengasumsikan sel-sel ke dalam bentuk simpul (*node*) dan relasi ketetanggaan dari setiap simpul sebagai sisi (*edge*). Suatu simpul dikatakan saling bertetangga dengan simpul yang lainnya jika suatu simpul tersebut berada pada baris, kolom dan *subgrid* yang sama. Setelah semua simpul saling berhubungan maka terbentuklah konstruksi graf *sudoku* dengan *rank* 3 ( $X_3$ ) dari data permasalahan permainan *sudoku* dengan ukuran *grid*  $9 \times 9$ .
2. Permasalahan permainan *sudoku* dapat diselesaikan menggunakan Algoritma *Backtracking*, yaitu dengan cara merunut satu persatu kemungkinan solusi yang ada. Konsep dari algoritma ini yaitu pencarian solusi dibuat dalam bentuk pohon berakar (*rooted tree*) yang berbasis pada *Depth First Search* (DFS). Selanjutnya algoritma ini diterapkan dalam pewarnaan graf dalam menyelesaikan permainan *sudoku* yang diselesaikan per *subgrid*. Pencarian penyelesaian dimulai dari menentukan fungsi pembatas yang berisi kemungkinan-kemungkinan warna dari solusi yang dinotasikan dengan

bilangan bulat positif seperti 1,2, ...,9. Selanjutnya membangkitkan solusi warna berdasarkan urutan bilangan dalam fungsi pembatas. Jika terjadi *dead node* (simpul mati) maka dilakukan runut balik ke simpul hidup terdekat. Kemudian simpul hidup tersebut akan membangkitkan simpul anak dengan solusi yang baru. Proses ini dilakukan hingga permainan *sudoku* dapat diselesaikan.

3. Permainan *sudoku* termasuk dalam pewarnaan parsial dari graf. Teorema 3.1 menunjukkan bilangan kromatik dari graf *sudoku*, yaitu untuk *sudoku* berukuran  $n^2 \times n^2$  memiliki bilangan kromatik  $n^2$ . Pada graf *sudoku*  $X_3$  dapat digunakan minimal 9 buah warna untuk mencapai solusi permainan yang dapat diselesaikan melalui proses manual dan rancang bangun menggunakan bahasa pemrograman VB.Net 2015. Rancang bangun yang dihasilkan berupa aplikasi penyelesaian permainan *sudoku* yang diberi nama "Pecahkan Sudoku". Aplikasi ini berfungsi untuk mencari solusi penyelesaian dengan tingkat efisiensi waktu yang lebih singkat dibandingkan dengan proses pencarian manual. Dengan menerapkan algoritma *backtracking* dalam pewarnaan graf pada penyelesaian *sudoku*  $9 \times 9$  maka diperoleh solusi penyelesaian tunggal. Solusi tersebut dikelompokkan pada tabel solusi permainan *sudoku* (lihat Tabel 3.44) dan direpresentasikan dalam bentuk graf yang simpul-simpulnya telah diwarnai (lihat Gambar 3.48). Solusi yang dicari dengan proses manual sama dengan solusi yang dicari dengan rancang bangun. Perbedaannya hanya terletak pada jumlah iterasi penyelesaian. Untuk studi kasus penelitian ini, pada proses manual diperoleh

iterasi penyelesaian sebanyak 72 iterasi. Sedangkan pada proses rancang bangun diperoleh 92 iterasi penyelesaian (lihat Gambar 3.60).

#### 4.2 Saran

Berdasarkan penelitian yang telah dilakukan, maka saran-saran yang disampaikan sebagai berikut :

1. Pada penelitian ini menggunakan algoritma *backtracking* dalam pewarnaan graf untuk menyelesaikan permainan *sudoku*. Diharapkan dapat dijadikan inspirasi untuk penelitian selanjutnya. Misalnya dengan objek yang sama namun menggunakan algoritma lain, seperti algoritma genetika, *harmony search*, dan lain-lain atau sebaliknya dengan algoritma yang sama dapat memilih objek permainan lain, seperti teka-teki silang, labirin, *tic-tac-toe*, dan lain-lain atau dengan objek permainan *sudoku* tetapi dengan ukuran *grid* yang berbeda, misalnya  $16 \times 16$ ,  $25 \times 25$  dan seterusnya. Dengan demikian, dapat memberikan hasil penelitian yang bervariasi.
2. Penyelesaian permainan *sudoku* dalam pewarnaan graf dengan menggunakan algoritma *backtracking* dalam pewarnaan graf menggunakan bahasa pemrograman VB.Net. Sehingga disarankan bagi peneliti selanjutnya dapat menggunakan bahasa pemrograman lain, seperti Matlab, PHP, C++, Java, dan lain-lain.

## DAFTAR PUSTAKA

- Adrianto, Fari Ardilla. Permanasari, Yurika. Sukarsih, Ichi. 2015. *Penerapan Pewarnaan Graf sebagai Metode untuk Mencari Solusi Permainan Sudoku*. Prodi Matematika, Fakultas MIPA, Universitas Islam Bandung. (ISSN dan EISSN 1234-1234)
- Aldous, Joan M.. Wilson, Robin J.. 2000. *Graphs and Applications (An Introductory Approach)*. New York : Springer-Verlag.
- Andi. 2003. *Panduan Praktis Beralih ke Pemrograman Visual Basic.NET*. Yogyakarta : Andi.
- Chytry, Vlastimil. 2004. *Sudoku Game Based on Graph Theory and Suitable For School-Mathematics*. Faculty of Natural Sciences, Costantine The Philosopher University in Nitra, Acta Mathematica 17.
- Halvorson, Michael. 2008. *Microsoft Visual Basic 2008*. Washington : Microsoft Press.
- Herzberg, Agnes M.. Murty, M. Ram. 2007. *Sudoku Squares and Chromatic Polynomials*. American Mathematical Society (AMS).
- Jussien, Narendra. 2007. *A-Z of Sudoku*. London : ISTE Ltd.
- Lipschutz, Seymour dan Lipson, Marc. 2008. *Schaum's Outline : Matematika Diskret, Edisi Ketiga*. Jakarta : Erlangga.
- Manongga, Danny dan Nataliani, Yessica. 2013. *Matematika Diskrit*. Jakarta : Prenadamedia Group.
- Mitchell, Kim. 2007. *Thesis: Bounds On The Chromatic Number*. University of Colorado at Denver and Health Sciences Center.
- Molloy, Michael. Reed, Bruce. 2000. *Graph Colouring and The Probabilistic Method*. New York : Springer.
- MT, Suryadi. 1996. *Pengantar Analisis Algoritma*. Jakarta : Gunadarma.
- Munir, Rinaldi. 2012. *Matematika Diskrit*. Bandung : Informatika.
- Niswah, Lutfi Arisatun. 2009. Skripsi: *Pembuktian Teorema Polinomial Khromatik dalam Sudoku*. Universitas Islam Negeri Malang.
- Rosen, Kenneth H. 2012. *Discrete Mathematics and Its Applications, Seventh Edition*. New York : McGraw-Hill.

- Santrock, John W. 2002. *Life Span Development: Perkembangan Masa Hidup, Jilid I*. Jakarta : Erlangga.
- Sari, Deasy Ramadiyan. Widyasari, Wulan. Ria, Eunice Sherta. 2005. *Penerapan Algoritma Backtracking pada Pewarnaan Graf*. Departemen Teknik Informatika, Fakultas Teknologi Industri, Institut Teknologi Bandung (ITB).
- Sari, Rina Dewi Indah. 2011. *Analisis Penyelesaian Puzzle Sudoku dengan Menerapkan Algoritma Backtracking*. Sekolah Tinggi Teknik Surabaya.
- Sarkar, Amites. 2008. *Lecture Handout: Brooks' Theorem*. <http://myweb.facstaff.wwu.edu/sarkara/brooks.pdf> di akses pada tanggal 16 Oktober 2016 pukul 16.52.
- Schaab, Beth. 2008. *Paper: Finding Bounds for The Number of Sudoku Squares*. [http://www.siue.edu/~aweyhau/teaching/seniorprojects/schaab\\_final\\_paper.pdf](http://www.siue.edu/~aweyhau/teaching/seniorprojects/schaab_final_paper.pdf) di akses pada tanggal 21 April 2016 pukul 23.18.
- Yossi, Hanay Dian. 2013. Skripsi: *Penerapan Algoritma Runut Balik (Backtracking) dalam n-Queen Problem Permainan Catur*. UIN Sunan Kalijaga Yogyakarta.

## LAMPIRAN 1

### Status Ketetanggaan pada Permainan *Sudoku* $9 \times 9$

$V_{i,j}$	Titik-titik Tetangga																			
$V_{1,1}$	$V_{1,2}$	$V_{1,3}$	$V_{1,4}$	$V_{1,5}$	$V_{1,6}$	$V_{1,7}$	$V_{1,8}$	$V_{1,9}$	$V_{2,1}$	$V_{2,2}$	$V_{2,3}$	$V_{3,1}$	$V_{3,2}$	$V_{3,3}$	$V_{4,1}$	$V_{5,1}$	$V_{6,1}$	$V_{7,1}$	$V_{8,1}$	$V_{9,1}$
$V_{1,2}$	$V_{1,1}$	$V_{1,3}$	$V_{1,4}$	$V_{1,5}$	$V_{1,6}$	$V_{1,7}$	$V_{1,8}$	$V_{1,9}$	$V_{2,1}$	$V_{2,2}$	$V_{2,3}$	$V_{3,1}$	$V_{3,2}$	$V_{3,3}$	$V_{4,2}$	$V_{5,2}$	$V_{6,2}$	$V_{7,2}$	$V_{8,2}$	$V_{9,2}$
$V_{1,3}$	$V_{1,1}$	$V_{1,2}$	$V_{1,4}$	$V_{1,5}$	$V_{1,6}$	$V_{1,7}$	$V_{1,8}$	$V_{1,9}$	$V_{2,1}$	$V_{2,2}$	$V_{2,3}$	$V_{3,1}$	$V_{3,2}$	$V_{3,3}$	$V_{4,3}$	$V_{5,3}$	$V_{6,3}$	$V_{7,3}$	$V_{8,3}$	$V_{9,3}$
$V_{1,4}$	$V_{1,1}$	$V_{1,2}$	$V_{1,3}$	$V_{1,5}$	$V_{1,6}$	$V_{1,7}$	$V_{1,8}$	$V_{1,9}$	$V_{2,4}$	$V_{2,5}$	$V_{2,6}$	$V_{3,4}$	$V_{3,5}$	$V_{3,6}$	$V_{4,4}$	$V_{5,4}$	$V_{6,4}$	$V_{7,4}$	$V_{8,4}$	$V_{9,4}$
$V_{1,5}$	$V_{1,1}$	$V_{1,2}$	$V_{1,3}$	$V_{1,4}$	$V_{1,6}$	$V_{1,7}$	$V_{1,8}$	$V_{1,9}$	$V_{2,4}$	$V_{2,5}$	$V_{2,6}$	$V_{3,4}$	$V_{3,5}$	$V_{3,6}$	$V_{4,5}$	$V_{5,5}$	$V_{6,5}$	$V_{7,5}$	$V_{8,5}$	$V_{9,5}$
$V_{1,6}$	$V_{1,1}$	$V_{1,2}$	$V_{1,3}$	$V_{1,4}$	$V_{1,5}$	$V_{1,7}$	$V_{1,8}$	$V_{1,9}$	$V_{2,4}$	$V_{2,5}$	$V_{2,6}$	$V_{3,4}$	$V_{3,5}$	$V_{3,6}$	$V_{4,6}$	$V_{5,6}$	$V_{6,6}$	$V_{7,6}$	$V_{8,6}$	$V_{9,6}$
$V_{1,7}$	$V_{1,1}$	$V_{1,2}$	$V_{1,3}$	$V_{1,4}$	$V_{1,5}$	$V_{1,6}$	$V_{1,8}$	$V_{1,9}$	$V_{2,7}$	$V_{2,8}$	$V_{2,9}$	$V_{3,7}$	$V_{3,8}$	$V_{3,9}$	$V_{4,7}$	$V_{5,7}$	$V_{6,7}$	$V_{7,7}$	$V_{8,7}$	$V_{9,7}$
$V_{1,8}$	$V_{1,1}$	$V_{1,2}$	$V_{1,3}$	$V_{1,4}$	$V_{1,5}$	$V_{1,6}$	$V_{1,7}$	$V_{1,9}$	$V_{2,7}$	$V_{2,8}$	$V_{2,9}$	$V_{3,7}$	$V_{3,8}$	$V_{3,9}$	$V_{4,8}$	$V_{5,8}$	$V_{6,8}$	$V_{7,8}$	$V_{8,8}$	$V_{9,8}$
$V_{1,9}$	$V_{1,1}$	$V_{1,2}$	$V_{1,3}$	$V_{1,4}$	$V_{1,5}$	$V_{1,6}$	$V_{1,7}$	$V_{1,8}$	$V_{2,7}$	$V_{2,8}$	$V_{2,9}$	$V_{3,7}$	$V_{3,8}$	$V_{3,9}$	$V_{4,9}$	$V_{5,9}$	$V_{6,9}$	$V_{7,9}$	$V_{8,9}$	$V_{9,9}$
$V_{2,1}$	$V_{1,1}$	$V_{1,2}$	$V_{1,3}$	$V_{2,2}$	$V_{2,3}$	$V_{2,4}$	$V_{2,5}$	$V_{2,6}$	$V_{2,7}$	$V_{2,8}$	$V_{2,9}$	$V_{3,1}$	$V_{3,2}$	$V_{3,3}$	$V_{4,1}$	$V_{5,1}$	$V_{6,1}$	$V_{7,1}$	$V_{8,1}$	$V_{9,1}$
$V_{2,2}$	$V_{1,1}$	$V_{1,2}$	$V_{1,3}$	$V_{2,1}$	$V_{2,3}$	$V_{2,4}$	$V_{2,5}$	$V_{2,6}$	$V_{2,7}$	$V_{2,8}$	$V_{2,9}$	$V_{3,1}$	$V_{3,2}$	$V_{3,3}$	$V_{4,2}$	$V_{5,2}$	$V_{6,2}$	$V_{7,2}$	$V_{8,2}$	$V_{9,2}$
$V_{2,3}$	$V_{1,1}$	$V_{1,2}$	$V_{1,3}$	$V_{2,1}$	$V_{2,2}$	$V_{2,4}$	$V_{2,5}$	$V_{2,6}$	$V_{2,7}$	$V_{2,8}$	$V_{2,9}$	$V_{3,1}$	$V_{3,2}$	$V_{3,3}$	$V_{4,3}$	$V_{5,3}$	$V_{6,3}$	$V_{7,3}$	$V_{8,3}$	$V_{9,3}$
$V_{2,4}$	$V_{1,4}$	$V_{1,5}$	$V_{1,6}$	$V_{2,1}$	$V_{2,2}$	$V_{2,3}$	$V_{2,5}$	$V_{2,6}$	$V_{2,7}$	$V_{2,8}$	$V_{2,9}$	$V_{3,4}$	$V_{3,5}$	$V_{3,6}$	$V_{4,4}$	$V_{5,4}$	$V_{6,4}$	$V_{7,4}$	$V_{8,4}$	$V_{9,4}$



$V_{2,5}$	$V_{1,4}$	$V_{1,5}$	$V_{1,6}$	$V_{2,1}$	$V_{2,2}$	$V_{2,3}$	$V_{2,4}$	$V_{2,6}$	$V_{2,7}$	$V_{2,8}$	$V_{2,9}$	$V_{3,4}$	$V_{3,5}$	$V_{3,6}$	$V_{4,5}$	$V_{5,5}$	$V_{6,5}$	$V_{7,5}$	$V_{8,5}$	$V_{9,5}$
$V_{2,6}$	$V_{1,4}$	$V_{1,5}$	$V_{1,6}$	$V_{2,1}$	$V_{2,2}$	$V_{2,3}$	$V_{2,4}$	$V_{2,5}$	$V_{2,7}$	$V_{2,8}$	$V_{2,9}$	$V_{3,4}$	$V_{3,5}$	$V_{3,6}$	$V_{4,6}$	$V_{5,6}$	$V_{6,6}$	$V_{7,6}$	$V_{8,6}$	$V_{9,6}$
$V_{2,7}$	$V_{1,7}$	$V_{1,8}$	$V_{1,9}$	$V_{2,1}$	$V_{2,2}$	$V_{2,3}$	$V_{2,4}$	$V_{2,5}$	$V_{2,6}$	$V_{2,8}$	$V_{2,9}$	$V_{3,7}$	$V_{3,8}$	$V_{3,9}$	$V_{4,7}$	$V_{5,7}$	$V_{6,7}$	$V_{7,7}$	$V_{8,7}$	$V_{9,7}$
$V_{2,8}$	$V_{1,7}$	$V_{1,8}$	$V_{1,9}$	$V_{2,1}$	$V_{2,2}$	$V_{2,3}$	$V_{2,4}$	$V_{2,5}$	$V_{2,6}$	$V_{2,7}$	$V_{2,9}$	$V_{3,7}$	$V_{3,8}$	$V_{3,9}$	$V_{4,8}$	$V_{5,8}$	$V_{6,8}$	$V_{7,8}$	$V_{8,8}$	$V_{9,8}$
$V_{2,9}$	$V_{1,7}$	$V_{1,8}$	$V_{1,9}$	$V_{2,1}$	$V_{2,2}$	$V_{2,3}$	$V_{2,4}$	$V_{2,5}$	$V_{2,6}$	$V_{2,7}$	$V_{2,8}$	$V_{3,7}$	$V_{3,8}$	$V_{3,9}$	$V_{4,9}$	$V_{5,9}$	$V_{6,9}$	$V_{7,9}$	$V_{8,9}$	$V_{9,9}$
$V_{3,1}$	$V_{1,1}$	$V_{1,2}$	$V_{1,3}$	$V_{2,1}$	$V_{2,2}$	$V_{2,3}$	$V_{3,2}$	$V_{3,3}$	$V_{3,4}$	$V_{3,5}$	$V_{3,6}$	$V_{3,7}$	$V_{3,8}$	$V_{3,9}$	$V_{4,1}$	$V_{5,1}$	$V_{6,1}$	$V_{7,1}$	$V_{8,1}$	$V_{9,1}$
$V_{3,2}$	$V_{1,1}$	$V_{1,2}$	$V_{1,3}$	$V_{2,1}$	$V_{2,2}$	$V_{2,3}$	$V_{3,1}$	$V_{3,3}$	$V_{3,4}$	$V_{3,5}$	$V_{3,6}$	$V_{3,7}$	$V_{3,8}$	$V_{3,9}$	$V_{4,2}$	$V_{5,2}$	$V_{6,2}$	$V_{7,2}$	$V_{8,2}$	$V_{9,2}$
$V_{3,3}$	$V_{1,1}$	$V_{1,2}$	$V_{1,3}$	$V_{2,1}$	$V_{2,2}$	$V_{2,3}$	$V_{3,1}$	$V_{3,2}$	$V_{3,4}$	$V_{3,5}$	$V_{3,6}$	$V_{3,7}$	$V_{3,8}$	$V_{3,9}$	$V_{4,3}$	$V_{5,3}$	$V_{6,3}$	$V_{7,3}$	$V_{8,3}$	$V_{9,3}$
$V_{3,4}$	$V_{1,4}$	$V_{1,5}$	$V_{1,6}$	$V_{2,4}$	$V_{2,5}$	$V_{2,6}$	$V_{3,1}$	$V_{3,2}$	$V_{3,3}$	$V_{3,5}$	$V_{3,6}$	$V_{3,7}$	$V_{3,8}$	$V_{3,9}$	$V_{4,4}$	$V_{5,4}$	$V_{6,4}$	$V_{7,4}$	$V_{8,4}$	$V_{9,4}$
$V_{3,5}$	$V_{1,4}$	$V_{1,5}$	$V_{1,6}$	$V_{2,4}$	$V_{2,5}$	$V_{2,6}$	$V_{3,1}$	$V_{3,2}$	$V_{3,3}$	$V_{3,4}$	$V_{3,6}$	$V_{3,7}$	$V_{3,8}$	$V_{3,9}$	$V_{4,5}$	$V_{5,5}$	$V_{6,5}$	$V_{7,5}$	$V_{8,5}$	$V_{9,5}$
$V_{3,6}$	$V_{1,4}$	$V_{1,5}$	$V_{1,6}$	$V_{2,4}$	$V_{2,5}$	$V_{2,6}$	$V_{3,1}$	$V_{3,2}$	$V_{3,3}$	$V_{3,4}$	$V_{3,5}$	$V_{3,7}$	$V_{3,8}$	$V_{3,9}$	$V_{4,6}$	$V_{5,6}$	$V_{6,6}$	$V_{7,6}$	$V_{8,6}$	$V_{9,6}$
$V_{3,7}$	$V_{1,7}$	$V_{1,8}$	$V_{1,9}$	$V_{2,7}$	$V_{2,8}$	$V_{2,9}$	$V_{3,1}$	$V_{3,2}$	$V_{3,3}$	$V_{3,4}$	$V_{3,5}$	$V_{3,6}$	$V_{3,8}$	$V_{3,9}$	$V_{4,7}$	$V_{5,7}$	$V_{6,7}$	$V_{7,7}$	$V_{8,7}$	$V_{9,7}$
$V_{3,8}$	$V_{1,7}$	$V_{1,8}$	$V_{1,9}$	$V_{2,7}$	$V_{2,8}$	$V_{2,9}$	$V_{3,1}$	$V_{3,2}$	$V_{3,3}$	$V_{3,4}$	$V_{3,5}$	$V_{3,6}$	$V_{3,7}$	$V_{3,9}$	$V_{4,8}$	$V_{5,8}$	$V_{6,8}$	$V_{7,8}$	$V_{8,8}$	$V_{9,8}$
$V_{3,9}$	$V_{1,7}$	$V_{1,8}$	$V_{1,9}$	$V_{2,7}$	$V_{2,8}$	$V_{2,9}$	$V_{3,1}$	$V_{3,2}$	$V_{3,3}$	$V_{3,4}$	$V_{3,5}$	$V_{3,6}$	$V_{3,7}$	$V_{3,8}$	$V_{4,9}$	$V_{5,9}$	$V_{6,9}$	$V_{7,9}$	$V_{8,9}$	$V_{9,9}$
$V_{4,1}$	$V_{1,1}$	$V_{2,1}$	$V_{3,1}$	$V_{4,2}$	$V_{4,3}$	$V_{4,4}$	$V_{4,5}$	$V_{4,6}$	$V_{4,7}$	$V_{4,8}$	$V_{4,9}$	$V_{5,1}$	$V_{5,2}$	$V_{5,3}$	$V_{6,1}$	$V_{6,2}$	$V_{6,3}$	$V_{7,1}$	$V_{8,1}$	$V_{9,1}$
$V_{4,2}$	$V_{1,2}$	$V_{2,2}$	$V_{3,2}$	$V_{4,1}$	$V_{4,3}$	$V_{4,4}$	$V_{4,5}$	$V_{4,6}$	$V_{4,7}$	$V_{4,8}$	$V_{4,9}$	$V_{5,1}$	$V_{5,2}$	$V_{5,3}$	$V_{6,1}$	$V_{6,2}$	$V_{6,3}$	$V_{7,2}$	$V_{8,2}$	$V_{9,2}$
$V_{4,3}$	$V_{1,3}$	$V_{2,3}$	$V_{3,3}$	$V_{4,1}$	$V_{4,2}$	$V_{4,4}$	$V_{4,5}$	$V_{4,6}$	$V_{4,7}$	$V_{4,8}$	$V_{4,9}$	$V_{5,1}$	$V_{5,2}$	$V_{5,3}$	$V_{6,1}$	$V_{6,2}$	$V_{6,3}$	$V_{7,3}$	$V_{8,3}$	$V_{9,3}$

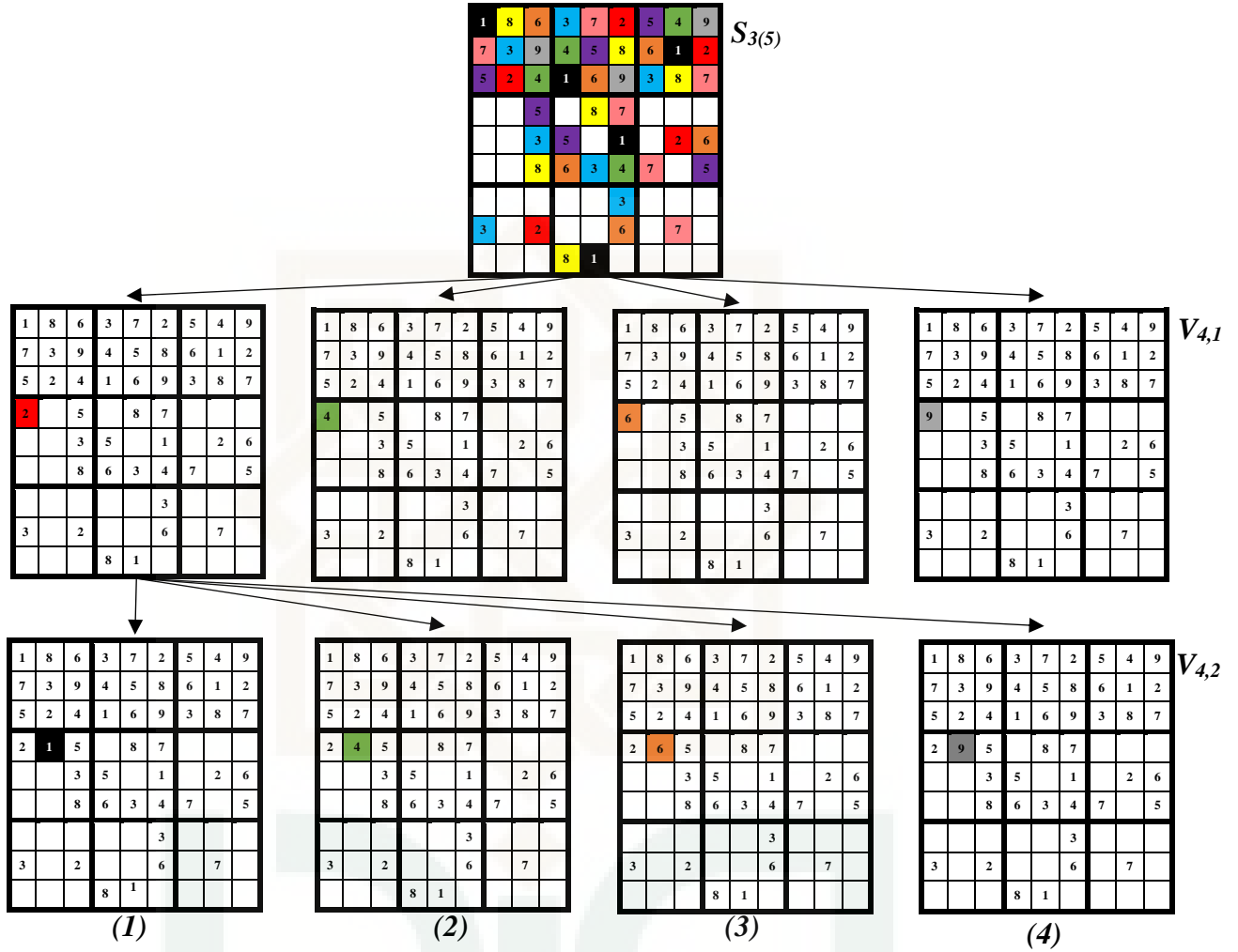


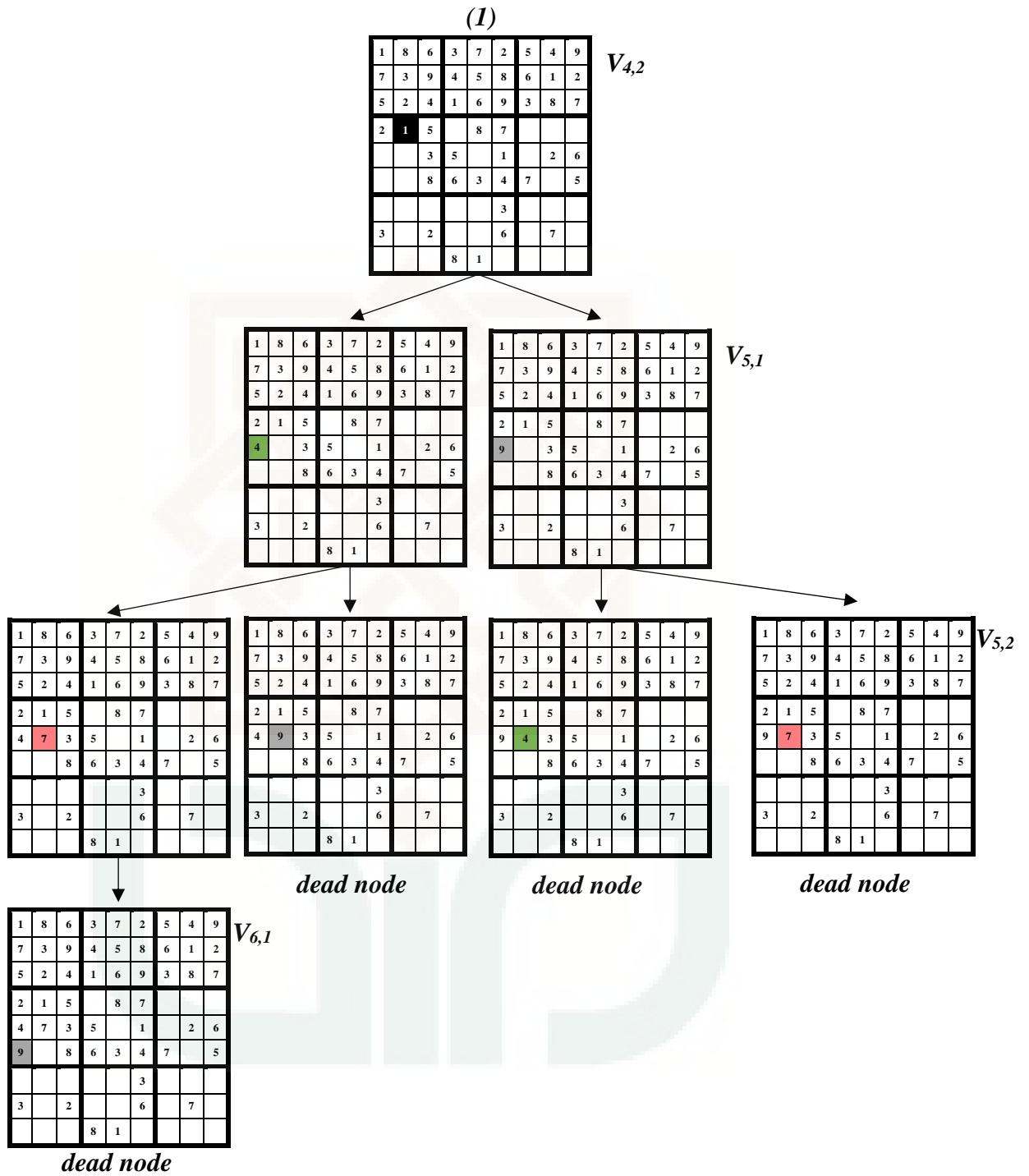
<b>V<sub>6,3</sub></b>	V <sub>1,3</sub>	V <sub>2,3</sub>	V <sub>3,3</sub>	V <sub>4,1</sub>	V <sub>4,2</sub>	V <sub>4,3</sub>	V <sub>5,1</sub>	V <sub>5,2</sub>	V <sub>5,3</sub>	V <sub>6,1</sub>	V <sub>6,2</sub>	V <sub>6,4</sub>	V <sub>6,5</sub>	V <sub>6,6</sub>	V <sub>6,7</sub>	V <sub>6,8</sub>	V <sub>6,9</sub>	V <sub>7,3</sub>	V <sub>8,3</sub>	V <sub>9,3</sub>
<b>V<sub>6,4</sub></b>	V <sub>1,4</sub>	V <sub>2,4</sub>	V <sub>3,4</sub>	V <sub>4,4</sub>	V <sub>4,5</sub>	V <sub>4,6</sub>	V <sub>5,4</sub>	V <sub>5,5</sub>	V <sub>5,6</sub>	V <sub>6,1</sub>	V <sub>6,2</sub>	V <sub>6,3</sub>	V <sub>6,5</sub>	V <sub>6,6</sub>	V <sub>6,7</sub>	V <sub>6,8</sub>	V <sub>6,9</sub>	V <sub>7,4</sub>	V <sub>8,4</sub>	V <sub>9,4</sub>
<b>V<sub>6,5</sub></b>	V <sub>1,5</sub>	V <sub>2,5</sub>	V <sub>3,5</sub>	V <sub>4,4</sub>	V <sub>4,5</sub>	V <sub>4,6</sub>	V <sub>5,4</sub>	V <sub>5,5</sub>	V <sub>5,6</sub>	V <sub>6,1</sub>	V <sub>6,2</sub>	V <sub>6,3</sub>	V <sub>6,4</sub>	V <sub>6,6</sub>	V <sub>6,7</sub>	V <sub>6,8</sub>	V <sub>6,9</sub>	V <sub>7,5</sub>	V <sub>8,5</sub>	V <sub>9,5</sub>
<b>V<sub>6,6</sub></b>	V <sub>1,6</sub>	V <sub>2,6</sub>	V <sub>3,6</sub>	V <sub>4,4</sub>	V <sub>4,5</sub>	V <sub>4,6</sub>	V <sub>5,4</sub>	V <sub>5,5</sub>	V <sub>5,6</sub>	V <sub>6,1</sub>	V <sub>6,2</sub>	V <sub>6,3</sub>	V <sub>6,5</sub>	V <sub>6,6</sub>	V <sub>6,7</sub>	V <sub>6,8</sub>	V <sub>6,9</sub>	V <sub>7,6</sub>	V <sub>8,6</sub>	V <sub>9,6</sub>
<b>V<sub>6,7</sub></b>	V <sub>1,7</sub>	V <sub>2,7</sub>	V <sub>3,7</sub>	V <sub>4,7</sub>	V <sub>4,8</sub>	V <sub>4,9</sub>	V <sub>5,7</sub>	V <sub>5,8</sub>	V <sub>5,9</sub>	V <sub>6,1</sub>	V <sub>6,2</sub>	V <sub>6,3</sub>	V <sub>6,4</sub>	V <sub>6,5</sub>	V <sub>6,6</sub>	V <sub>6,8</sub>	V <sub>6,9</sub>	V <sub>7,7</sub>	V <sub>8,7</sub>	V <sub>9,7</sub>
<b>V<sub>6,8</sub></b>	V <sub>1,8</sub>	V <sub>2,8</sub>	V <sub>3,8</sub>	V <sub>4,7</sub>	V <sub>4,8</sub>	V <sub>4,9</sub>	V <sub>5,7</sub>	V <sub>5,8</sub>	V <sub>5,9</sub>	V <sub>6,1</sub>	V <sub>6,2</sub>	V <sub>6,3</sub>	V <sub>6,4</sub>	V <sub>6,5</sub>	V <sub>6,6</sub>	V <sub>6,7</sub>	V <sub>6,9</sub>	V <sub>7,8</sub>	V <sub>8,8</sub>	V <sub>9,8</sub>
<b>V<sub>6,9</sub></b>	V <sub>1,9</sub>	V <sub>2,9</sub>	V <sub>3,9</sub>	V <sub>4,7</sub>	V <sub>4,8</sub>	V <sub>4,9</sub>	V <sub>5,7</sub>	V <sub>5,8</sub>	V <sub>5,9</sub>	V <sub>6,1</sub>	V <sub>6,2</sub>	V <sub>6,3</sub>	V <sub>6,4</sub>	V <sub>6,5</sub>	V <sub>6,6</sub>	V <sub>6,7</sub>	V <sub>6,8</sub>	V <sub>7,9</sub>	V <sub>8,9</sub>	V <sub>9,9</sub>
<b>V<sub>7,1</sub></b>	V <sub>1,1</sub>	V <sub>2,1</sub>	V <sub>3,1</sub>	V <sub>4,1</sub>	V <sub>5,1</sub>	V <sub>6,1</sub>	V <sub>7,2</sub>	V <sub>7,3</sub>	V <sub>7,4</sub>	V <sub>7,5</sub>	V <sub>7,6</sub>	V <sub>7,7</sub>	V <sub>7,8</sub>	V <sub>7,9</sub>	V <sub>8,1</sub>	V <sub>8,2</sub>	V <sub>8,3</sub>	V <sub>9,1</sub>	V <sub>9,2</sub>	V <sub>9,3</sub>
<b>V<sub>7,2</sub></b>	V <sub>1,2</sub>	V <sub>2,2</sub>	V <sub>3,2</sub>	V <sub>4,2</sub>	V <sub>5,2</sub>	V <sub>6,2</sub>	V <sub>7,1</sub>	V <sub>7,3</sub>	V <sub>7,4</sub>	V <sub>7,5</sub>	V <sub>7,6</sub>	V <sub>7,7</sub>	V <sub>7,8</sub>	V <sub>7,9</sub>	V <sub>8,1</sub>	V <sub>8,2</sub>	V <sub>8,3</sub>	V <sub>9,1</sub>	V <sub>9,2</sub>	V <sub>9,3</sub>
<b>V<sub>7,3</sub></b>	V <sub>1,3</sub>	V <sub>2,3</sub>	V <sub>3,3</sub>	V <sub>4,3</sub>	V <sub>5,3</sub>	V <sub>6,3</sub>	V <sub>7,1</sub>	V <sub>7,2</sub>	V <sub>7,4</sub>	V <sub>7,5</sub>	V <sub>7,6</sub>	V <sub>7,7</sub>	V <sub>7,8</sub>	V <sub>7,9</sub>	V <sub>8,1</sub>	V <sub>8,2</sub>	V <sub>8,3</sub>	V <sub>9,1</sub>	V <sub>9,2</sub>	V <sub>9,3</sub>
<b>V<sub>7,4</sub></b>	V <sub>1,4</sub>	V <sub>2,4</sub>	V <sub>3,4</sub>	V <sub>4,4</sub>	V <sub>5,4</sub>	V <sub>6,4</sub>	V <sub>7,1</sub>	V <sub>7,2</sub>	V <sub>7,3</sub>	V <sub>7,5</sub>	V <sub>7,6</sub>	V <sub>7,7</sub>	V <sub>7,8</sub>	V <sub>7,9</sub>	V <sub>8,4</sub>	V <sub>8,5</sub>	V <sub>8,6</sub>	V <sub>9,4</sub>	V <sub>9,5</sub>	V <sub>9,6</sub>
<b>V<sub>7,5</sub></b>	V <sub>1,5</sub>	V <sub>2,5</sub>	V <sub>3,5</sub>	V <sub>4,5</sub>	V <sub>5,5</sub>	V <sub>6,5</sub>	V <sub>7,1</sub>	V <sub>7,2</sub>	V <sub>7,3</sub>	V <sub>7,4</sub>	V <sub>7,6</sub>	V <sub>7,7</sub>	V <sub>7,8</sub>	V <sub>7,9</sub>	V <sub>8,4</sub>	V <sub>8,5</sub>	V <sub>8,6</sub>	V <sub>9,4</sub>	V <sub>9,5</sub>	V <sub>9,6</sub>
<b>V<sub>7,6</sub></b>	V <sub>1,6</sub>	V <sub>2,6</sub>	V <sub>3,6</sub>	V <sub>4,6</sub>	V <sub>5,6</sub>	V <sub>6,6</sub>	V <sub>7,1</sub>	V <sub>7,2</sub>	V <sub>7,3</sub>	V <sub>7,4</sub>	V <sub>7,5</sub>	V <sub>7,7</sub>	V <sub>7,8</sub>	V <sub>7,9</sub>	V <sub>8,4</sub>	V <sub>8,5</sub>	V <sub>8,6</sub>	V <sub>9,4</sub>	V <sub>9,5</sub>	V <sub>9,6</sub>
<b>V<sub>7,7</sub></b>	V <sub>1,7</sub>	V <sub>2,7</sub>	V <sub>3,7</sub>	V <sub>4,7</sub>	V <sub>5,7</sub>	V <sub>6,7</sub>	V <sub>7,1</sub>	V <sub>7,2</sub>	V <sub>7,3</sub>	V <sub>7,4</sub>	V <sub>7,5</sub>	V <sub>7,6</sub>	V <sub>7,8</sub>	V <sub>7,9</sub>	V <sub>8,7</sub>	V <sub>8,8</sub>	V <sub>8,9</sub>	V <sub>9,7</sub>	V <sub>9,8</sub>	V <sub>9,9</sub>
<b>V<sub>7,8</sub></b>	V <sub>1,8</sub>	V <sub>2,8</sub>	V <sub>3,8</sub>	V <sub>4,8</sub>	V <sub>5,8</sub>	V <sub>6,8</sub>	V <sub>7,1</sub>	V <sub>7,2</sub>	V <sub>7,3</sub>	V <sub>7,4</sub>	V <sub>7,5</sub>	V <sub>7,6</sub>	V <sub>7,7</sub>	V <sub>7,9</sub>	V <sub>8,7</sub>	V <sub>8,8</sub>	V <sub>8,9</sub>	V <sub>9,7</sub>	V <sub>9,8</sub>	V <sub>9,9</sub>
<b>V<sub>7,9</sub></b>	V <sub>1,9</sub>	V <sub>2,9</sub>	V <sub>3,9</sub>	V <sub>4,9</sub>	V <sub>5,9</sub>	V <sub>6,9</sub>	V <sub>7,1</sub>	V <sub>7,2</sub>	V <sub>7,3</sub>	V <sub>7,4</sub>	V <sub>7,5</sub>	V <sub>7,6</sub>	V <sub>7,7</sub>	V <sub>7,8</sub>	V <sub>8,7</sub>	V <sub>8,8</sub>	V <sub>8,9</sub>	V <sub>9,7</sub>	V <sub>9,8</sub>	V <sub>9,9</sub>
<b>V<sub>8,1</sub></b>	V <sub>1,1</sub>	V <sub>2,1</sub>	V <sub>3,1</sub>	V <sub>4,1</sub>	V <sub>5,1</sub>	V <sub>6,1</sub>	V <sub>7,1</sub>	V <sub>7,2</sub>	V <sub>7,3</sub>	V <sub>8,2</sub>	V <sub>8,3</sub>	V <sub>8,4</sub>	V <sub>8,5</sub>	V <sub>8,6</sub>	V <sub>8,7</sub>	V <sub>8,8</sub>	V <sub>8,9</sub>	V <sub>9,1</sub>	V <sub>9,2</sub>	V <sub>9,3</sub>

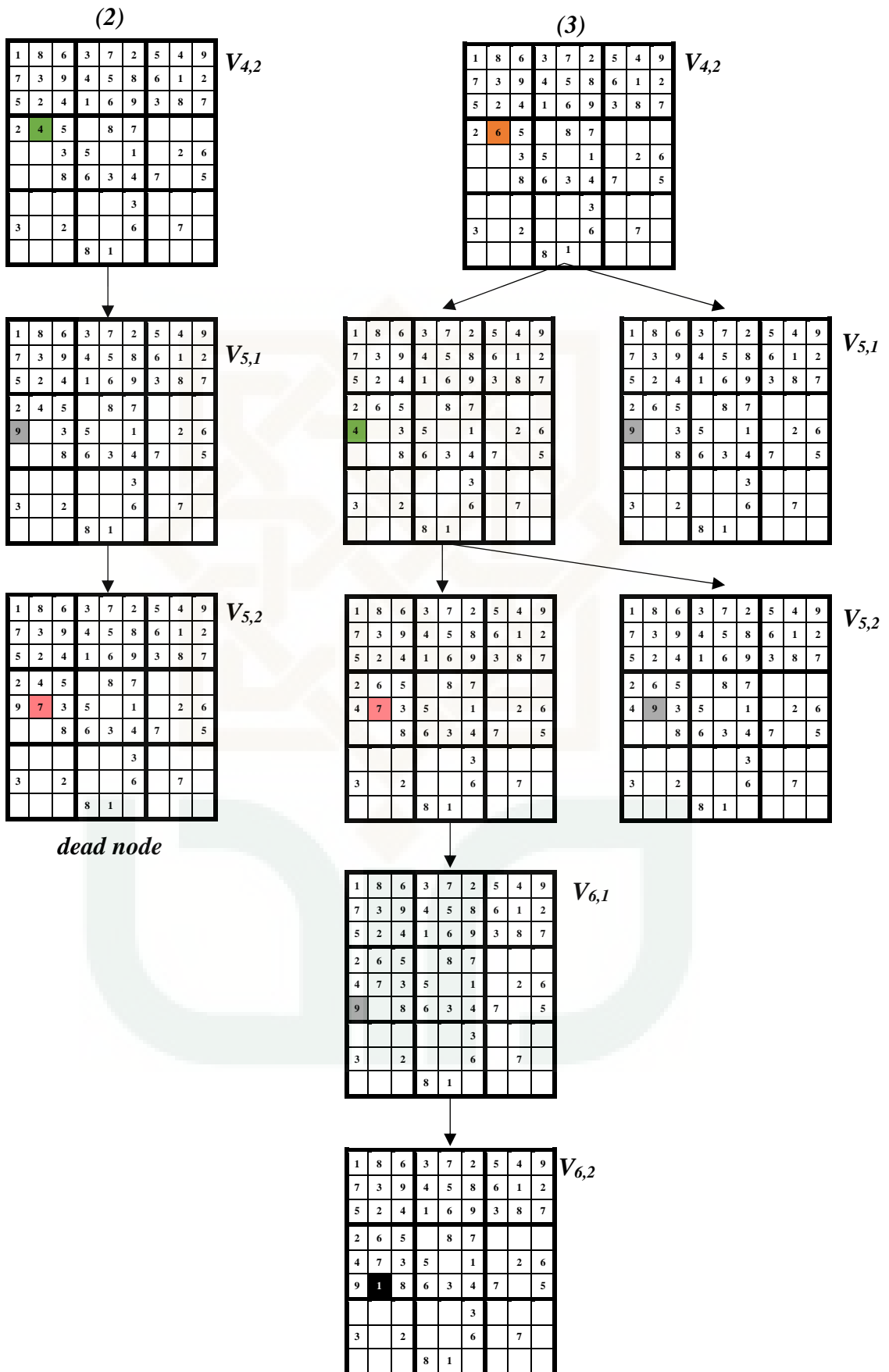


## LAMPIRAN 2

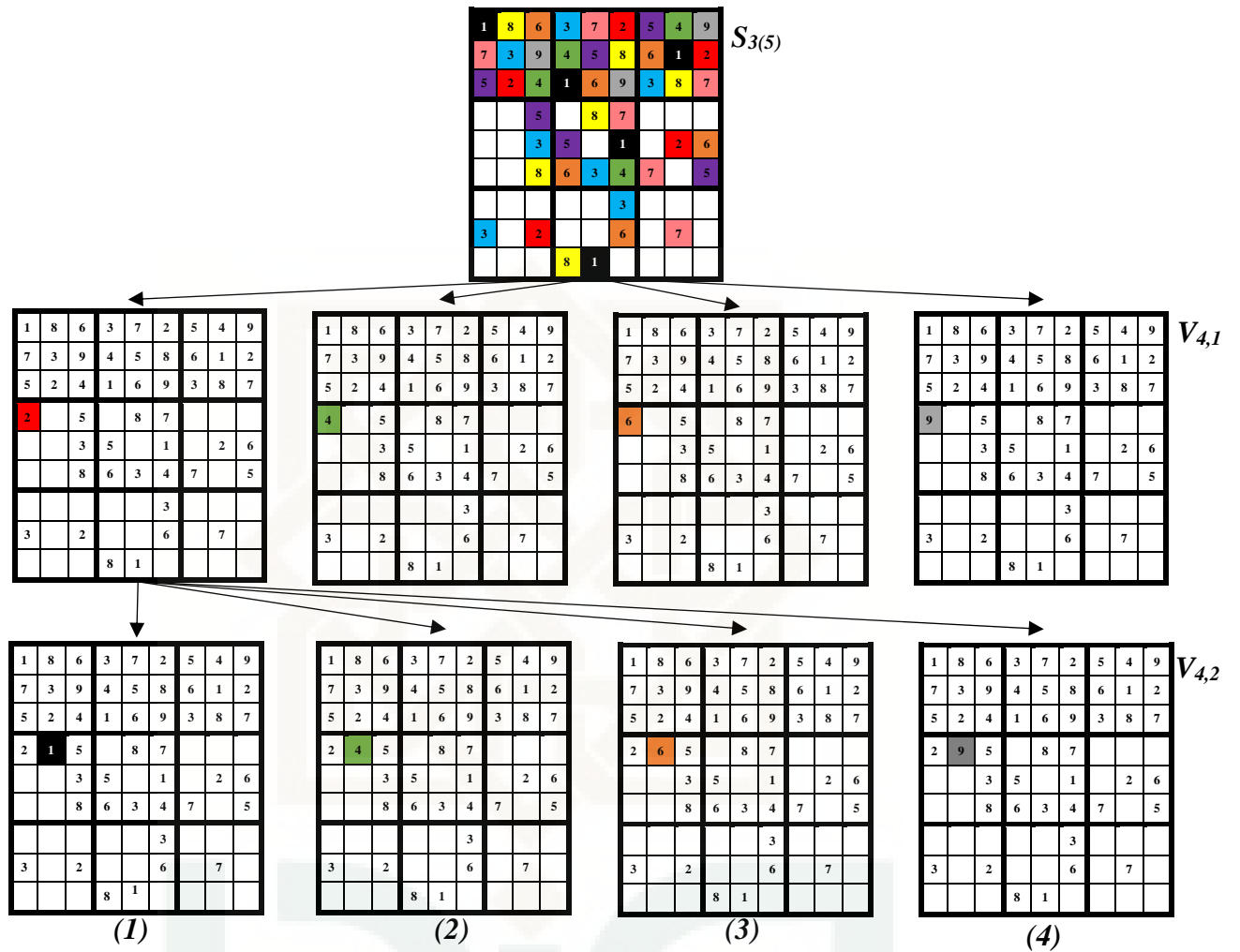
Gambar 3.10 Pohon Solusi  $S_{4(11)}$







Gambar 3.12a Pohon Solusi Backtracking  $S_{4(16)}$





(3)

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
2	6	5		8	7			
		3	5		1		2	6
		8	6	3	4	7		5
					3			
3		2			6		7	
				8	1			

$V_{4,2}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
2	6	5		8	7			
		3	5		1		2	6
		8	6	3	4	7		5
					3			
3		2			6		7	
				8	1			

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
2	6	5		8	7			
9		3	5		1		2	6
		8	6	3	4	7		5
					3			
3		2			6		7	
				8	1			

$V_{5,1}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
2	6	5		8	7			
4	7	3	5		1		2	6
		8	6	3	4	7		5
					3			
3		2			6		7	
				8	1			

*dead node*

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
2	6	5		8	7			
4	9	3	5		1		2	6
		8	6	3	4	7		5
					3			
3		2			6		7	
				8	1			

*dead node*

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
2	6	5		8	7			
9	4	3	5		1		2	6
		8	6	3	4	7		5
					3			
3		2			6		7	
				8	1			

*dead node*

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
2	6	5		8	7			
9	7	3	5		1		2	6
		8	6	3	4	7		5
					3			
3		2			6		7	
				8	1			

$V_{5,2}$

*dead node*

(4)

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
2	9	5		8	7			
		3	5		1		2	6
		8	6	3	4	7		5
					3			
3		2			6		7	
			8	1				

$V_{4,2}$

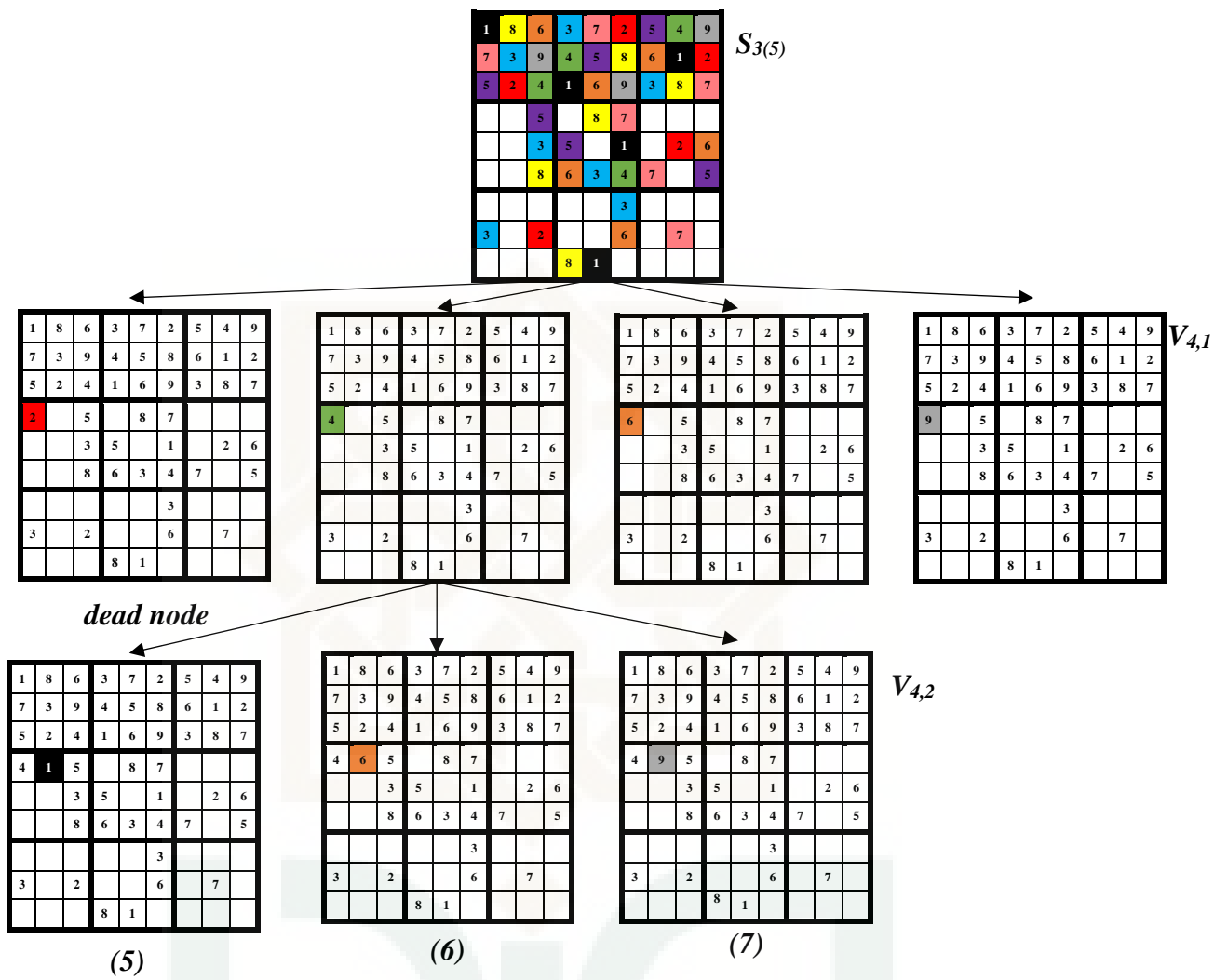
1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
2	9	5		8	7			
4		3	5		1		2	6
		8	6	3	4	7		5
					3			
3		2			6		7	
			8	1				

$V_{5,1}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
2	9	5		8	7			
4	7	3	5		1		2	6
		8	6	3	4	7		5
					3			
3		2			6		7	
			8	1				

$V_{5,2}$

*dead node*

Gambar 3.12b Pohon Solusi *Backtracking*  $S_4(18)$ 

(5)  $V_{4,2}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
4	1	5		8	7			
		3	5		1		2	6
		8	6	3	4	7		5
					3			
3	2			6		7		
			8	1				

$V_{5,1}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
4	1	5		8	7			
9		3	5		1		2	6
		8	6	3	4	7		5
					3			
3	2			6		7		
			8	1				

$V_{5,2}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
4	1	5		8	7			
9	7	3	5		1		2	6
		8	6	3	4	7		5
					3			
3	2			6		7		
			8	1				

$V_{6,1}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
4	1	5		8	7			
9	7	3	5		1		2	6
2		8	6	3	4	7		5
					3			
3	2			6		7		
			8	1				

*dead node*

(6)  $V_{4,2}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
4	6	5		8	7			
		3	5		1		2	6
		8	6	3	4	7		5
					3			
3	2			6		7		
			8	1				

$V_{5,1}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
4	6	5		8	7			
9		3	5		1		2	6
		8	6	3	4	7		5
					3			
3	2			6		7		
			8	1				

$V_{5,2}$

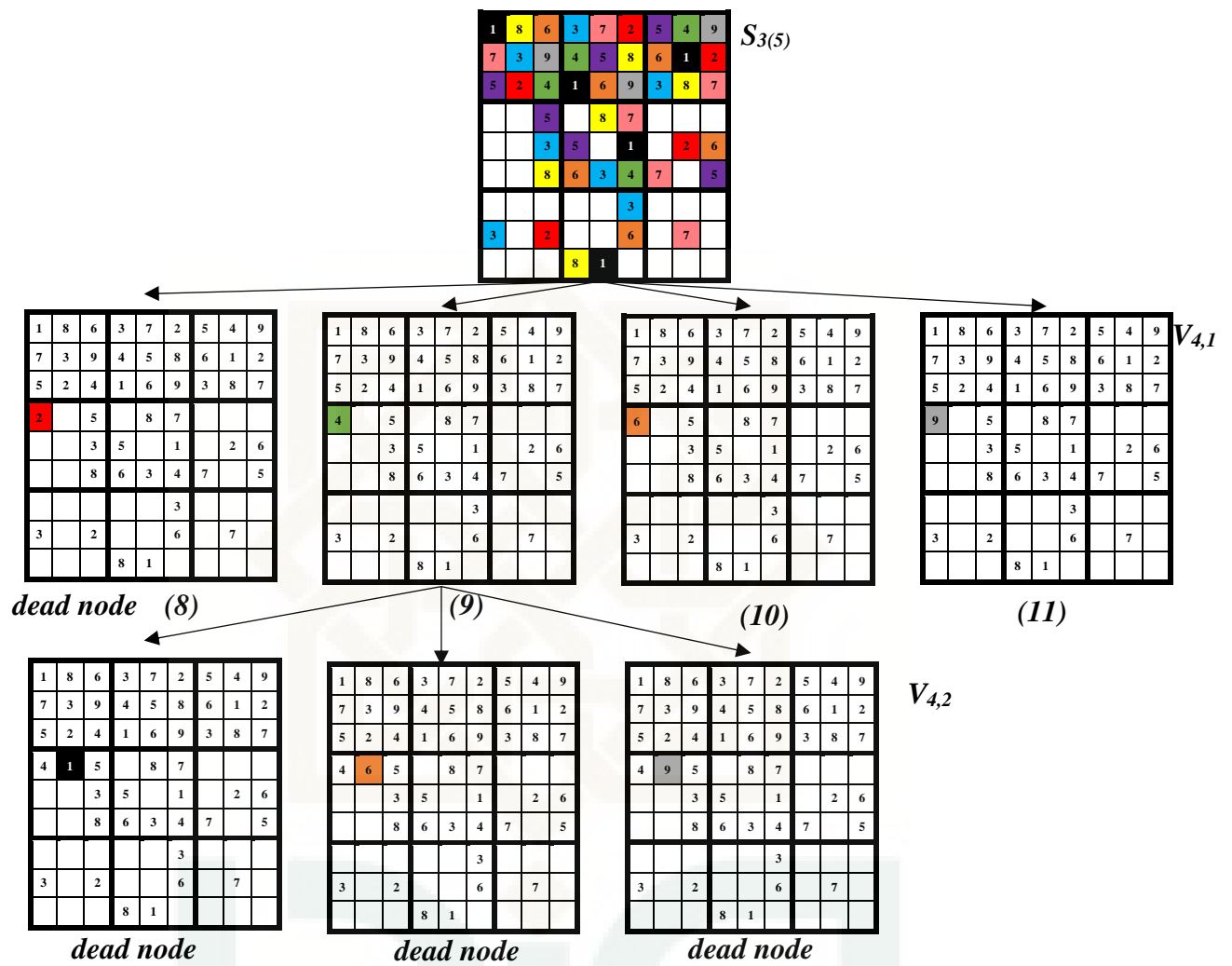
1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
4	6	5		8	7			
9	7	3	5		1		2	6
		8	6	3	4	7		5
					3			
3	2			6		7		
			8	1				

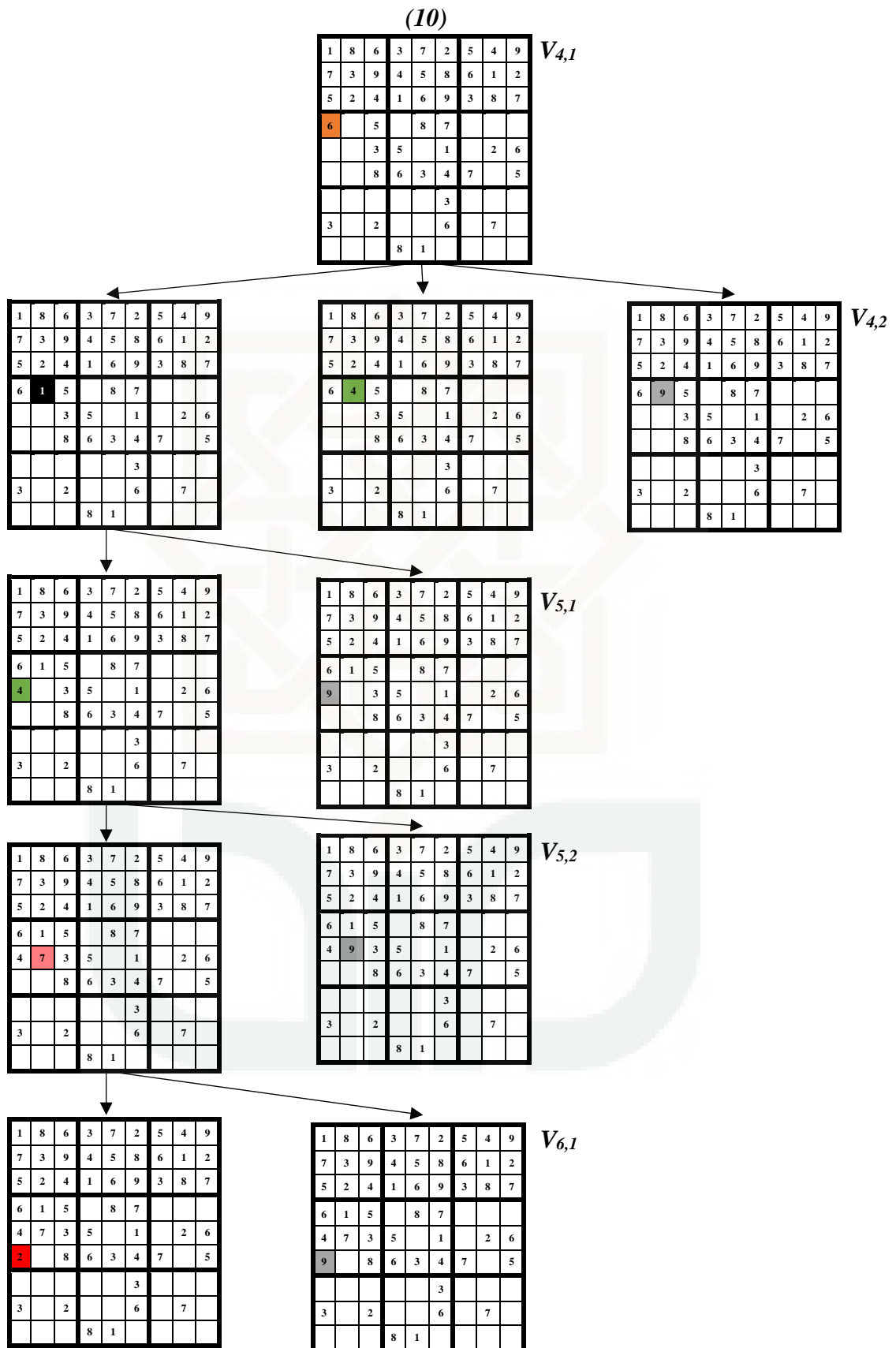
$V_{6,1}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
4	6	5		8	7			
9	7	3	5		1		2	6
2		8	6	3	4	7		5
					3			
3	2			6		7		
			8	1				

$V_{6,2}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
4	6	5		8	7			
9	7	3	5		1		2	6
2	1	8	6	3	4	7		5
					3			
3	2			6		7		
			8	1				

Gambar 3.14 Pohon Solusi Backtracking  $S_{4(21)}$ 



1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	1	5		8	7			
4	7	3	5		1		2	6
2		8	6	3	4	7		5
					3			
3	2				6		7	
			8	1				

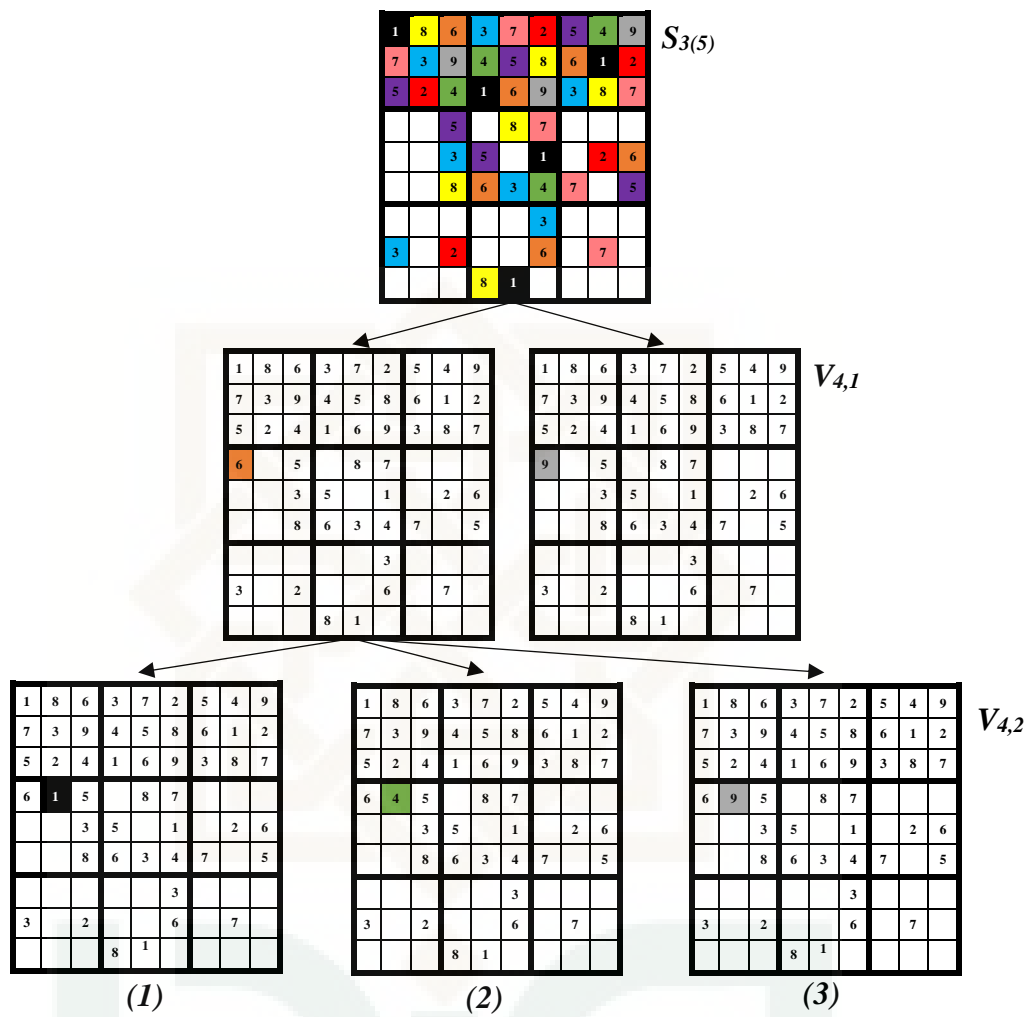
$V_{6,1}$

↓

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	1	5		8	7			
4	7	3	5		1		2	6
2	9	8	6	3	4	7		5
					3			
3	2				6		7	
			8	1				

$V_{6,2}$

Gambar 3.18 Pohon Solusi *Backtracking*  $S_{4(29)}$





(I)

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	1	5		8	7			
		3	5		1		2	6
		8	6	3	4	7		5
					3			
3	2			6	7			
			8	1				

$V_{4,2}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	1	5		8	7			
4		3	5		1		2	6
		8	6	3	4	7		5
					3			
3	2			6	7			
			8	1				

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	1	5		8	7			
9		3	5		1		2	6
		8	6	3	4	7		5
					3			
3	2			6	7			
			8	1				

$V_{5,1}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	1	5		8	7			
4	7	3	5		1		2	6
		8	6	3	4	7		5
					3			
3	2			6	7			
			8	1				

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	1	5		8	7			
4	9	3	5		1		2	6
		8	6	3	4	7		5
					3			
3	2			6	7			
			8	1				

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	1	5		8	7			
9	4	3	5		1		2	6
		8	6	3	4	7		5
					3			
3	2			6	7			
			8	1				

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	1	5		8	7			
9	7	3	5		1		2	6
		8	6	3	4	7		5
					3			
3	2			6	7			
			8	1				

$V_{5,2}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	1	5		8	7			
4	7	3	5		1		2	6
9		8	6	3	4	7		5
					3			
3	2			6	7			
			8	1				

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	1	5		8	7			
4	9	3	5		1		2	6
2		8	6	3	4	7		5
					3			
3	2			6	7			
			8	1				

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	1	5		8	7			
9	4	3	5		1		2	6
2		8	6	3	4	7		5
					3			
3	2			6	7			
			8	1				

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	1	5		8	7			
9	4	3	5		1		2	6
2		8	6	3	4	7		5
					3			
3	2			6	7			
			8	1				

$V_{6,1}$

*dead node*

*dead node*

*dead node*

*dead node*

(2)

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	4	5		8	7			
		3	5		1		2	6
		8	6	3	4	7		5
					3			
3	2				6		7	
			8	1				

$V_{4,2}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	4	5		8	7			
9		3	5		1		2	6
		8	6	3	4	7		5
					3			
3	2				6		7	
			8	1				

$V_{5,1}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	4	5		8	7			
9	7	3	5		1		2	6
		8	6	3	4	7		5
					3			
3	2				6		7	
			8	1				

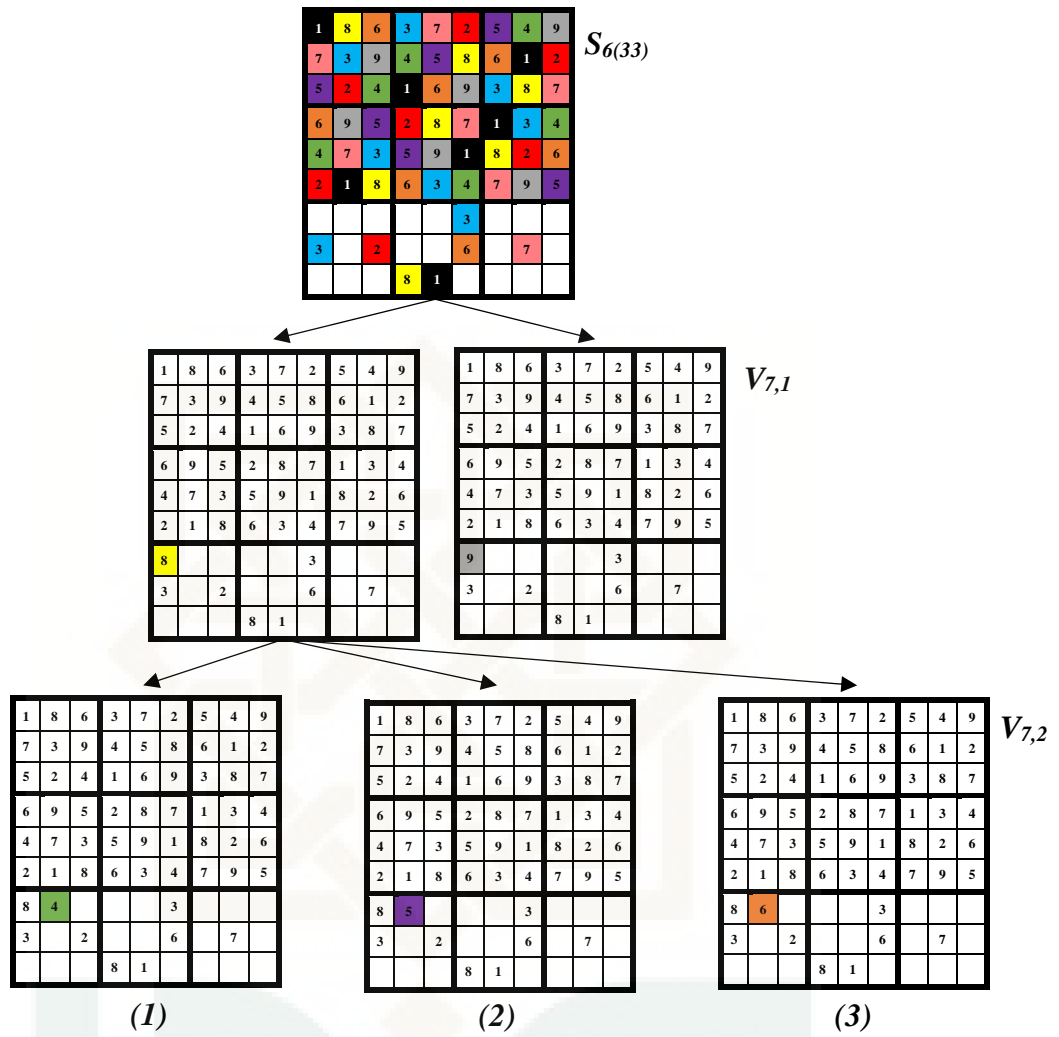
$V_{5,2}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	4	5		8	7			
9	7	3	5		1		2	6
2		8	6	3	4	7		5
					3			
3	2				6		7	
			8	1				

$V_{6,1}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	4	5		8	7			
9	7	3	5		1		2	6
2	1	8	6	3	4	7		5
					3			
3	2				6		7	
			8	1				

$V_{6,2}$

Gambar 3.27 Pohon Solusi *Backtracking*  $S_{7(39)}$ 

(1)

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	1	3	4
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	4			3				
3	2			6			7	
			8	1				

$V_{7,2}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	1	3	4
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	4	7		3				
3	2			6			7	
			8	1				

$V_{7,3}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	1	3	4
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	4	7		3				
3	5	2		6			7	
			8	1				

$V_{8,2}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	1	3	4
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	4	7		3				
3	5	2		6			7	
9			8	1				

$V_{9,1}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	1	3	4
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	4	7		3				
3	5	2		6			7	
9	6		8	1				

$V_{9,2}$

*dead node*

(2)

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	1	3	4
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	5				3			
3		2			6		7	
			8	1				

$V_{7,2}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	1	3	4
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	5	1			3			
3		2			6		7	
			8	1				

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	1	3	4
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	5	7			3			
3		2			6		7	
			8	1				

$V_{7,3}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	1	3	4
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	5	1			3			
3	4	2			6		7	
			8	1				

$V_{8,2}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	1	3	4
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	5	1			3			
3	4	2			6		7	
9			8	1				

$V_{9,1}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	1	3	4
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	5	1			3			
3	4	2			6		7	
9	6		8	1				

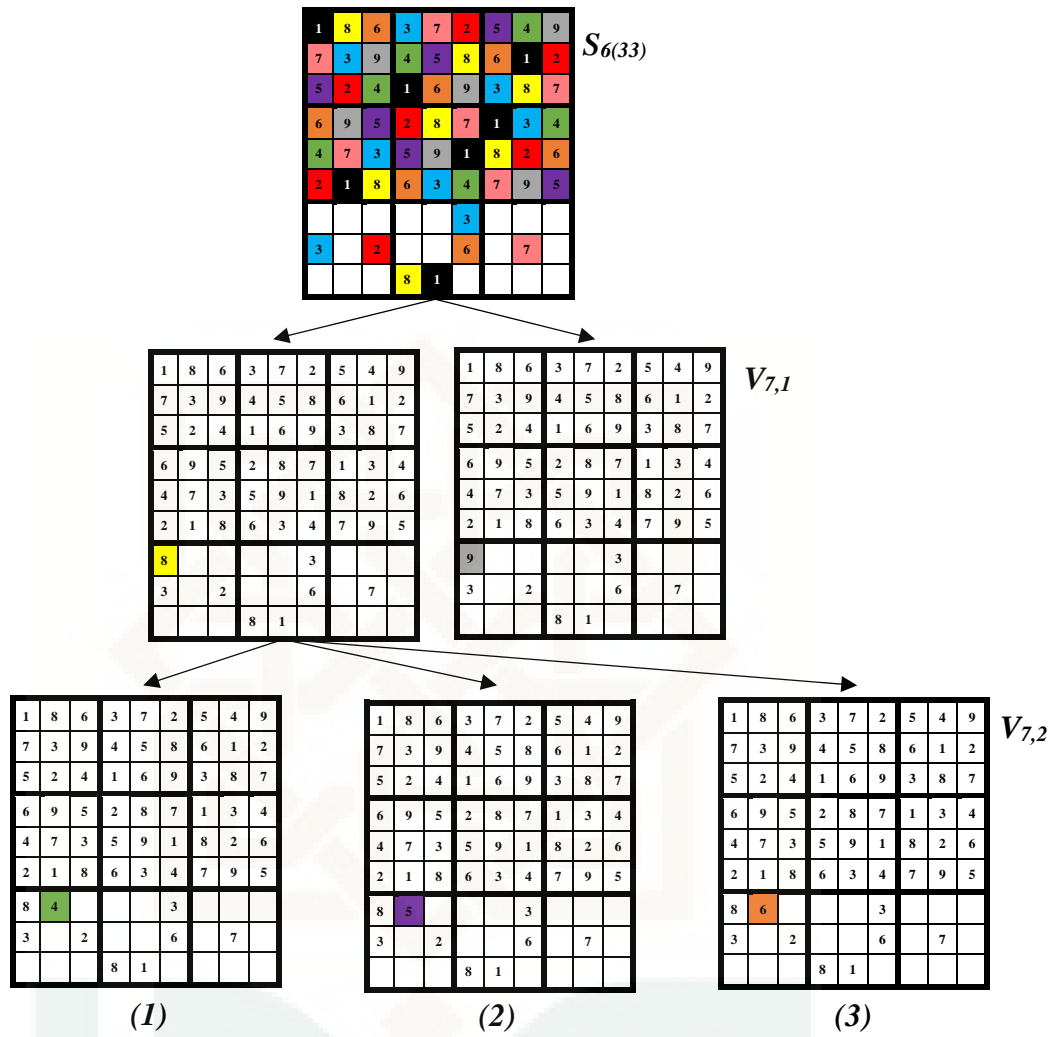
$V_{9,2}$

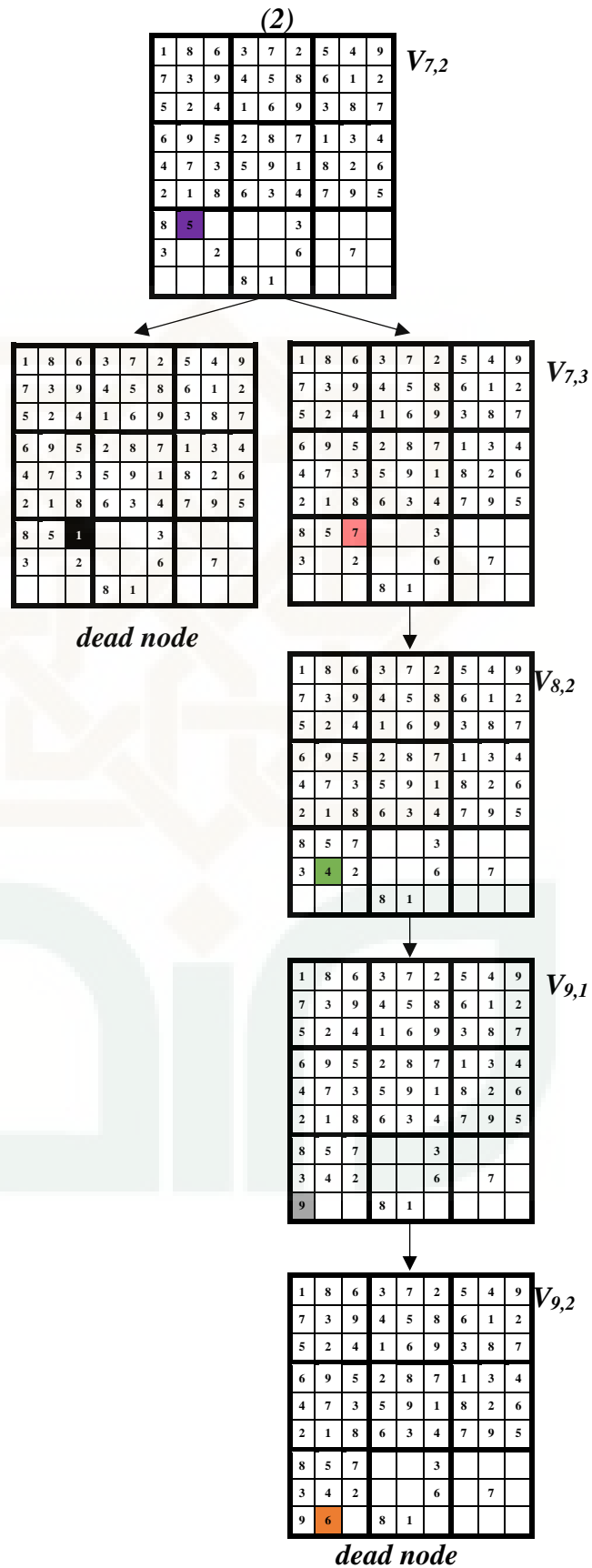
1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	1	3	4
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	5	1			3			
3	4	2			6		7	
9	6		8	1				

 $V_{9,2}$ 

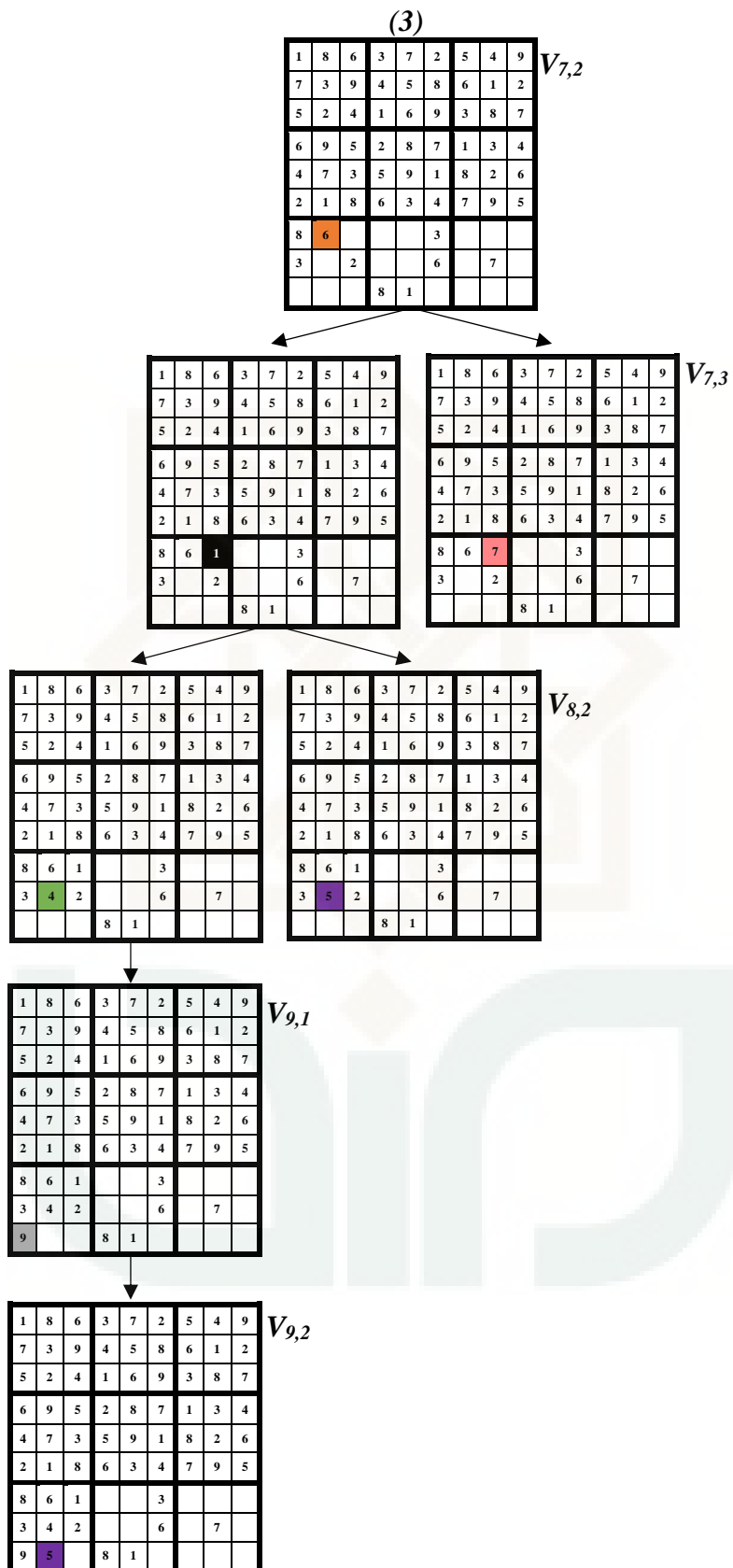
1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	1	3	4
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	5	1			3			
3	4	2			6		7	
9	6	7	8	1				

 $V_{9,3}$

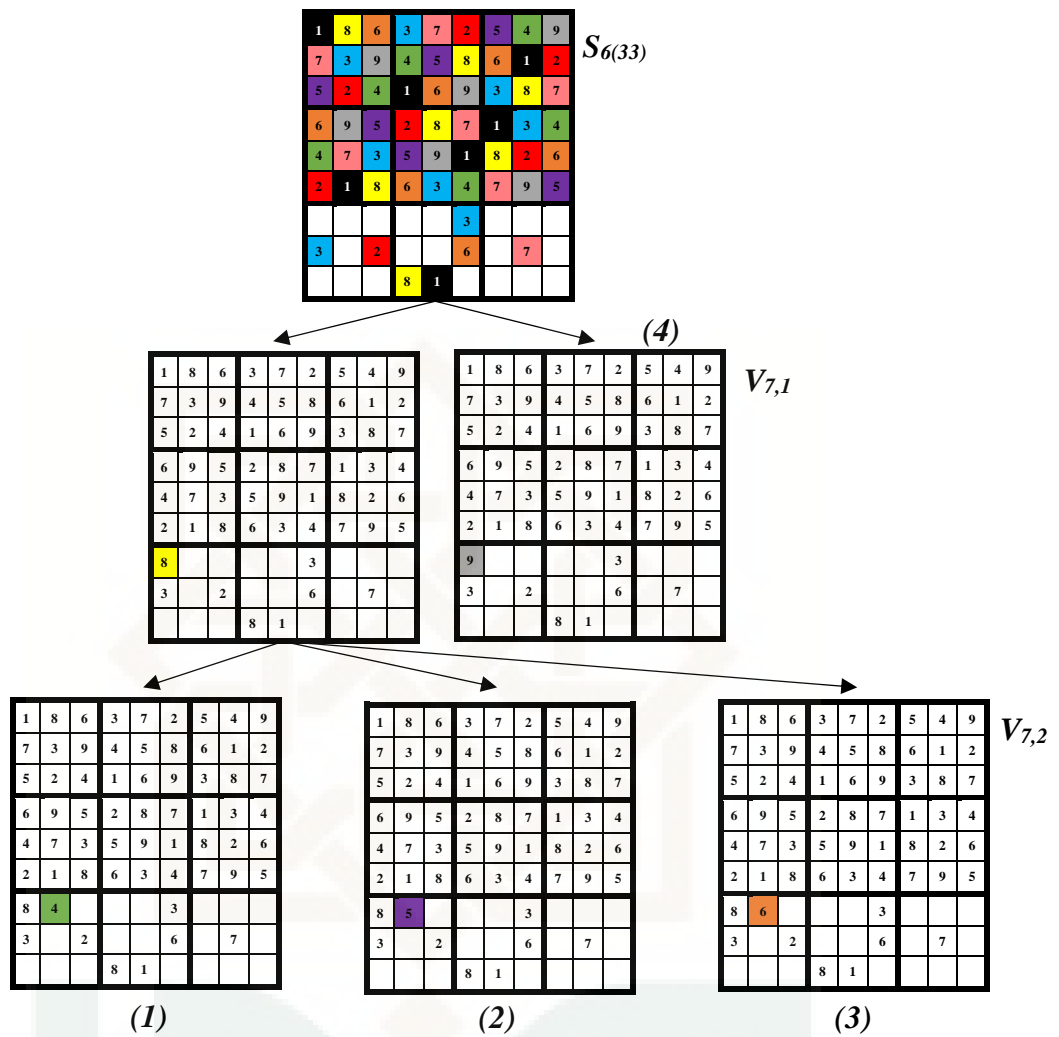
Gambar 3.29b Pohon Solusi *Backtracking*  $S_{7(42)}$ 









Gambar 3.35 Pohon Solusi Backtracking  $S_{7(58)}$ 

(3)

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	1	3	4
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	6			3				
3		2		6		7		
			8	1				

$V_{7,2}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	1	3	4
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	6	1		3				
3		2		6		7		
			8	1				

*dead node*

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	1	3	4
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	6	7		3				
3		2		6		7		
			8	1				

$V_{7,3}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	1	3	4
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	6	7		3				
3	4	2		6		7		
			8	1				

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	1	3	4
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	6	7		3				
3	5	2		6		7		
			8	1				

$V_{8,2}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	1	3	4
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	6	7		3				
3	4	2		6		7		
9			8	1				

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	1	3	4
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	6	7		3				
3	5	2		6		7		
9			8	1				

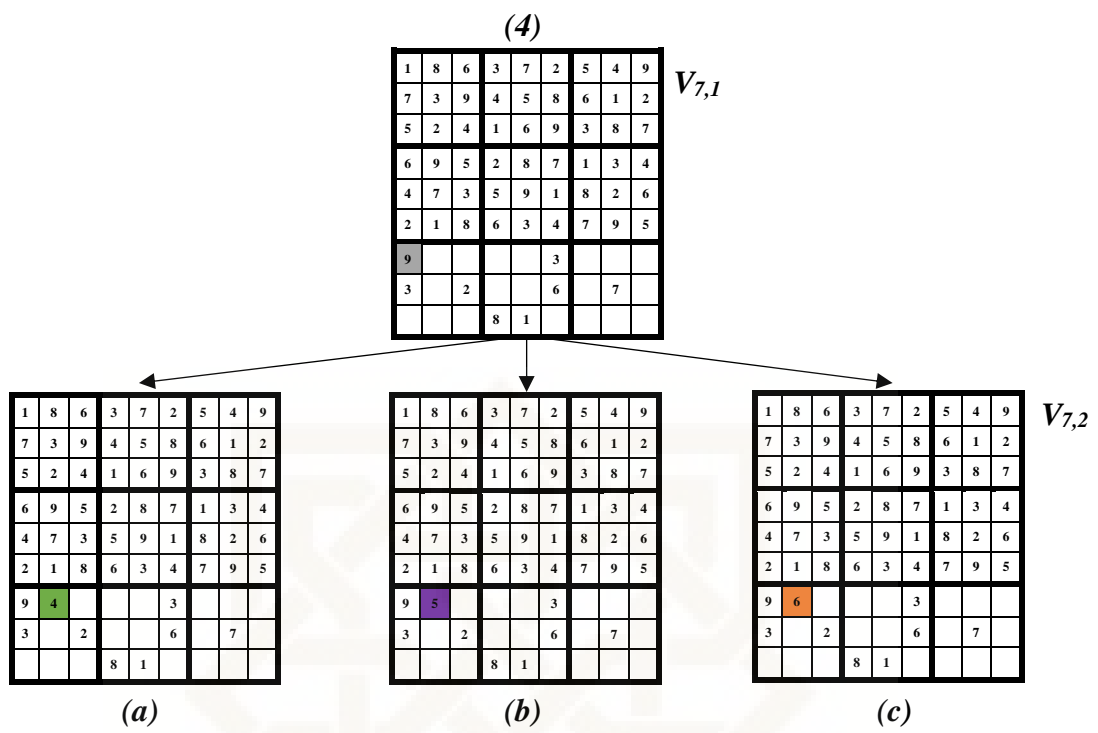
$V_{9,1}$

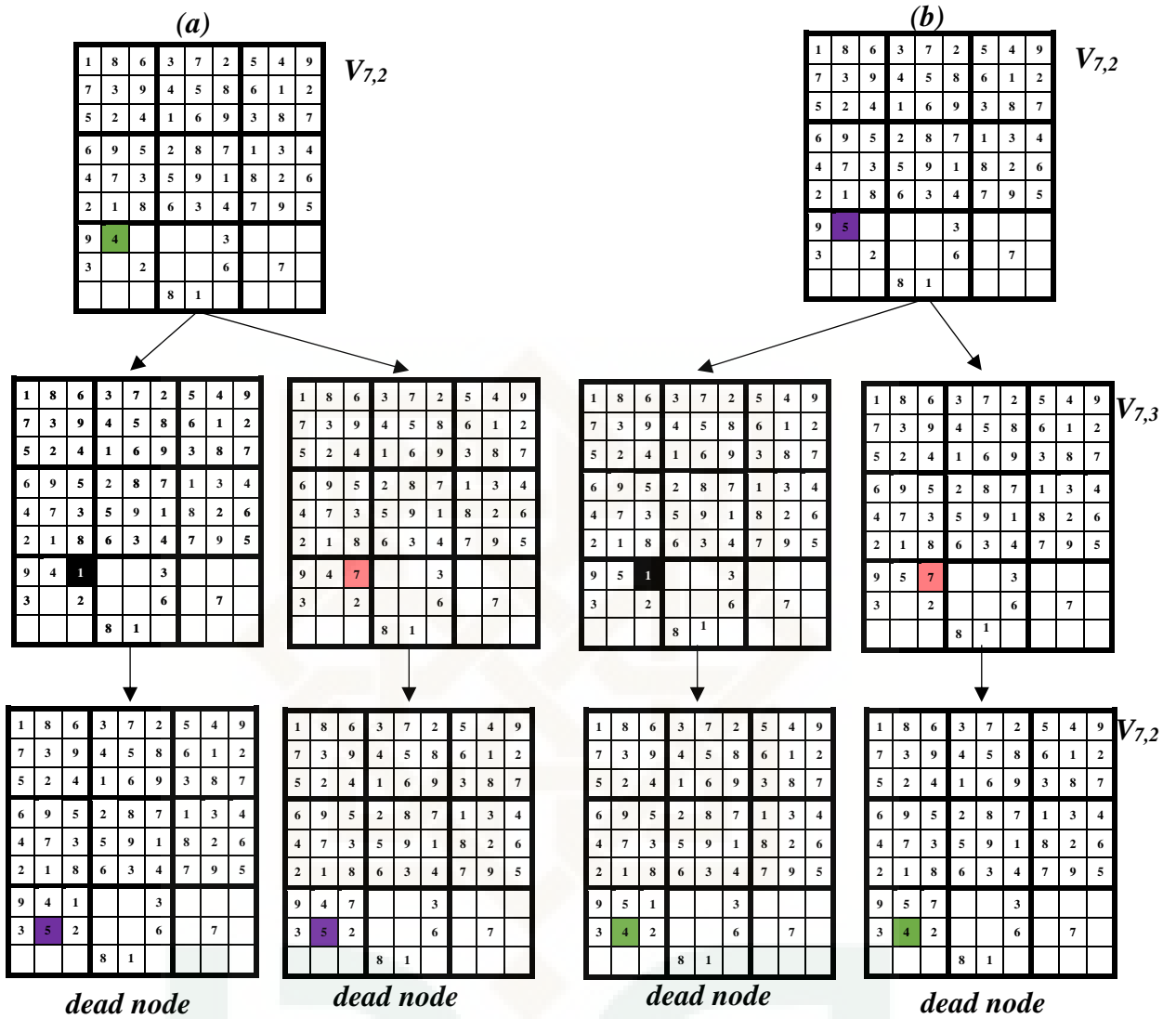
1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	1	3	4
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	6	7		3				
3	4	2		6		7		
9	5		8	1				

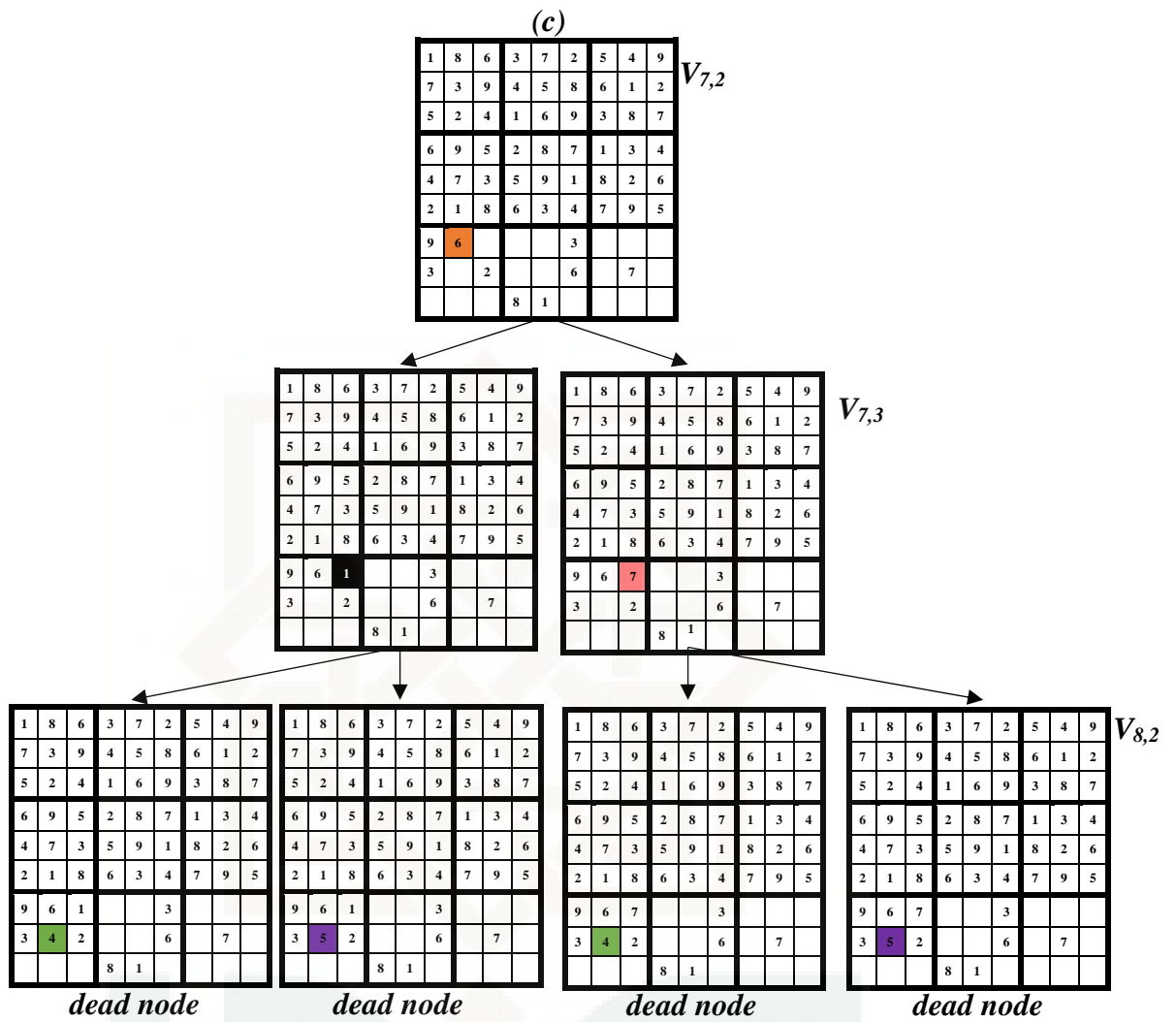
*dead node*

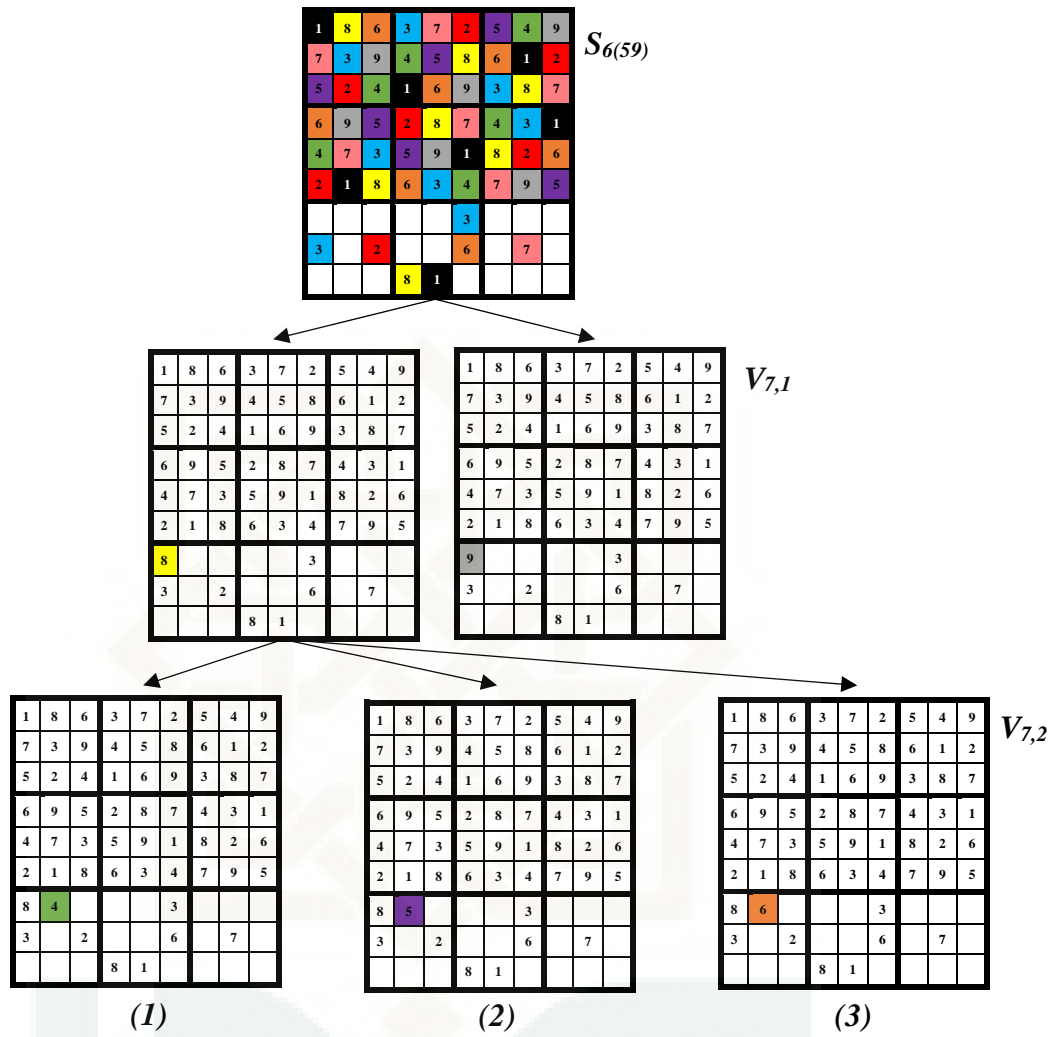
1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	1	3	4
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	6	7		3				
3	5	2		6		7		
4			8	1				

$V_{9,2}$







Gambar 3.41 Pohon Solusi *Backtracking*  $S_{7(65)}$ 



(1)

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	4	3	1
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	4			3				
3	2			6			7	
			8	1				

$V_{7,2}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	4	3	1
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	4	7		3				
3	2			6			7	
			8	1				

$V_{7,3}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	4	3	1
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	4	7		3				
3	5	2		6			7	
			8	1				

$V_{8,2}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	4	3	1
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	4	7		3				
3	5	2		6			7	
9			8	1				

$V_{9,1}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	4	3	1
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	4	7		3				
3	5	2		6			7	
9	6		8	1				

$V_{9,2}$

*dead node*

(2)

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	4	3	1
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	5				3			
3		2			6		7	
			8	1				

$V_{7,2}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	4	3	1
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	5	1			3			
3		2			6		7	
			8	1				

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	4	3	1
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	5	7			3			
3		2			6		7	
			8	1				

$V_{7,3}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	4	3	1
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	5	1			3			
3	4	2			6		7	
			8	1				

$V_{8,2}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	4	3	1
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	5	1			3			
3	4	2			6		7	
9			8	1				

$V_{9,1}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	4	3	1
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	5	1			3			
3	4	2			6		7	
9	6		8	1				

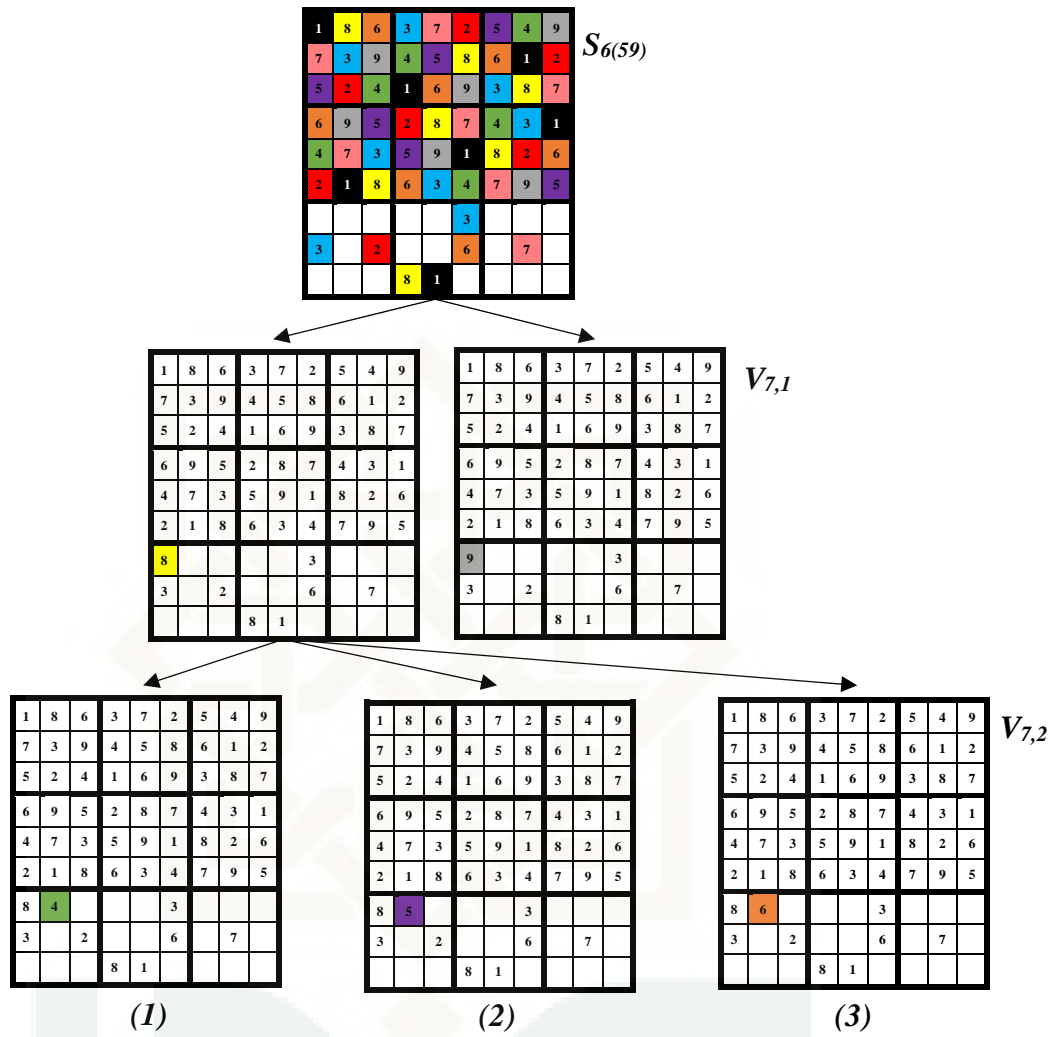
$V_{9,2}$

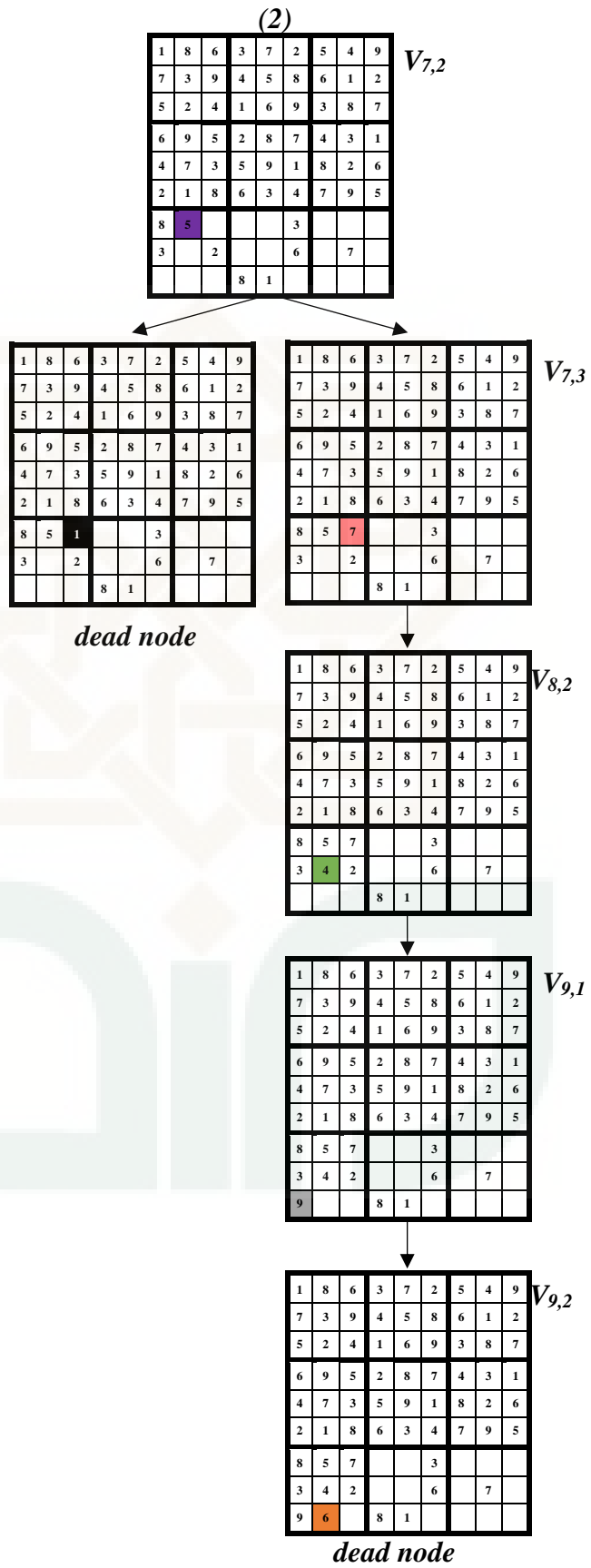
1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	4	3	1
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	5	1			3			
3	4	2			6		7	
9	6		8	1				

$V_{9,2}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	4	3	1
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	5	1			3			
3	4	2			6		7	
9	6	7	8	1				

$V_{9,3}$

Gambar 3.43 Pohon Solusi Backtracking  $S_{7(68)}$ 



(3)

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	4	3	1
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	6				3			
3		2			6		7	
			8	1				

$V_{7,2}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	4	3	1
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	6	1			3			
3		2			6		7	
			8	1				

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	4	3	1
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	6	7			3			
3		2			6		7	
			8	1				

$V_{7,3}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	4	3	1
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	6	1			3			
3	4	2			6		7	
			8	1				

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	4	3	1
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	6	1			3			
3	5	2			6		7	
			8	1				

$V_{8,2}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	4	3	1
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	6	1			3			
3	4	2			6		7	
9			8	1				

$V_{9,1}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	4	3	1
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	6	1			3			
3	4	2			6		7	
9	5		8	1				

$V_{9,2}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	4	3	1
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	6	1			3			
3	4	2			6		7	
9	5		8	1				

$V_{9,2}$

1	8	6	3	7	2	5	4	9
7	3	9	4	5	8	6	1	2
5	2	4	1	6	9	3	8	7
6	9	5	2	8	7	4	3	1
4	7	3	5	9	1	8	2	6
2	1	8	6	3	4	7	9	5
8	6	1			3			
3	4	2			6		7	
9	5	7	8	1				

$V_{9,3}$

Tabel 3.44 Solusi Permainan *Sudoku*

<i>Subgrid</i>	<i>Simpul</i>	<i>Warna</i>
$S_1$	$V_{1,1}$	1
	$V_{2,2}$	3
	$V_{2,3}$	9
	$V_{3,1}$	5
$S_2$	$V_{1,4}$	3
	$V_{1,5}$	7
	$V_{2,4}$	4
	$V_{2,5}$	5
	$V_{2,6}$	8
	$V_{3,4}$	1
	$V_{3,5}$	6
	$V_{3,6}$	9
$S_3$	$V_{1,8}$	4
	$V_{1,9}$	9
	$V_{2,7}$	6
	$V_{2,9}$	2
	$V_{3,7}$	3
	$V_{3,8}$	8
	$V_{3,9}$	7
$S_4$	$V_{4,1}$	6
	$V_{4,2}$	9
	$V_{5,1}$	4
	$V_{5,2}$	7
	$V_{6,1}$	2
	$V_{6,2}$	1
	$S_5$	$V_{4,4}$
$V_{5,5}$		9

<i>Subgrid</i>	<i>Simpul</i>	<i>Warna</i>
$S_6$	$V_{4,7}$	4
	$V_{4,8}$	3
	$V_{4,9}$	1
	$V_{5,7}$	8
	$V_{6,8}$	9
$S_7$	$V_{7,1}$	8
	$V_{7,2}$	6
	$V_{7,3}$	1
	$V_{8,2}$	5
	$V_{9,1}$	9
	$V_{9,2}$	4
	$V_{9,3}$	7
	$S_8$	$V_{7,4}$
$V_{7,5}$		2
$V_{8,4}$		9
$V_{8,5}$		4
$V_{9,6}$		5
$S_9$		$V_{3,6}$
	$V_{7,8}$	5
	$V_{7,9}$	4
	$V_{8,7}$	1
	$V_{8,9}$	8
	$V_{9,7}$	2
	$V_{9,8}$	6
	$V_{9,9}$	3



## LAMPIRAN 3

### Source Code VB.Net

```
Public Class Form1
    Dim temp As String 'buat variabel temp
    'Dim deadnode As New Dictionary(Of String, String)()
    'Dim dipakai As New Dictionary(Of String, String)()
    Dim v As New Dictionary(Of String, String)() 'untuk menyimpan pembatas
    tiap simpul
    'Dim solusiV As New Dictionary(Of String, String)() 'untuk menyimpan
    solusi
    'Dim indexsv As New Dictionary(Of String, Integer)() ' menyimpan index
    fungsi pembatas yang dipakai pada node
    Dim unk(625, 2) As Integer ' urutan node kosong
    Dim node As Integer
    'Dim node2 As Integer
    Dim n As Integer = 3 ' n
    Dim vp(25, 25, 25) As Integer ' simpan variabel pembatas
    Dim isp(25, 25) As Integer ' simpan index variabel pembatas yang sedang
    dipakai
    Dim used(25, 25, 25) As Integer ' simpan variabel pembatas yang sudah
    dipakai

    Private Sub txtBox_KeyPress(ByVal sender As Object, ByVal e As
    System.Windows.Forms.KeyPressEventArgs)
        'hanya bisa input angka
        If Asc(e.KeyChar) <> 8 AndAlso Not IsNumeric(e.KeyChar) Then
            e.Handled = True
        End If
        'Label11.Text = Me.Controls(sender.name).Text
    End Sub

    Private Sub txtBox_TextChanged(sender As System.Object, e As
    System.EventArgs)
        'membatasi nilai input sesuai n
        If n = 2 Then
            If Me.Controls(sender.name).Text.Length > 0 Then
                If Convert.ToInt32(Me.Controls(sender.name).Text) > 4 Or
                Convert.ToInt32(Me.Controls(sender.name).Text) < 1 Then
                    SendKeys.Send("{BackSpace}")
                End If
            End If
        End If
        If n = 3 Then
            If Me.Controls(sender.name).Text.Length > 0 Then
                If Convert.ToInt32(Me.Controls(sender.name).Text) > 9 Or
                Convert.ToInt32(Me.Controls(sender.name).Text) < 1 Then
                    SendKeys.Send("{BackSpace}")
                End If
            End If
        End If
        If n = 4 Then
            If Me.Controls(sender.name).Text.Length > 1 Then
```

```

        If Convert.ToInt32(Me.Controls(sender.name).Text) > 16 Or
Convert.ToInt32(Me.Controls(sender.name).Text) < 1 Then
            SendKeys.Send("{BackSpace}")
        End If
    End If
End If
If n = 5 Then
    If Me.Controls(sender.name).Text.Length > 1 Then
        If Convert.ToInt32(Me.Controls(sender.name).Text) > 25 Or
Convert.ToInt32(Me.Controls(sender.name).Text) < 1 Then
            SendKeys.Send("{BackSpace}")
        End If
    End If
End If
End Sub

Function cek(ByVal x As String, ByVal y As String) 'Cek fungsi pembatas
dengan parameter x dan y
    Dim a As Integer = 1
    v("v" & x & "-" & y) = ""
    ListBox1.Items.Add("v" & x & "-" & y)
    ListBox1.TopIndex = ListBox2.Items.Count - 1
    Dim xx As String
    Dim yy As String
    xx = Math.Floor((x - 1) / n) * n + 1
    yy = Math.Floor((y - 1) / n) * n + 1

    For k = 1 To n * n
        Dim country As Integer = 0
        For l = 1 To n * n
            'ListBox1.Items.Add(xx & yy)
            If Me.Controls("txt" & x & "-" & l).Text = "" AndAlso
Me.Controls("txt" & l & "-" & y).Text = "" Then
                country = country + 1
            End If

            If Me.Controls("txt" & x & "-" & l).Text <> "" Then ' cek
horizontal

                If Me.Controls("txt" & x & "-" & l).Text = k Then
                    temp = ""
                End If
            Else
                temp = k
                'ListBox1.Items.Add(x & l & k)
            End If
        End If
    End If
    If Me.Controls("txt" & l & "-" & y).Text <> "" Then ' cek
vertikal

        If Me.Controls("txt" & l & "-" & y).Text = k Then
            temp = ""
        End If
    Else
        temp = k
        'ListBox1.Items.Add(l & y & k)
    End If
End If
End Sub

```

```

Dim xxx As Integer = Math.Floor((l - 1) / n) + xx
Dim yyy As Integer = ((l - 1) Mod n) + yy
If Me.Controls("txt" & xxx & "-" & yyy).Text <> "" Then
    'MsgBox(xxx & yyy)
    If Me.Controls("txt" & xxx & "-" & yyy).Text = k Then
        temp = ""
        Exit For
    Else
        temp = k
        'MsgBox(xxx & yyy & " " & k)
    End If
End If
End If

Next
If temp <> "" Then
    v("v" & x & "-" & y) = v("v" & x & "-" & y) + temp
    vp(x, y, a) = temp
    a = a + 1
End If
If country = n * n Then
    'ListBox1.Items.Add(country)
    For z = 1 To n * n
        vp(x, y, z) = z
    Next
    a = n * n + 1
    Exit For
End If
End Function

```

Function tentukanpembatas(ByVal g As Integer) 'tentukan simpul kosong per subgrid lalu tentukan fungsi pembatas

```

Dim xx As Integer
Dim yy As Integer
xx = Math.Floor((g - 1) / n) * n + 1
yy = ((g - 1) Mod n) * n + 1
For i = xx To xx + n - 1
    For j = yy To yy + n - 1
        If Me.Controls("txt" & i & "-" & j).Text = "" Then
            cek(i, j) 'panggil fungsi cek
        End If
    Next
Next
End Function

```

Private Sub indexgridkosong() 'mengindeks node kosong di semua subgrid

```

node = 1
Dim xx As Integer
Dim yy As Integer

For i = 1 To n * n
    xx = Math.Floor((i - 1) / n) * n + 1
    yy = ((i - 1) Mod n) * n + 1
    For k = xx To xx + n - 1
        For l = yy To yy + n - 1
            If Me.Controls("txt" & k & "-" & l).Text = "" Then
                unk(node, 0) = k
                unk(node, 1) = l
            End If
        Next
    Next
    node = node + 1
Next

```

```

        'ListBox1.Items.Add(k & "-" & l)
        node = node + 1
    End If
Next
Next
Next
End Sub

```

```

Private Sub selesaikan(sender As Object, e As EventArgs) Handles
eksekusi.Click

```

```

    indexgridkosong()
    solusi()
    MsgBox("Selesai")
End Sub

```

```

Private Sub solusi()
    Dim iterasi As Integer = 0
    Dim subgrid As Integer = cekposisi(unk(1, 0), unk(1, 1))
    tentukanpembatas(cekposisi(unk(1, 0), unk(1, 1)))
    'node2 = node - 1
    'ListBox1.Items.Add("urutan")
    For i = 1 To node - 1
        'tentukanpembatas(cekposisi(unk(i, 0), unk(i, 1)))

        'ListBox2.Items.Add("iterasi : " & iterasi) 'detail iterasi
        If cekposisi(unk(i, 0), unk(i, 1)) > subgrid Then
            subgrid += 1
            tentukanpembatas(cekposisi(unk(i, 0), unk(i, 1)))
            iterasi += 1
            ListBox2.Items.Add("iterasi : " & iterasi)
        End If

        If cekposisi(unk(i, 0), unk(i, 1)) < subgrid Then
            tentukanpembatas(cekposisi(unk(i, 0), unk(i, 1)))
            subgrid -= 1
            'iterasi += 1
            ListBox2.Items.Add("iterasi : " & iterasi)
        End If

        If cekposisi(unk(i, 0), unk(i, 1)) = subgrid Then
            tentukanpembatas(cekposisi(unk(i, 0), unk(i, 1)))
        End If

        If vp(unk(i, 0), unk(i, 1), 0) = isp(unk(i, 0), unk(i, 1)) And
vp(unk(i, 0), unk(i, 1), 0) <> 0 Then
            isp(unk(i, 0), unk(i, 1)) = Nothing
            used(unk(i, 0), unk(i, 1), 0) = 0
            Me.Controls("txt" & unk(i, 0) & "-" & unk(i, 1)).Text = ""
            Me.Controls("txt" & unk(i, 0) & "-" & unk(i, 1)).BackColor
= Color.White
            Me.Controls("txt" & unk(i, 0) & "-" & unk(i, 1)).ForeColor
= Color.Black
            'MsgBox("wow" & unk(i, 0) & unk(i, 1) & " " & vp(unk(i, 0),
unk(i, 1), 0) & " " & i)
            i = i - 2
        Else
            For j = 1 To vp(unk(i, 0), unk(i, 1), 0)

```

```

        If ceksubgrid(unk(i, 0), unk(i, 1), vp(unk(i, 0),
unk(i, 1), j)) = "ketemu" Then
            'MsgBox(urutannodekosong(i, 0) &
urutannodekosong(i, 1) & vp(urutannodekosong(i, 0), urutannodekosong(i, 1),
j))
            Me.Controls("txt" & unk(i, 0) & "-" & unk(i,
1)).Text = ""
            Me.Controls("txt" & unk(i, 0) & "-" & unk(i,
1)).BackColor = Color.White
            Me.Controls("txt" & unk(i, 0) & "-" & unk(i,
1)).ForeColor = Color.Black
        Else
            If j <= used(unk(i, 0), unk(i, 1), 0) Then
                'Exit For
            Else
                Me.Controls("txt" & unk(i, 0) & "-" & unk(i,
1)).Text = vp(unk(i, 0), unk(i, 1), j)
                warna(vp(unk(i, 0), unk(i, 1), j), unk(i, 0) &
"-" & unk(i, 1))
                ListBox2.Items.Add(vp(unk(i, 0), unk(i, 1), j)
& " " & unk(i, 0) & "-" & unk(i, 1))
                isp(unk(i, 0), unk(i, 1)) = j
                used(unk(i, 0), unk(i, 1), 0) += 1
                used(unk(i, 0), unk(i, 1), used(unk(i, 0),
unk(i, 1), 0)) = j
                Exit For
            End If
        End If
    Next
End If
'MsgBox("wow" & unk(i, 0) & unk(i, 1) & " " & vp(unk(i, 0),
unk(i, 1), 0) & " " & i)
If i <> 0 Then
    If Me.Controls("txt" & unk(i, 0) & "-" & unk(i, 1)).Text =
"" Then
        ListBox2.Items.Add("backtrack" & unk(i, 0) & "-" &
unk(i, 1))
        i = i - 2
        iterasi += 1
    End If
End If
Next
End Sub

```

```

Private Sub warna(ByVal val As Integer, ByVal node As String)
    If val = 1 Then
        Me.Controls("txt" & node).BackColor = Color.Black
        Me.Controls("txt" & node).ForeColor = Color.White
    ElseIf val = 2 Then
        Me.Controls("txt" & node).BackColor = Color.Red
        Me.Controls("txt" & node).ForeColor = Color.White
    ElseIf val = 3 Then
        Me.Controls("txt" & node).BackColor = Color.Blue
        Me.Controls("txt" & node).ForeColor = Color.White
    ElseIf val = 4 Then

```

```

        Me.Controls("txt" & node).BackColor = Color.Green
        Me.Controls("txt" & node).ForeColor = Color.White
    ElseIf val = 5 Then
        Me.Controls("txt" & node).BackColor = Color.Purple
        Me.Controls("txt" & node).ForeColor = Color.White
    ElseIf val = 6 Then
        Me.Controls("txt" & node).BackColor = Color.Orange
        Me.Controls("txt" & node).ForeColor = Color.White
    ElseIf val = 7 Then
        Me.Controls("txt" & node).BackColor = Color.DeepPink
        Me.Controls("txt" & node).ForeColor = Color.White
    ElseIf val = 8 Then
        Me.Controls("txt" & node).BackColor = Color.Yellow
        Me.Controls("txt" & node).ForeColor = Color.Black
    ElseIf val = 9 Then
        Me.Controls("txt" & node).BackColor = Color.Gray
        Me.Controls("txt" & node).ForeColor = Color.White
    ElseIf val = 10 Then
        Me.Controls("txt" & node).BackColor = Color.DarkSeaGreen
        Me.Controls("txt" & node).ForeColor = Color.Black
    ElseIf val = 11 Then
        Me.Controls("txt" & node).BackColor = Color.Firebrick
        Me.Controls("txt" & node).ForeColor = Color.White
    ElseIf val = 12 Then
        Me.Controls("txt" & node).BackColor = Color.DarkBlue
        Me.Controls("txt" & node).ForeColor = Color.White
    ElseIf val = 13 Then
        Me.Controls("txt" & node).BackColor = Color.LightSeaGreen
        Me.Controls("txt" & node).ForeColor = Color.Black
    ElseIf val = 14 Then
        Me.Controls("txt" & node).BackColor = Color.LightYellow
        Me.Controls("txt" & node).ForeColor = Color.Black
    ElseIf val = 15 Then
        Me.Controls("txt" & node).BackColor = Color.LightSkyBlue
        Me.Controls("txt" & node).ForeColor = Color.Black
    ElseIf val = 16 Then
        Me.Controls("txt" & node).BackColor = Color.Moccasin
        Me.Controls("txt" & node).ForeColor = Color.Black
    ElseIf val = 17 Then
        Me.Controls("txt" & node).BackColor = Color.PaleVioletRed
        Me.Controls("txt" & node).ForeColor = Color.Black
    ElseIf val = 18 Then
        Me.Controls("txt" & node).BackColor = Color.Peru
        Me.Controls("txt" & node).ForeColor = Color.Black
    ElseIf val = 19 Then
        Me.Controls("txt" & node).BackColor = Color.PowderBlue
        Me.Controls("txt" & node).ForeColor = Color.Black
    ElseIf val = 20 Then
        Me.Controls("txt" & node).BackColor = Color.Salmon
        Me.Controls("txt" & node).ForeColor = Color.Black
    ElseIf val = 21 Then
        Me.Controls("txt" & node).BackColor = Color.SaddleBrown
        Me.Controls("txt" & node).ForeColor = Color.White
    ElseIf val = 22 Then
        Me.Controls("txt" & node).BackColor = Color.Thistle
        Me.Controls("txt" & node).ForeColor = Color.Black
    ElseIf val = 23 Then

```

```

        Me.Controls("txt" & node).BackColor = Color.Wheat
        Me.Controls("txt" & node).ForeColor = Color.Black
    ElseIf val = 24 Then
        Me.Controls("txt" & node).BackColor = Color.Lavender
        Me.Controls("txt" & node).ForeColor = Color.Black
    ElseIf val = 25 Then
        Me.Controls("txt" & node).BackColor = Color.Goldenrod
        Me.Controls("txt" & node).ForeColor = Color.Black
    End If
End Sub

Private Function cekposisi(ByVal x As Integer, ByVal y As Integer)
    Dim posisi As Integer = 0
    Dim xx As Integer
    Dim yy As Integer
    xx = Math.Floor((x - 1) / n) + 1
    yy = Math.Floor((y - 1) / n) + 1
    If xx < 2 Then
        posisi = xx * yy + (n * (xx - 1) - (xx - 1))
    Else
        posisi = xx * yy + (n * (xx - 1) - (xx - 1) - (((yy - 1) * (xx
- 1)) Mod (n * (xx - 1))))
    End If
    Return posisi
End Function

Private Function ceksubgrid(ByVal x As Integer, ByVal y As Integer,
ByVal val As String)

    Dim xx As Integer
    Dim yy As Integer
    xx = Math.Floor((x - 1) / n) * n + 1
    yy = Math.Floor((y - 1) / n) * n + 1
    For i = xx To xx + n - 1
        For j = yy To yy + n - 1
            'ListBox2.Items.Add(i & j)
            If Me.Controls("txt" & i & "-" & j).Text = val Then
                'MsgBox("ketemu" & i & j)
                Return "ketemu"
            End If
        Next
    Next
End Function

Private Sub Button1_Click(sender As Object, e As EventArgs) Handles
Button1.Click ' fungsi bersihkan sudoku
    For Each cntl In Me.Controls
        If TypeOf (cntl) Is System.Windows.Forms.TextBox Then
            cntl.clear()
            cntl.BackColor = Color.White
            cntl.ForeColor = Color.Black
        End If
    Next
    v.Clear()
    ListBox2.Items.Clear()

```

```

        ListBox1.Items.Clear()
        Array.Clear(vp, 0, vp.Length)
        Array.Clear(isp, 0, isp.Length)
        Array.Clear(used, 0, used.Length)
    End Sub

    Private Sub hancur()
        Dim count As Integer = Me.Controls.Count

        For i As Integer = count - 1 To 0 Step -1
            Dim cControl As Control = Me.Controls(i)

            If (TypeOf cControl Is CheckBox OrElse TypeOf cControl Is
                TextBox) Then
                Me.Controls.Remove(cControl)
            End If
        Next
        v.Clear()
        ListBox2.Items.Clear()
        ListBox1.Items.Clear()
        Array.Clear(vp, 0, vp.Length)
        Array.Clear(isp, 0, isp.Length)
        Array.Clear(used, 0, used.Length)
    End Sub

    Public Sub buatBox() 'Fungsi membuat grid n
        Dim txtBox As New System.Windows.Forms.TextBox
        Dim xPos As Integer = 100
        Dim yPos As Integer = 120
        Dim xLen As Integer = xPos
        Dim yLen As Integer = yPos
        Dim fontSize As Double = 15.75
        Dim BoxHeight As Integer = 31
        If n = 5 Then
            BoxHeight = 20
            fontSize = 12
        End If
        Dim BoxGridPosition As Integer = 1
        For i1 = 1 To n * n
            yLen = (Math.Floor((i1 - 1) / n) + 1) * 10 + yPos
            For i2 = 1 To n * n
                txtBox.Name = "txt" & i1 & "-" & i2
                xLen = (Math.Floor((i2 - 1) / n) + 1) * 10 + xPos
                txtBox.Location = New Point((BoxHeight * i2) + xLen,
                    (BoxHeight * i1) + yLen)
                txtBox.Multiline = True
                txtBox.MinimumSize = New Size(BoxHeight, BoxHeight)
                txtBox.Width = (BoxHeight)
                txtBox.Height = (BoxHeight)
                txtBox.Font = New Font("Segoe UI Light", fontSize)
                txtBox.MaxLength = 2
                txtBox.TextAlign = HorizontalAlignment.Center
                txtBox.BorderStyle = BorderStyle.FixedSingle
                Me.Controls.Add(txtBox)
                AddHandler txtBox.KeyPress, AddressOf txtBox_KeyPress
                AddHandler txtBox.TextChanged, AddressOf txtBox_TextChanged
            Next i2
        Next i1
    End Sub

```



```

        txtBox = New System.Windows.Forms.TextBox
    Next
    xLen = 0
Next

End Sub

Private Sub ComboBox1_SelectedIndexChanged(sender As Object, e As
EventArgs) Handles ComboBox1.SelectedIndexChanged
    n = ComboBox1.SelectedItem
    hancur()
    buatBox()
End Sub

Private Sub Bantuan_Click(sender As Object, e As EventArgs) Handles
Bantuan.Click
    MsgBox("CARA KERJA APLIKASI PECAHKAN SUDOKU
1.Pilih n sesuai permasalahan yang akan diselesaikan.
2.Input permasalahan sudoku ke dalam kotak yang telah tersedia.
3.Klik tombol 'Selesaikan' dan akan terlihat hasil dari permasalahan
sudoku.
4.Klik tpmbol 'Detail' untuk menampilkan proses pencarian solusi yang
berupa fungsi pembatas dan iterasi, klik 'tutup' untuk mengembalikan
jendela seperti semula.
5.Klik 'Bersihkan Angka' untuk menghapus permasalahan yang telah diinput
dan dapat diinput kembali untuk permasalahan sudoku yang baru.")
End Sub

Private Sub Detail_Click(sender As Object, e As EventArgs) Handles
Detail.Click 'fungsi detail
    If Detail.Text = "Detail" Then
        Me.Width = 1160
        Detail.Text = "Tutup"
    ElseIf Detail.Text = "Tutup" Then
        Me.Width = 719
        Detail.Text = "Detail"
    End If
End Sub
End Class

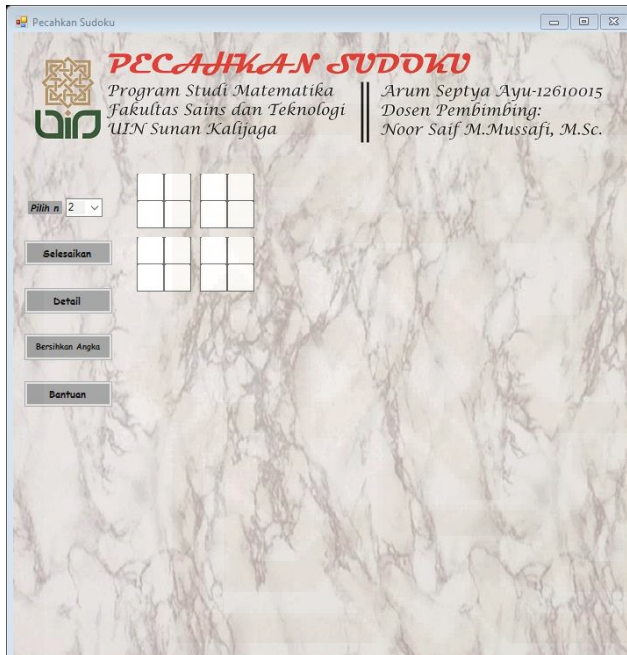
```

## LAMPIRAN 4

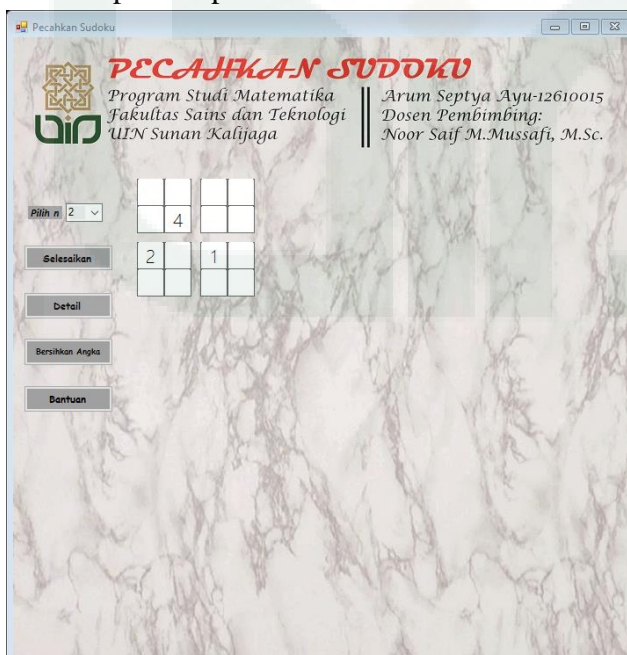
### Output Aplikasi “Pecahkan Sudoku”

#### 1. Untuk $n = 2$

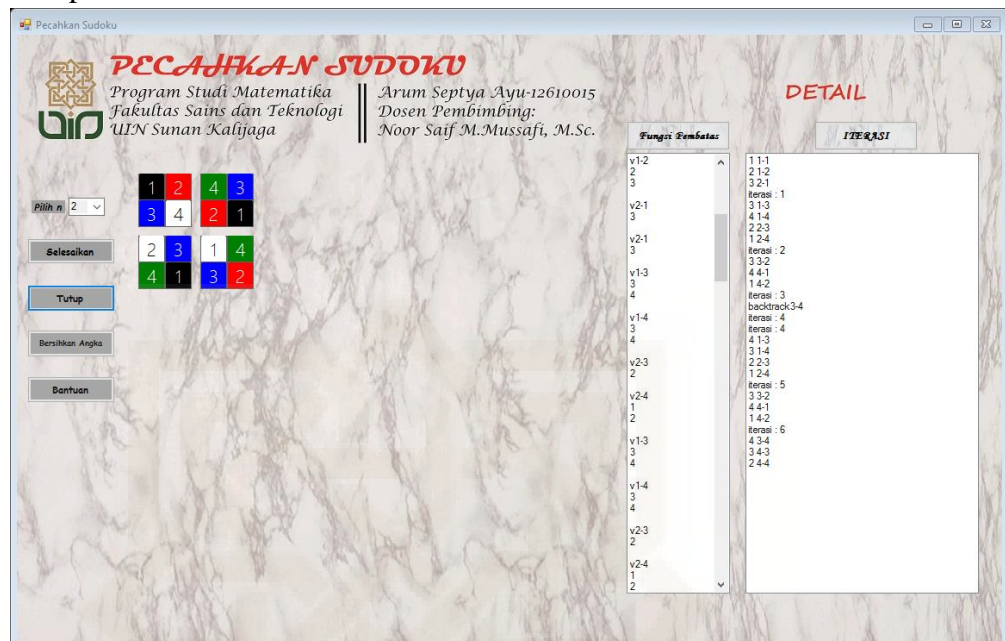
- Tampilan awal



- Tampilan Input

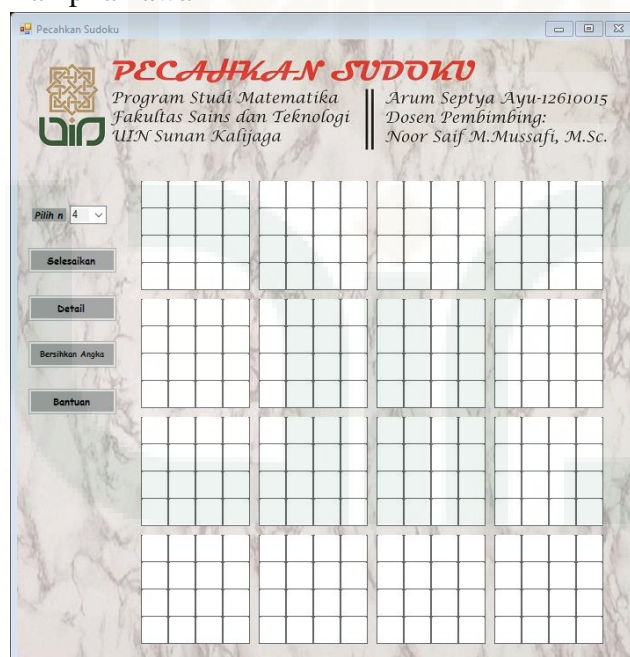


- Tampilan Solusi dan Detail



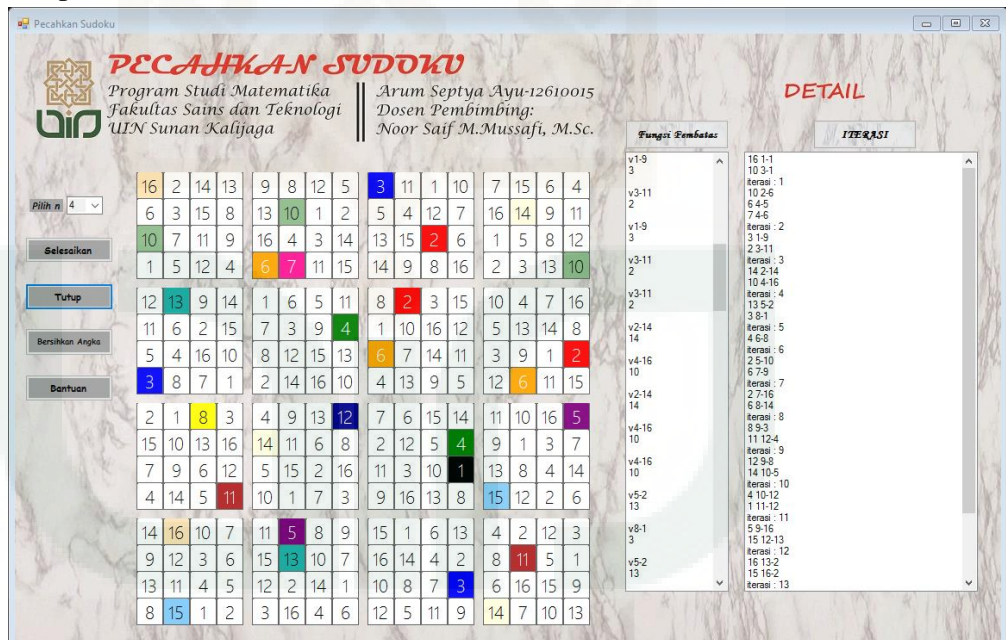
2. Untuk  $n = 4$

- Tampilan awal





- Tampilan Solusi dan Detail

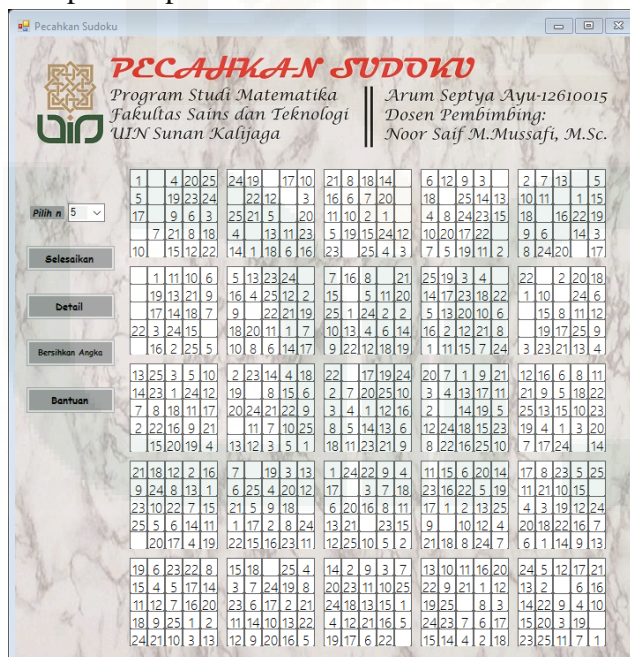


3. Untuk  $n = 5$

- Tampilan awal



- Tampilan Input



- Tampilan Solusi dan Detail

The screenshot displays the 'PECAHKAN SUDOKU' application interface. The main window shows a 15x15 grid with numbers 1-15. The grid is divided into 15 columns and 15 rows. The numbers are arranged in a pattern that satisfies the Sudoku constraints. The interface includes a title bar 'Pecahkan Sudoku', a logo, and text identifying the program as 'Program Studi Matematika Fakultas Sains dan Teknologi UIN Sunan Kalijaga'. The author is 'Arum Septya Ayu-12610015' and the supervisor is 'Noor Saif M. Mussajfi, M.Sc.'. On the left, there are buttons for 'Pilih n', 'Selesaikan', 'Tutup', 'Bersihkan Angka', and 'Bantuan'. On the right, there is a 'DETAIL' panel with two sections: 'Fungsi Pembatas' and 'ITERASI'. The 'Fungsi Pembatas' section lists various constraints like 'v2-6', 'v2-9', etc. The 'ITERASI' section shows a list of iterations and their corresponding values.

**Bantuan**

The screenshot shows a help dialog box titled 'Skrripsi Arum'. It contains the following instructions:

**CARA KERJA APLIKASI PECAHKAN SUDOKU**

1. Pilih n sesuai permasalahan yang akan diselesaikan.
2. Input permasalahan sudoku ke dalam kotak yang telah tersedia.
3. Klik tombol 'Selesaikan' dan akan terlihat hasil dari permasalahan sudoku.
4. Klik tombol 'Detail' untuk menampilkan proses pencarian solusi yang berupa fungsi pembatas dan iterasi, klik 'tutup' untuk mengembalikan jendela seperti semula.
5. Klik 'Bersihkan Angka' untuk menghapus permasalahan yang telah diinput dan dapat diinput kembali untuk permasalahan sudoku yang baru.

An 'OK' button is located at the bottom right of the dialog box.

## DAFTAR RIWAYAT HIDUP



Nama Lengkap : Arum Septya Ayu

Tempat, Tanggal Lahir : Pontianak, 12 September 1994

Alamat Asal : Jalan Pangeran Natakusuma Gang Bambu No 24 A  
Kalimantan Barat

Alamat Sekarang : Jalan Mojo I No 396, Baciro, Yogyakarta

Email : arumseptya@gmail.com

### RIWAYAT PENDIDIKAN:

2012-2016 : Matematika, Fakultas Sains dan Teknologi, UIN  
Sunan Kalijaga Yogyakarta.

2009-2012 : SMAN 3 Pontianak

2006-2009 : MTs N 1 Pontianak

2000-2006 : MIS AL-IKHWAH Pontianak