

PEMODELAN 8-PUZZLE MENGGUNAKAN *INTEGER LINEAR PROGRAMMING*

AGUNG HARGIANTORO



**DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT PERTANIAN BOGOR
BOGOR
2015**



@Hik cipta mitr IPB University

IPB University



IPB University
— *berpola hamparan* —

Hal Cipta (branding) Unmang-urndang

1. Diambil mengutip sebagian atau seluruh karya atau hasil penciptaan, inkuar dan ekspresi ide kreatif :
4. Pengalihan jenis atau bentuk ekspresi seni/dikawatir, awasilah, perusakan karya ilmiah, penyalahgunaan laporan, penyalahgunaan kritik atau tanggapan suatu masalah
5. Pengalihan tidak mengutip ekspresi yang wajar IPB University
2. Dianggap mengunsihkan dan menipertahankan selajun akan seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB University

PERNYATAAN MENGENAI SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA*

Dengan ini saya menyatakan bahwa skripsi berjudul *Pemodelan 8-Puzzle Menggunakan Integer Linear Programming* adalah benar karya saya dengan arahan dari komisi pembimbing dan belum diajukan dalam bentuk apa pun kepada perguruan tinggi mana pun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka di bagian akhir skripsi ini.

Dengan ini saya melimpahkan hak cipta dari karya tulis saya kepada Institut Pertanian Bogor.

Bogor, November 2015

Agung Hargiantoro
NIM G54090039

ABSTRAK

AGUNG HARGIANTORO. *Pemodelan 8-Puzzle Menggunakan Integer Linear Programming*. Dibimbing oleh AMRIL AMAN dan NGAKAN KOMANG KUTHA ARDANA.

8-puzzle adalah salah satu jenis permainan *puzzle*. Permainan ini menggunakan bidang persegi yang dibangun oleh sel 3×3 dengan delapan sel diisi oleh blok yang diberi label 1 sampai dengan 8 dan satu sel kosong. Tugas pemain adalah menyusun delapan blok persegi tersebut sampai membentuk susunan solusi. Dalam karya ilmiah ini akan dibahas bagaimana menentukan susunan awal yang dapat mencapai susunan solusi dan kemudian memformulasikan *8-puzzle* menggunakan *integer linear programming* serta menyelesaikannya dengan *software* LINGO 11.0.

Kata kunci: *8-puzzle, integer linear programming*

ABSTRACT

AGUNG HARGIANTORO. *Modeling 8-Puzzle Using Integer Linear Programming*. Supervised by AMRIL AMAN and NGAKAN KOMANG KUTHA ARDANA.

8-puzzle is a puzzle that uses a square field constructed by 3×3 cells. Its eight cells are filled in by blocks labeled 1 through 8 and the ninth cell is empty. Player should arrange eight square blocks until those eight square blocks forms the solution arrangement. This study presents a characterization of starting arrangement to the required arrangement. The *8-puzzle* is considered as an integer linear programming. It is solved by using LINGO 11.0 computer software.

Keywords: *8-puzzle, integer linear programming*

PEMODELAN 8-PUZZLE MENGGUNAKAN *INTEGER LINEAR PROGRAMMING*

AGUNG HARGIANTORO

Skripsi
sebagai salah satu syarat untuk memperoleh gelar
Sarjana Sains
pada
Departemen Matematika

**DEPARTEMEN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
INSTITUT PERTANIAN BOGOR
2015**



@Hik cipta mitr IPB University

Hik Cipta (Hindone) Unmang-urandeng

1. Diambil mengutip sebagian atau seluruh karya kita yang telah dipublikasikan dan diproses dengan nomor :
 - a. Pengadaan hasil karya untuk kepentingan pendidikan, penelitian, pertukaran karya ilmiah, penyusunan laporan, penulisan kritik atau tinjauan suatu masalah
 - b. Pengalihan tidak merugikan kepentingan yang wajar IPB University.
2. Dianggap mengizinkan dan menyetujui sekalian akan seluruh karya kita di dalam bentuk apapun tanpa izin IPB University.



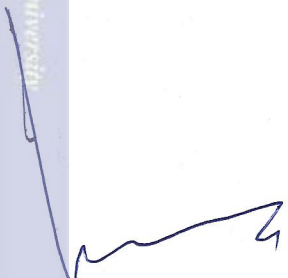
Judul Skripsi: *Pemodelan 8-Puzzle Menggunakan Integer Linear Programming*

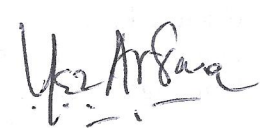
Nama : Agung Hargiantoro

NIM : G54090039

60164 cipta milik IPB University

Disetujui oleh


Dr. Ir. Amril Aman, M.Sc.
Pembimbing I


Ir. N. K. Kutha Ardana, M.Sc.
Pembimbing II

Diketahui oleh




Dr. Toni Bakhtiar, M.Sc.
Ketua Departemen

Tanggal Lulus: **15 DEC 2015**



PRAKATA

Puji syukur kehadiran Allah SWT yang telah memberikan rahmat, karunia dan hidayah-Nya sehingga penulis dapat menyelesaikan penelitian yang berjudul *Pemodelan 8-Puzzle Menggunakan Integer Linear Programming*. Penelitian ini disusun sebagai salah satu syarat kelulusan program sarjana di Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Institut Pertanian Bogor. Oleh karena itu, penulis mengucapkan banyak terima kasih kepada:

1. Allah SWT yang memberikan banyak rahmat dan kasih sayang kepada penulis selaku hamba-Nya,
2. Bapak Ngadimin dan Ibu Warngiatun selaku orang tua tercinta yang selalu memberikan yang terbaik untuk hidup penulis,
3. Dr. Ir. Amril Aman, M.Sc. selaku dosen pembimbing I dan Ir. Ngakan Komang Kutha Ardana, M.Si. selaku dosen pembimbing II yang sudah memberikan banyak dukungan dan arahan sehingga penelitian ini dapat diselesaikan,
4. Dra. Farida Hanum, M.Si selaku dosen penguji atas dukungan dan bimbingan yang diberikan,
5. seluruh dosen dan staf Departemen Matematika untuk semua ilmu, bantuan dan saran yang diberikan,
6. Haryono, Dicky, Ihsan, Adit, Andri, David, Galih, Rangga, Ami, Asin, Syaepul, Parara, Yana, Evy, Sudarsono, Kak Maya, Sevira dan teman-teman di Departemen Matematika lainnya,
7. teman seperjuangan bimbingan: Elysa, Suzi, Irfan, Kak Gita, Kak Zaenal, Eric, Alin, Pupu,
8. teman seperjuangan: Mufqi, Rezza, Eprut, Syarif, Abud, Adul, Yosa, Fatih, Furi dan seluruh teman penulis di Institut Pertanian Bogor.,
9. dan semua pihak yang secara langsung atau tidak langsung membantu penulis dalam menyelesaikan karya ilmiah ini.

Akhir kata, dengan adanya tulisan ini diharapkan dapat memberikan manfaat yang besar. Kritik dan saran yang membangun sangat penulis harapkan untuk kemajuan penelitian ini. Semoga Allah SWT senantiasa melimpahkan rahmat dan karunia-Nya untuk kita semua. Amin.

Bogor, November 2015

Agung Hargiantoro

DAFTAR ISI

DAFTAR TABEL	vi
DAFTAR GAMBAR	vi
DAFTAR LAMPIRAN	vi
PENDAHULUAN	1
Latar Belakang	1
Tujuan Penelitian	1
LANDASAN TEORI	2
Pemrograman Linear	2
Pemrograman Linear <i>Integer</i>	3
Permutasi	3
PEMODELAN 8-PUZZLE	4
Keberadaan Solusi Suatu Susunan Awal	4
Model Penyelesaian Solusi	9
APLIKASI MODEL	16
SIMPULAN DAN SARAN	20
Simpulan	20
Saran	20
DAFTAR PUSTAKA	21
LAMPIRAN	22
RIWAYAT HIDUP	76

DAFTAR TABEL

1	Solusi Model 1 untuk susunan awal <i>8-puzzle</i> pada Gambar 9	18
2	Solusi Model 2 untuk susunan <i>8-puzzle</i> pada Gambar 10	19

DAFTAR GAMBAR

1	Blok-blok pada <i>8-puzzle</i>	4
2	Penomoran sel-sel pada <i>8-puzzle</i>	4
3	Ringkasan dari semua kemungkinan permutasi dari slot dicapai dengan memindahkan blok kosong	5
4	Susunan awal <i>8-puzzle</i> dengan posisi blok kosong sama dengan posisi blok kosong susunan solusi	7
5	Susunan awal <i>8-puzzle</i> dengan posisi blok kosong berjarak n blok dari posisi blok kosong susunan solusi	7
6	Susunan awal <i>8-puzzle</i>	8
7	Ilustrasi variabel-variabel keputusan pada bidang <i>8-puzzle</i>	10
8	Penghilangan beberapa variabel pada Model 1	14
9	Salah satu susunan tersulit <i>8-puzzle</i>	16
10	Susunan <i>8-puzzle</i> hasil solusi Model 1	18
11	Susunan solusi yang didapat dari susunan <i>8-puzzle</i> pada Gambar 9 melalui Model 1 dan Model 2	19

DAFTAR LAMPIRAN

1	Formulasi dan solusi model untuk memeriksa bisa tidaknya suatu susunan awal (pada Gambar 9) mencapai susunan solusi <i>8-puzzle</i> pada LINGO 11.0	22
2	Formulasi dan solusi Model 1 untuk suatu susunan awal (pada Gambar 9) menggunakan LINGO 11.0	26
3	Ilustrasi perubahan susunan berdasarkan solusi Model 1	44
4	Formulasi dan solusi Model 2 untuk suatu susunan solusi <i>8-puzzle</i> dari solusi Model 1 (Gambar 10) menggunakan LINGO 11.0	46
5	Ilustrasi perubahan <i>puzzle</i> berdasarkan solusi Model 2	55
6	Susunan tersulit <i>puzzle</i> 2×3 dapat dicapai dengan 21 langkah dari susunan solusi	56

PENDAHULUAN

Latar Belakang

8-puzzle adalah *puzzle* versi ukuran yang lebih kecil dari *15-puzzle*, yaitu salah satu jenis permainan teka-teki yang ditemukan oleh Noyes Palmer Chapman yang merupakan seorang kepala kantor pos di Canastota, New York pada awal 1874. Permainan ini bertujuan untuk menyusun blok-blok yang memiliki angka 1 sampai 8 dalam suatu bidang persegi yang memiliki ruang untuk 9 blok dengan satu ruang kosong yang tersedia. Permainan dilakukan dengan menggeser salah satu blok ke ruang kosong yang tersedia sehingga mengubah urutan angka-angka yang terdapat pada blok sampai pada akhirnya urutannya sesuai dengan yang diharapkan (angka-angkanya berurutan dari 1 sampai 8 dengan ruang kosong di pojok kanan bawah).

Pada tahun 1870-an Sam Loyd menimbulkan kegemparan di Amerika Serikat, Inggris dan Eropa dengan menawarkan hadiah \$ 1000 untuk siapa saja yang dapat menyelesaikan *puzzle* miliknya, yaitu *15-puzzle* dengan angka berurutan 1 sampai 13 tetapi angka 14 dan 15 ditukar. Banyak yang telah meneliti *puzzle* tersebut, salah satunya Johnson & Story (1879) yang membuktikan bahwa tidak ada solusi untuk *puzzle* Sam Loyd tersebut (Archer 1999). Hal ini mendorong banyak orang untuk meneliti berbagai hal mengenai *15-puzzle* dengan berbagai metode. Salah satu hal menarik adalah bahwa adanya kondisi awal tertentu pada *puzzle* ini yang tidak memiliki solusi dan salah satunya dijelaskan oleh Aaron F. Archer dalam *A Modern Treatment of 15-Puzzle* dan tentunya pada *8-puzzle* juga terdapat suatu susunan awal yang tidak dapat mencapai susunan solusi.

Dalam makalah ini akan digunakan *integer linear programming* untuk diterapkan pada permainan *8-puzzle* sehingga dapat ditentukan langkah-langkah yang akan diambil sehingga *puzzle* ini dapat diselesaikan.

Tujuan Penelitian

Adapun tujuan karya ilmiah ini yaitu:

- Menentukan bisa atau tidaknya suatu susunan awal mencapai susunan solusi dan membuat model untuk memeriksanya;
- membuat suatu formulasi *integer linear programming* yang dapat menentukan pergerakan yang harus diambil untuk mencapai susunan solusi dari *8-puzzle*.

LANDASAN TEORI

Pemrograman Linear

Konsep dasar yang harus dipahami sebelum mendefinisikan pemrograman linear adalah fungsi linear, persamaan linear dan pertaksamaan linear.

Definisi 1 (Fungsi Linear)

Sebuah fungsi $f(x_1, x_2, \dots, x_n)$ dalam suatu variabel-variabel x_1, x_2, \dots, x_n adalah suatu fungsi linear jika dan hanya jika untuk suatu himpunan konstanta c_1, c_2, \dots, c_n , fungsi f dapat dituliskan sebagai $f(x_1, x_2, \dots, x_n) = c_1x_1 + c_2x_2 + \dots + c_nx_n$ (Winston 2004).

Definisi 2 (Persamaan dan Pertaksamaan Linear)

Untuk sembarang fungsi linear $f(x_1, x_2, \dots, x_n)$ dan sembarang bilangan b , suatu persamaan $f(x_1, x_2, \dots, x_n) = b$ merupakan persamaan linear sedangkan pertidaksamaan $f(x_1, x_2, \dots, x_n) \geq b$ dan $f(x_1, x_2, \dots, x_n) \leq b$ merupakan pertidaksamaan linear (Winston 2004).

Definisi 3 (Pemrograman Linear)

Menurut Winston (2004) pemrograman linear (PL) adalah suatu masalah optimasi yang memenuhi ketentuan-ketentuan sebagai berikut:

- tujuan masalah tersebut adalah memaksimumkan atau meminimumkan suatu fungsi linear dari sejumlah variabel keputusan. Fungsi yang akan dimaksimumkan atau diminimumkan ini disebut fungsi objektif.
- nilai variabel-variabel keputusannya harus memenuhi suatu himpunan kendala. Setiap kendala harus berupa persamaan linear atau pertidaksamaan linear,
- ada pembatasan tanda untuk setiap variabel dalam masalah ini. Untuk sembarang variabel x_i , pembatasan tanda menentukan x_i harus taknegatif ($x_i \geq 0$) atau tidak dibatasi tandanya (*unrestricted in sign*).

Bentuk umum pemrograman linear adalah:

memaksimumkan atau meminimumkan

$$z = f(x_1, x_2, \dots, x_n)$$

terhadap kendala:

$$\begin{aligned} g_1(x_1, x_2, \dots, x_n) (\leq, =, \geq) b_1, \\ g_2(x_1, x_2, \dots, x_n) (\leq, =, \geq) b_2, \\ \vdots \\ g_m(x_1, x_2, \dots, x_n) (\leq, =, \geq) b_m, \end{aligned} \quad (1)$$

dengan $f(x_1, x_2, \dots, x_n)$ merupakan fungsi objektif dari pemrograman linear dan $g_i(x_1, x_2, \dots, x_n) (\leq, =, \geq) b_i$ merupakan kendala pemrograman linear dengan $g_i(x_1, x_2, \dots, x_n)$ merupakan fungsi linear (Winston 2004).

Definisi 4 (Daerah Fisibel Pemrograman Linear)

Daerah fisibel untuk pemrograman linear (1) adalah himpunan dari nilai-nilai (x_1, x_2, \dots, x_n) yang memenuhi sejumlah m kendala di (1). Sebuah nilai di

dalam daerah fisibel adalah nilai fisibel dan sebuah nilai di luar daerah fisibel disebut nilai takfisibel (Winston 2004).

Definisi 5 (Nilai Optimum Pemrograman Linear)

Untuk masalah maksimisasi, nilai optimum dari suatu persamaan linear merupakan suatu nilai yang berada dalam daerah fisibel dengan nilai fungsi objektif yang terbesar. Untuk masalah minimisasi, nilai optimum dari suatu persamaan linear merupakan suatu nilai yang berada dalam daerah fisibel dengan nilai fungsi objektif yang terkecil (Winston 2004).

Pemrograman Linear *Integer*

Pemrograman linear *integer* atau disebut juga *integer programming* (IP) merupakan suatu pemrograman linear yang sebagian atau semua variabel yang digunakan merupakan *integer* yang taknegatif. Jika suatu IP menggunakan semua variabel yang berupa *integer*, maka IP tersebut disebut *pure integer programming* (PIP). Jika suatu IP menggunakan sebagian variabel saja yang berupa *integer*, maka IP tersebut disebut *mixed integer programming* (MIP). Jika suatu IP menggunakan semua variabel yang bernilai 0 atau 1, maka IP tersebut disebut 0-1 IP (Winston 2004).

Permutasi

Definisi 6 (Permutasi)

Permutasi dari himpunan takkosong S didefinisikan sebagai pemetaan satu-satu dari S ke S (Durbin 1992).

Definisi 7 (Grup Simetri)

Himpunan semua permutasi dari himpunan takkosong S adalah grup berkaitan dengan komposisi, grup ini disebut grup simetri pada S dan akan dinotasikan $Sym(S)$. Ketika S adalah himpunan $\{1, 2, \dots, n\}$ yang terdiri dari n bilangan bulat positif pertama, grup $Sym(S)$ biasanya dinotasikan S_n (Durbin 1992).

Definisi 8 (Cycle)

Unsur S_n sering ditulis menggunakan notasi *cycle*. Jika S adalah suatu himpunan dan $\alpha_1, \alpha_2, \dots, \alpha_k \in S$, maka $(\alpha_1 \alpha_2 \dots \alpha_k)$ menunjukkan permutasi dari S yang

$$\alpha_1 \rightarrow \alpha_2, \alpha_2 \rightarrow \alpha_3, \dots, \alpha_{k-1} \rightarrow \alpha_k, \alpha_k \rightarrow \alpha_1$$

dan

$$x \rightarrow x \text{ untuk semua } x \in S \text{ lainnya.}$$

Sebuah permutasi tersebut disebut *cycle* atau *k-cycle* (Durbin 1992).

Definisi 9 (Transposisi)

Transposisi adalah 2-cycle dalam S_n (Durbin 1992).

Definisi 10 (Permutasi Genap dan Permutasi Ganjil)

Permutasi adalah genap jika dapat dituliskan sebagai hasil sejumlah genap transposisi dan permutasi adalah ganjil jika dapat dituliskan sebagai hasil sejumlah ganjil transposisi (Durbin 1992).

Teorema 1

Setiap permutasi dapat ditulis sebagai produk dari transposisi (Grove 1983).

Teorema 2

Jika $\sigma \in S_n$, kemudian σ dapat ditulis sebagai hasil sebanyak genap transposisi jika dan hanya jika σ tidak bisa ditulis sebagai hasil sebanyak ganjil transposisi (Grove 1983).

PEMODELAN 8-PUZZLE

Keberadaan Solusi Suatu Susunan Awal

Berikut akan dibahas beberapa teorema berdasarkan sumber-sumber yang didapat untuk membuktikan bahwa terdapat suatu kondisi awal tertentu sehingga solusi tidak dapat dicapai.

1	2	3
4	5	6
7	8	

Gambar 1 Blok-blok pada 8-puzzle

Delapan keping kotak berangka pada Gambar 1 disebut dengan blok dan sembilan kotak berbeda pada papan disebut dengan sel. Untuk suatu alasan, urutan sel-sel dibuat seperti jalur ular seperti Gambar 2.

1	2	3
6	5	4
7	8	9

Gambar 2 Penomoran sel-sel pada 8-puzzle

Suatu sel akan kosong jika ditempati oleh blok kosong. Setiap pergerakan yang sah terdiri dari menukarkan blok kosong dengan blok tetangganya secara

vertikal atau horizontal. *Placement* adalah bijeksi dari himpunan sel ke himpunan blok (termasuk blok kosong).

Perhatikan bahwa jika blok kosong digerakkan sepanjang jalur ular maka blok kosong dapat dipindahkan ke setiap sel tanpa mengubah urutan blok sebelumnya pada jalur ini. Ini berakibat terdapatnya hubungan ekuivalen dari himpunan *placement*, dua *placement* ekuivalen jika satu *placement* dapat dicapai dari yang lain dengan memindahkan blok kosong sepanjang jalur ular.

Setiap kelas ekuivalensi disebut dengan konfigurasi, setiap konfigurasi mengandung 9 *placement*, satu untuk setiap sel yang ditempati blok kosong.

Misal blok i menempati sel j dan blok kosong menempati sel m , jika $m > j$ maka disebut blok i menempati slot j tetapi jika $m < j$ maka disebut blok i menempati slot $(j - 1)$.

Dalam satu konfigurasi, blok-blok *placement* satu dengan blok-blok *placement* lainnya menempati slot yang sama, jadi konfigurasi dapat dinotasikan oleh $[a_1, a_2, \dots, a_8]$, dengan a_i adalah slot yang blok i tempati dalam konfigurasi.

Setiap langkah dari blok kosong menyebabkan permutasi pada slot-slot yang ditempati oleh blok-blok. Misalnya, memindahkan blok kosong dari sel 5 ke sel 8 menyebabkan permutasi $(5,6,7)$ karena blok sebelumnya dalam sel 8 (slot 7) adalah pindah ke sel 5 (yang menjadi slot 5) dan blok dalam sel 6 sampai 7 naik satu slot. Konfigurasi $[a_1, a_2, \dots, a_8]$ mengalami permutasi σ ditransformasikan ke dalam konfigurasi $[a_1, a_2, \dots, a_8] \sigma = [a_1\sigma, a_2\sigma, \dots, a_8\sigma]$. Karena permutasi bertindak di sebelah kanan maka perkaliannya adalah dari kiri ke kanan.

Misal $\sigma_{i,j}$ menotasikan permutasi yang dihasilkan oleh pergerakan blok kosong dari sel i ke sel j . Kemudian jelas bahwa $\sigma_{i,i+1}$ adalah identitas dan $\sigma_{j,i} = \sigma_{i,j}^{-1}$. Ini menghasilkan 4 permutasi untuk dikerjakan yang terdapat pada Gambar 3. Kuncinya adalah blok kosong dapat digerakkan sepanjang jalur ular ke setiap sel tanpa mengubah konfigurasi. Oleh karena itu, 4 permutasi yang tercantum dalam Gambar 5 dan inversnya dapat diterapkan dalam urutan apapun, sehingga masalah berkurang untuk mengidentifikasi subgrup dari S_8 (grup simetri pada 8 slot) yang dihasilkan oleh permutasi-permutasi ini. Akan dibuktikan bahwa permutasi-permutasi ini menghasilkan A_8 (semua permutasi genap).

$$\begin{aligned} \sigma_{1,6} &= (1, 2, 3, 4, 5) \\ \sigma_{2,5} &= (2, 3, 4) \\ \sigma_{4,9} &= (4, 5, 6, 7, 8) \\ \sigma_{5,8} &= (5, 6, 7) \\ \sigma_{n,n+1} &= \text{id}, n = 1, 2, 3, \dots, 8 \\ \sigma_{i,j} &= \sigma_{j,i}^{-1} \text{ untuk semua } i > j \end{aligned}$$

Gambar 3 Ringkasan dari semua kemungkinan permutasi dari slot dicapai dengan memindahkan blok kosong. Memindahkan blok kosong dari sel i ke sel j menyebabkan permutasi $\sigma_{i,j}$

Lema 1. Untuk $n \geq 3$ 3-cycle menghasilkan A_n . (Archer 1999)

Lema 2. Untuk $n \geq 3$ 3-cycle berurutan $\{(1, 2, 3), (2, 3, 4), \dots, (n-2, n-1, n)\}$ menghasilkan A_n . (Archer 1999)

Teorema 3. Semua cycle pada Gambar 3 menghasilkan A_8 .

Pembuktian ini didasarkan pada teorema yang dibuktikan Archer sebelumnya untuk 15-puzzle.

Bukti: Karena semua cycle adalah ganjil, semua cycle merupakan permutasi genap dan menghasilkan subgroup dari A_8 . Perhatikan bahwa untuk permutasi σ terdapat $\sigma^{-1}(a_1, \dots, a_k)\sigma = (a_1\sigma, \dots, a_k\sigma)$. Kemudian,

$$(1,2,3,4,5)^{-n}(2,3,4)(1,2,3,4,5)^n \text{ menghasilkan } (1,2,3), (2,3,4), (3,4,5);$$

$$(4,5,6,7,8)^{-n}(5,6,7)(4,5,6,7,8)^n \text{ menghasilkan } (4,5,6), (5,6,7), (6,7,8)$$

saat n diasumsikan bernilai -1, 0, 1. Ini merupakan semua 3-cycle berurutan di S_8 , berdasarkan Lema 2 ini menghasilkan A_8 (semua permutasi genap 8-puzzle). ■

Dari Teorema 2 pada bab Landasan Teori telah diketahui bahwa permutasi genap tidak dapat dituliskan dengan sebanyak ganjil transposisi juga sebaliknya, permutasi ganjil tidak dapat dituliskan dengan sebanyak genap transposisi. Karena telah diketahui pada 8-puzzle pergerakan blok kosong menghasilkan permutasi genap maka sebuah permutasi ganjil tidak dapat dibentuk. Ini menghasilkan suatu kondisi bahwa ada susunan awal yang tidak dapat ditemukan solusinya yaitu jika susunan awal tersebut merupakan permutasi ganjil dari konfigurasi susunan solusi pada 8-puzzle.

Dengan demikian, diberikan suatu dua placement Pl_1 dan Pl_2 yang masing-masing memiliki konfigurasi Cf_1 dan Cf_2 , Pl_2 dapat dicapai dari Pl_1 jika dan hanya jika Cf_2 merupakan permutasi genap dari Cf_1 . Perlu diketahui bahwa satu pertukaran slot pada konfigurasi suatu susunan mengakibatkan satu pertukaran blok pada placement susunan tersebut. Oleh karena itu, untuk Pl_1 dan Pl_2 yang mempunyai blok kosong di sel yang sama, Pl_2 dapat dicapai dari Pl_1 jika dan hanya jika Pl_2 merupakan permutasi genap dari 8 blok bernomor di Pl_1 . Misalkan n adalah banyaknya langkah kotak kosong di Pl_1 diukur dari kotak kosong di Pl_2 dan karena setiap langkah kotak kosong menyebabkan transposisi dua blok, maka untuk n ganjil (atau genap), Pl_2 dapat dicapai dari Pl_1 jika dan hanya jika Pl_2 merupakan permutasi ganjil (atau genap) dari 9 blok di Pl_1 . Hal ini dapat dijadikan landasan untuk menentukan suatu susunan awal dapat mencapai solusi atau tidak.

Jika posisi blok kosong pada suatu susunan awal dibandingkan dengan posisi blok kosong susunan solusi (terletak di pojok kanan bawah) maka akan ada 2 kemungkinan, yaitu posisi blok kosong pada susunan awal sama dengan posisi blok kosong pada susunan solusi atau posisinya tidak sama dengan posisi blok kosong pada posisi solusi.

Untuk posisi blok kosong yang terletak sama dengan posisi susunan solusi, misalkan

1	2	3
4	5	6
7	8	

Pl_1 (Susunan Solusi)

1	2	3
4	8	5
7	6	

Pl_2 (Susunan Awal)

Gambar 4 Susunan awal 8-puzzle dengan posisi blok kosong sama dengan posisi blok kosong susunan solusi

$$Pl_1 = 1, 2, 3, 6, 5, 4, 7, 8;$$

$$Pl_2 = 1, 2, 3, 5, 8, 4, 7, 6.$$

Cycle dibentuk dari perubahan angka yang terjadi, dapat dilihat bahwa 6 berubah menjadi 5, 5 menjadi 8, 8 menjadi 6, lalu dapat ditulis (6,5,8). Karena $(6,5,8) = (6,5)(6,8)$, terbentuk dari 2 (genap) transposisi maka susunan tersebut merupakan permutasi genap dari susunan solusi dan dapat diselesaikan (diubah menjadi susunan yang seharusnya dari 8-puzzle). Jika *cycle* yang dibentuk dari perubahan angka pada *placement* terdapat lebih dari satu *cycle* maka tinggal ditambahkan saja banyaknya transposisinya dan dilihat ganjil atau genapnya.

Untuk posisi blok kosong yang terletak sejauh n blok dari posisi blok kosong susunan solusi, misalkan

1	2	3
4	5	6
7	8	

Pl_1 (Susunan Solusi)

1	2	3
	8	5
4	7	6

Pl_2 (Susunan Awal)

Gambar 5 Susunan awal 8-puzzle dengan posisi blok kosong berjarak n blok dari posisi blok kosong susunan solusi

$$Pl_1 = 1, 2, 3, 6, 5, 4, 7, 8, 0;$$

$$Pl_2 = 1, 2, 3, 5, 8, 0, 4, 7, 6.$$

Karena $n = 3$ (ganjil) maka Pl_2 harus merupakan permutasi ganjil dari Pl_1 . *Cycle* yang terbentuk adalah $(6,5,8,7,4,0) = (6,5)(6,8)(6,7)(6,4)(6,0)$. Dapat dilihat bahwa terjadi 5 (ganjil) transposisi sehingga Pl_2 dapat mencapai solusi Pl_1 .

Selain cara tersebut ada cara lain untuk menentukan bisa tidaknya suatu susunan awal memiliki solusi atau tidak yaitu dengan menghitung banyaknya inversi yang terjadi karena genap ganjilnya suatu permutasi dapat ditentukan dari banyaknya inversi yang terjadi. Suatu permutasi σ dalam S_n , misalnya $\sigma = j_1, j_2, \dots, j_n$. σ disebut sebagai permutasi genap jika banyaknya inversi dalam σ adalah genap dan σ disebut permutasi ganjil jika banyaknya inversi dalam σ adalah ganjil. Inversi dalam σ adalah sepasang bilangan bulat (i, k) sedemikian rupa sehingga $i > k$, tetapi i mendahului nilai k dalam σ (Lipschutz 1991). Dapat

dibuat suatu formulasi untuk menentukan suatu susunan awal dapat mencapai susunan solusi atau tidak berdasarkan inversi yang terjadi pada suatu susunan awal. Pertama definisikan $x(i)$ adalah nilai (angka) yang tertera pada blok ke- i , $i = 1, 2, 3, \dots, 9$ seperti pada Gambar 6 berikut:

$x(1)$	$x(2)$	$x(3)$
$x(4)$	$x(5)$	$x(6)$
$x(7)$	$x(8)$	$x(9)$

Gambar 6 Susunan awal 8-puzzle

Pada Gambar 6 $x(i)$ bernilai 1 sampai 8 dan untuk blok kosong bernilai 15. Lalu untuk $j = 1, 2, 3, \dots, 9$ dan $k = 1, 2, 3, \dots, 9$, didefinisikan

$$w(j, k) = \begin{cases} 1, & \text{jika } x(k) < x(j) \\ 0, & \text{selainnya} \end{cases}$$

maka total inversi yang terjadi dapat ditulis

$$t = \sum_{j=1}^9 \sum_{k=1}^9 w(j, k) \quad , \forall j < k$$

lalu genap ganjilnya total inversi di atas dapat ditulis

$$t2 = t \bmod 2.$$

Jika $t2 = 1$ maka artinya banyaknya inversi yang terjadi adalah sebanyak ganjil inversi dan jika $t2 = 0$ artinya terjadi sebanyak genap inversi.

Pada cara sebelumnya telah diketahui bahwa ada dua kondisi posisi blok kosong yaitu sama dengan posisi blok kosong susunan solusi dan yang tidak sama dengan susunan solusi, untuk $i = 1, 2, 3, \dots, 9$ didefinisikan

$$v(i) = \begin{cases} (i + 1) \bmod 2, & \text{jika } x(i) = 15 \\ 0, & \text{selainnya.} \end{cases}$$

Jika blok kosong (blok bernilai 15) berada pada blok ke- i dan i ganjil maka $v(i) = 0$, sedangkan jika i genap maka $v(i) = 1$. Untuk mengetahui genap ganjilnya jarak posisi blok kosong pada suatu susunan dari posisi blok kosong susunan solusi didefinisikan

$$v2 = \sum_{i=1}^9 v(i).$$

Jika $v2 = 1$ maka artinya adalah jarak blok kosong pada susunan awal dengan blok kosong pada susunan solusi yaitu sebanyak ganjil langkah, sedangkan jika $v2 = 0$ jaraknya sebanyak genap langkah (dalam hal ini jika posisi blok kosong sama dengan blok kosong susunan solusi dianggap berjarak genap).

Langkah terakhir adalah membandingkan nilai t_2 dengan v_2 , didefinisikan

$$d = \begin{cases} 1, & \text{jika } t_2 = v_2 \\ 0, & \text{selainnya.} \end{cases}$$

Jika $d = 1$ maka susunan awal dapat mencapai susunan solusi dan jika $d = 0$ susunan awal tersebut tidak dapat mencapai susunan solusi atau tidak dapat diselesaikan.

Model Penyelesaian Solusi

Berdasarkan analisis yang telah dilakukan, maka dapat dibuat formulasi masalah tersebut ke dalam bentuk *integer linear programming*. Bentuk formulasi permasalahan tersebut yaitu:

Indeks

h = indeks yang menyatakan susunan.

m = indeks yang menyatakan banyaknya langkah terpendek dari susunan tersulit (susunan yang memerlukan jumlah maksimum bergerak dalam minimum urutan solusi [Brungger *et al.*1999]) untuk mencapai susunan solusi.

Himpunan

I = himpunan blok (1, 2, 3, 4, 5, 6, 7, 8, 9).

H = himpunan susunan (1, 2, 3, ..., $m+1$).

Variabel keputusan

$x_{i,h}$ = nilai (angka) yang tertera di blok ke- i pada susunan ke- h .

Untuk $h \geq 2$, didefinisikan

$$y_{1h} = \begin{cases} 1, & \text{jika } x_{1,h-1} \text{ ditukar dengan } x_{2,h-1} \text{ pada susunan ke-}h \\ 0, & \text{selainnya} \end{cases}$$

$$y_{2h} = \begin{cases} 1, & \text{jika } x_{2,h-1} \text{ ditukar dengan } x_{3,h-1} \text{ pada susunan ke-}h \\ 0, & \text{selainnya} \end{cases}$$

$$y_{3h} = \begin{cases} 1, & \text{jika } x_{4,h-1} \text{ ditukar dengan } x_{5,h-1} \text{ pada susunan ke-}h \\ 0, & \text{selainnya} \end{cases}$$

$$y_{4h} = \begin{cases} 1, & \text{jika } x_{5,h-1} \text{ ditukar dengan } x_{6,h-1} \text{ pada susunan ke-}h \\ 0, & \text{selainnya} \end{cases}$$

$$y_{5h} = \begin{cases} 1, & \text{jika } x_{7,h-1} \text{ ditukar dengan } x_{8,h-1} \text{ pada susunan ke-}h \\ 0, & \text{selainnya} \end{cases}$$

$$y_{6h} = \begin{cases} 1, & \text{jika } x_{8,h-1} \text{ ditukar dengan } x_{9,h-1} \text{ pada susunan ke-}h \\ 0, & \text{selainnya} \end{cases}$$

$$y_{7h} = \begin{cases} 1, & \text{jika } x_{1,h-1} \text{ ditukar dengan } x_{4,h-1} \text{ pada susunan ke-}h \\ 0, & \text{selainnya} \end{cases}$$

$$y_{8h} = \begin{cases} 1, & \text{jika } x_{2,h-1} \text{ ditukar dengan } x_{5,h-1} \text{ pada susunan ke-}h \\ 0, & \text{selainnya} \end{cases}$$

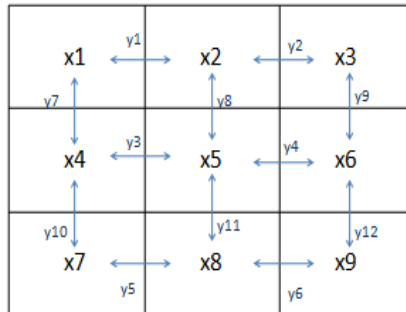
$$y_{9h} = \begin{cases} 1, & \text{jika } x_{3,h-1} \text{ ditukar dengan } x_{6,h-1} \text{ pada susunan ke-}h \\ 0, & \text{selainnya} \end{cases}$$

$$y_{10h} = \begin{cases} 1, & \text{jika } x_{4,h-1} \text{ ditukar dengan } x_{7,h-1} \text{ pada susunan ke-}h \\ 0, & \text{selainnya} \end{cases}$$

$$y_{11h} = \begin{cases} 1, & \text{jika } x_{5,h-1} \text{ ditukar dengan } x_{8,h-1} \text{ pada susunan ke-}h \\ 0, & \text{selainnya} \end{cases}$$

$$y_{12h} = \begin{cases} 1, & \text{jika } x_{6,h-1} \text{ ditukar dengan } x_{9,h-1} \text{ pada susunan ke-}h \\ 0, & \text{selainnya} \end{cases}$$

Variabel-variabel di atas jika dilustrasikan pada papan 8-puzzle adalah seperti berikut:



Gambar 7 Ilustrasi variabel-variabel keputusan pada bidang 8-puzzle

Formulasi Model 1

Fungsi Objektif Model 1

Meminimumkan jumlah perkalian antara nilai blok dengan konstanta yang telah ditentukan untuk baris pertama 8-puzzle pada susunan terakhir (karena $m = 31$ [Gasser 1995] maka susunan terakhir adalah ketika $h = 32$).

$$\min z = 3x_{1,32} + 2x_{2,32} + x_{3,32}$$

Kendala-Kendala Model 1

1. Kendala yang menunjukkan pertukaran blok.

$$x_{1,h} \geq x_{2,h-1} - (1 - y_{1h}) * M$$

$$x_{1,h} \leq x_{2,h-1} + (1 - y_{1h}) * M$$

$$x_{2,h} \geq x_{1,h-1} - (1 - y_{1h}) * M$$

$$x_{2,h} \leq x_{1,h-1} + (1 - y_{1h}) * M$$

$$x_{2,h} \geq x_{3,h-1} - (1 - y_{2h}) * M$$

$$x_{2,h} \leq x_{3,h-1} + (1 - y_{2h}) * M$$

$$x_{3,h} \geq x_{2,h-1} - (1 - y_{2h}) * M$$

$$x_{3,h} \leq x_{2,h-1} + (1 - y_{2h}) * M$$

$$\begin{aligned}
x_{4,h} &\geq x_{5,h-1} - (1 - y_{3h}) * M \\
x_{4,h} &\leq x_{5,h-1} + (1 - y_{3h}) * M \\
x_{5,h} &\geq x_{4,h-1} - (1 - y_{3h}) * M \\
x_{5,h} &\leq x_{4,h-1} + (1 - y_{3h}) * M \\
x_{5,h} &\geq x_{6,h-1} - (1 - y_{4h}) * M \\
x_{5,h} &\leq x_{6,h-1} + (1 - y_{4h}) * M \\
x_{6,h} &\geq x_{5,h-1} - (1 - y_{4h}) * M \\
x_{6,h} &\leq x_{5,h-1} + (1 - y_{4h}) * M \\
x_{7,h} &\geq x_{8,h-1} - (1 - y_{5h}) * M \\
x_{7,h} &\leq x_{8,h-1} + (1 - y_{5h}) * M \\
x_{8,h} &\geq x_{7,h-1} - (1 - y_{5h}) * M \\
x_{8,h} &\leq x_{7,h-1} + (1 - y_{5h}) * M \\
x_{8,h} &\geq x_{9,h-1} - (1 - y_{6h}) * M \\
x_{8,h} &\leq x_{9,h-1} + (1 - y_{6h}) * M \\
x_{9,h} &\geq x_{8,h-1} - (1 - y_{6h}) * M \\
x_{9,h} &\leq x_{8,h-1} + (1 - y_{6h}) * M \\
x_{1,h} &\geq x_{4,h-1} - (1 - y_{7h}) * M \\
x_{1,h} &\leq x_{4,h-1} + (1 - y_{7h}) * M \\
x_{4,h} &\geq x_{1,h-1} - (1 - y_{7h}) * M \\
x_{4,h} &\leq x_{1,h-1} + (1 - y_{7h}) * M \\
x_{2,h} &\geq x_{5,h-1} - (1 - y_{8h}) * M \\
x_{2,h} &\leq x_{5,h-1} + (1 - y_{8h}) * M \\
x_{5,h} &\geq x_{2,h-1} - (1 - y_{8h}) * M \\
x_{5,h} &\leq x_{2,h-1} + (1 - y_{8h}) * M \\
x_{3,h} &\geq x_{6,h-1} - (1 - y_{9h}) * M \\
x_{3,h} &\leq x_{6,h-1} + (1 - y_{9h}) * M \\
x_{6,h} &\geq x_{3,h-1} - (1 - y_{9h}) * M \\
x_{6,h} &\leq x_{3,h-1} + (1 - y_{9h}) * M \\
x_{4,h} &\geq x_{7,h-1} - (1 - y_{10h}) * M \\
x_{4,h} &\leq x_{7,h-1} + (1 - y_{10h}) * M \\
x_{7,h} &\geq x_{4,h-1} - (1 - y_{10h}) * M \\
x_{7,h} &\leq x_{4,h-1} + (1 - y_{10h}) * M \\
x_{5,h} &\geq x_{8,h-1} - (1 - y_{11h}) * M \\
x_{5,h} &\leq x_{8,h-1} + (1 - y_{11h}) * M \\
x_{8,h} &\geq x_{5,h-1} - (1 - y_{11h}) * M \\
x_{8,h} &\leq x_{5,h-1} + (1 - y_{11h}) * M \\
x_{6,h} &\geq x_{9,h-1} - (1 - y_{12h}) * M \\
x_{6,h} &\leq x_{9,h-1} + (1 - y_{12h}) * M \\
x_{9,h} &\geq x_{6,h-1} - (1 - y_{12h}) * M \\
x_{9,h} &\leq x_{6,h-1} + (1 - y_{12h}) * M \\
x_{1,h} &\geq x_{1,h-1} - (y_{1h} + y_{7h}) * M \\
x_{1,h} &\leq x_{1,h-1} + (y_{1h} + y_{7h}) * M \\
x_{2,h} &\geq x_{2,h-1} - (y_{1h} + y_{2h} + y_{8h}) * M \\
x_{2,h} &\leq x_{2,h-1} + (y_{1h} + y_{2h} + y_{8h}) * M
\end{aligned}$$



$$\begin{aligned}
x_{3,h} &\geq x_{3,h-1} - (y_{2h} + y_{9h}) * M \\
x_{3,h} &\leq x_{3,h-1} + (y_{2h} + y_{9h}) * M \\
x_{4,h} &\geq x_{4,h-1} - (y_{3h} + y_{7h} + y_{10h}) * M \\
x_{4,h} &\leq x_{4,h-1} + (y_{3h} + y_{7h} + y_{10h}) * M \\
x_{5,h} &\geq x_{5,h-1} - (y_{3h} + y_{4h} + y_{8h} + y_{11h}) * M \\
x_{5,h} &\leq x_{5,h-1} + (y_{3h} + y_{4h} + y_{8h} + y_{11h}) * M \\
x_{6,h} &\geq x_{6,h-1} - (y_{4h} + y_{9h} + y_{12h}) * M \\
x_{6,h} &\leq x_{6,h-1} + (y_{4h} + y_{9h} + y_{12h}) * M \\
x_{7,h} &\geq x_{7,h-1} - (y_{5h} + y_{10h}) * M \\
x_{7,h} &\leq x_{7,h-1} + (y_{5h} + y_{10h}) * M \\
x_{8,h} &\geq x_{8,h-1} - (y_{5h} + y_{6h} + y_{11h}) * M \\
x_{8,h} &\leq x_{8,h-1} + (y_{5h} + y_{6h} + y_{11h}) * M \\
x_{9,h} &\geq x_{9,h-1} - (y_{6h} + y_{12h}) * M \\
x_{9,h} &\leq x_{9,h-1} + (y_{6h} + y_{12h}) * M
\end{aligned}$$

$\forall h \in H$ dengan $h \geq 2$. M = bilangan yang cukup besar nilainya.

2. Dalam 1 langkah perubahan suatu susunan ke susunan berikutnya hanya ada 1 pertukaran blok.

$$y_{1h} + y_{2h} + y_{3h} + y_{4h} + y_{5h} + y_{6h} + y_{7h} + y_{8h} + y_{9h} + y_{10h} + y_{11h} + y_{12h} \leq 1$$

$\forall h \in H$ dengan $h \geq 2$.

3. Blok yang bisa ditukar hanya blok yang bernilai 15.

$$\begin{aligned}
16 * y_{1h} &\leq x_{1,h-1} + x_{2,h-1} \\
16 * y_{2h} &\leq x_{2,h-1} + x_{3,h-1} \\
16 * y_{3h} &\leq x_{4,h-1} + x_{5,h-1} \\
16 * y_{4h} &\leq x_{5,h-1} + x_{6,h-1} \\
16 * y_{5h} &\leq x_{7,h-1} + x_{8,h-1} \\
16 * y_{6h} &\leq x_{8,h-1} + x_{9,h-1} \\
16 * y_{7h} &\leq x_{1,h-1} + x_{4,h-1} \\
16 * y_{8h} &\leq x_{2,h-1} + x_{5,h-1} \\
16 * y_{9h} &\leq x_{3,h-1} + x_{6,h-1} \\
16 * y_{10h} &\leq x_{4,h-1} + x_{7,h-1} \\
16 * y_{11h} &\leq x_{5,h-1} + x_{8,h-1} \\
16 * y_{12h} &\leq x_{6,h-1} + x_{9,h-1}
\end{aligned}$$

$\forall h \in H$ dengan $h \geq 2$.

4. Tidak ada langkah pertukaran kosong di antara langkah-langkah yang diambil, misal $y_{i,2} = 1$ dan $y_{j,4} = 1$ maka $y_{k,3}$ untuk suatu k tertentu harus bernilai 1 juga.

$$\begin{aligned}
 & y1_h + y2_h + y3_h + y4_h + y5_h + y6_h + y7_h + y8_h + y9_h + y10_h + \\
 & \quad y11_h + y12_h \leq \\
 & y1_{h-1} + y2_{h-1} + y3_{h-1} + y4_{h-1} + y5_{h-1} + y6_{h-1} + y7_{h-1} + \\
 & \quad y8_{h-1} + y9_{h-1} + y10_{h-1} + y11_{h-1} + y12_{h-1}
 \end{aligned}$$

$\forall h \in H$ dengan $h \geq 3$.

5. Tidak ada langkah pertukaran yang sama seperti pertukaran sebelumnya, misal jika $y1 = 1$ untuk suatu h maka $y1 = 0$ untuk $h + 1$.

$$\begin{aligned}
 y1_h &\leq 1 - y1_{h-1} \\
 y2_h &\leq 1 - y2_{h-1} \\
 y3_h &\leq 1 - y3_{h-1} \\
 y4_h &\leq 1 - y4_{h-1} \\
 y5_h &\leq 1 - y5_{h-1} \\
 y6_h &\leq 1 - y6_{h-1} \\
 y7_h &\leq 1 - y7_{h-1} \\
 y8_h &\leq 1 - y8_{h-1} \\
 y9_h &\leq 1 - y9_{h-1} \\
 y10_h &\leq 1 - y10_{h-1} \\
 y11_h &\leq 1 - y11_{h-1} \\
 y12_h &\leq 1 - y12_{h-1}
 \end{aligned}$$

$\forall h \in H$ dengan $h \geq 2$.

6. Rentang nilai z dapat diketahui secara manual, yaitu

$$10 \leq z \leq 68$$

7. Kendala biner

$$yk_h \in \{0,1\}, k = 1, 2, 3, \dots, 12; \forall h \in H \text{ dengan } h \geq 2$$

Formulasi Model 2

Karena dengan formulasi tahap ke-1 telah didapat solusi untuk baris pertama dari *8-puzzle*, sekarang dengan menggunakan model yang sama tetapi dengan mengubah fungsi objektifnya dan memodifikasi kendala-kendalanya sehingga sesuai untuk *puzzle* 2×3 sisanya (blok 4 sampai 9).

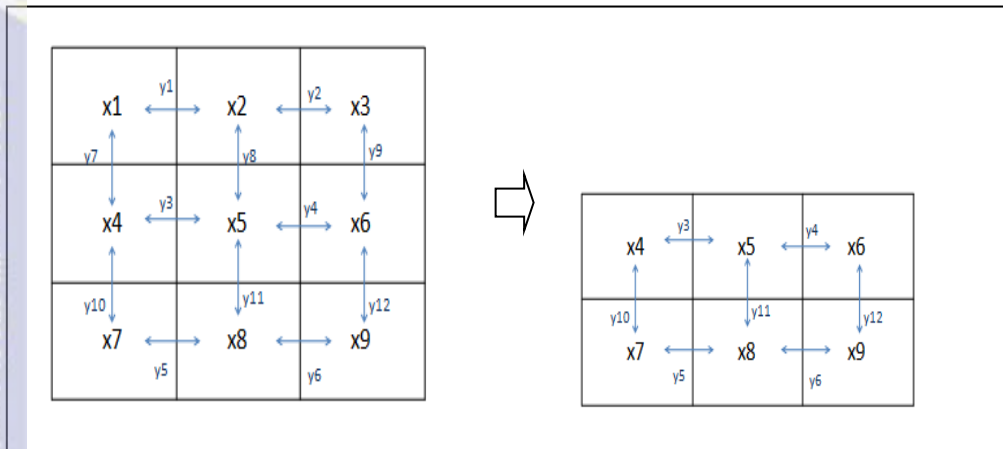
Fungsi Objektif Model 2

Memaksimumkan jumlah perkalian antara nilai kotak dengan konstanta yang telah ditentukan untuk setiap kotak pada *puzzle* 2×3 pada susunan terakhir (karena $m = 21$ [Lampiran 6]) maka susunan terakhir adalah ketika $h = 22$.

$$\max z = x_{4,22} + 2x_{5,22} + 3x_{6,22} + 4x_{7,22} + 5x_{8,22} + 6x_{9,22}$$

Karena baris pertama *8-puzzle* telah berada pada posisi solusi, kendala yang dipakai adalah sama seperti formulasi Model 1 tetapi dengan menghilangkan

beberapa variabel yang berkaitan dengan baris pertama *8-puzzle* seperti yang terlihat pada Gambar 8 berikut:



Gambar 8 Penghilangan beberapa variabel pada Model 1

Kendala-Kendala Model 2

1. Kendala yang menunjukkan pertukaran kotak.

$$\begin{aligned}
 x_{4,h} &\geq x_{5,h-1} - (1 - y_{3h}) * M \\
 x_{4,h} &\leq x_{5,h-1} + (1 - y_{3h}) * M \\
 x_{5,h} &\geq x_{4,h-1} - (1 - y_{3h}) * M \\
 x_{5,h} &\leq x_{4,h-1} + (1 - y_{3h}) * M \\
 x_{5,h} &\geq x_{6,h-1} - (1 - y_{4h}) * M \\
 x_{5,h} &\leq x_{6,h-1} + (1 - y_{4h}) * M \\
 x_{6,h} &\geq x_{5,h-1} - (1 - y_{4h}) * M \\
 x_{6,h} &\leq x_{5,h-1} + (1 - y_{4h}) * M \\
 x_{7,h} &\geq x_{8,h-1} - (1 - y_{5h}) * M \\
 x_{7,h} &\leq x_{8,h-1} + (1 - y_{5h}) * M \\
 x_{8,h} &\geq x_{7,h-1} - (1 - y_{5h}) * M \\
 x_{8,h} &\leq x_{7,h-1} + (1 - y_{5h}) * M \\
 x_{8,h} &\geq x_{9,h-1} - (1 - y_{6h}) * M \\
 x_{8,h} &\leq x_{9,h-1} + (1 - y_{6h}) * M \\
 x_{9,h} &\geq x_{8,h-1} - (1 - y_{6h}) * M \\
 x_{9,h} &\leq x_{8,h-1} + (1 - y_{6h}) * M \\
 x_{4,h} &\geq x_{7,h-1} - (1 - y_{10h}) * M \\
 x_{4,h} &\leq x_{7,h-1} + (1 - y_{10h}) * M \\
 x_{7,h} &\geq x_{4,h-1} - (1 - y_{10h}) * M \\
 x_{7,h} &\leq x_{4,h-1} + (1 - y_{10h}) * M \\
 x_{5,h} &\geq x_{8,h-1} - (1 - y_{11h}) * M \\
 x_{5,h} &\leq x_{8,h-1} + (1 - y_{11h}) * M \\
 x_{8,h} &\geq x_{5,h-1} - (1 - y_{11h}) * M \\
 x_{8,h} &\leq x_{5,h-1} + (1 - y_{11h}) * M \\
 x_{6,h} &\geq x_{9,h-1} - (1 - y_{12h}) * M
 \end{aligned}$$

$$\begin{aligned}
x_{6,h} &\leq x_{9,h-1} + (1 - y_{12_h}) * M \\
x_{9,h} &\geq x_{6,h-1} - (1 - y_{12_h}) * M \\
x_{9,h} &\leq x_{6,h-1} + (1 - y_{12_h}) * M \\
x_{4,h} &\geq x_{4,h-1} - (y_{3_h} + y_{10_h}) * M \\
x_{4,h} &\leq x_{4,h-1} + (y_{3_h} + y_{10_h}) * M \\
x_{5,h} &\geq x_{5,h-1} - (y_{3_h} + y_{4_h} + y_{11_h}) * M \\
x_{5,h} &\leq x_{5,h-1} + (y_{3_h} + y_{4_h} + y_{11_h}) * M \\
x_{6,h} &\geq x_{6,h-1} - (y_{4_h} + y_{12_h}) * M \\
x_{6,h} &\leq x_{6,h-1} + (y_{4_h} + y_{12_h}) * M \\
x_{7,h} &\geq x_{7,h-1} - (y_{5_h} + y_{10_h}) * M \\
x_{7,h} &\leq x_{7,h-1} + (y_{5_h} + y_{10_h}) * M \\
x_{8,h} &\geq x_{8,h-1} - (y_{5_h} + y_{6_h} + y_{11_h}) * M \\
x_{8,h} &\leq x_{8,h-1} + (y_{5_h} + y_{6_h} + y_{11_h}) * M \\
x_{9,h} &\geq x_{9,h-1} - (y_{6_h} + y_{12_h}) * M \\
x_{9,h} &\leq x_{9,h-1} + (y_{6_h} + y_{12_h}) * M
\end{aligned}$$

$\forall h \in H$ dengan $h \geq 2$. $M =$ bilangan yang cukup besar nilainya.

2. Dalam 1 langkah perubahan suatu susunan ke susunan berikutnya hanya ada 1 pertukaran blok .

$$y_{3_h} + y_{4_h} + y_{5_h} + y_{6_h} + y_{10_h} + y_{11_h} + y_{12_h} \leq 1$$

$\forall h \in H$ dengan $h \geq 2$.

3. Blok yang bisa ditukar hanya blok yang bernilai 15.

$$\begin{aligned}
16 * y_{3_h} &\leq x_{4,h-1} + x_{5,h-1} \\
16 * y_{4_h} &\leq x_{5,h-1} + x_{6,h-1} \\
16 * y_{5_h} &\leq x_{7,h-1} + x_{8,h-1} \\
16 * y_{6_h} &\leq x_{8,h-1} + x_{9,h-1} \\
16 * y_{10_h} &\leq x_{4,h-1} + x_{7,h-1} \\
16 * y_{11_h} &\leq x_{5,h-1} + x_{8,h-1} \\
16 * y_{12_h} &\leq x_{6,h-1} + x_{9,h-1}
\end{aligned}$$

$\forall h \in H$ dengan $h \geq 2$.

4. Tidak ada langkah pertukaran kosong di antara langkah-langkah yang terjadi pertukaran, misal $y_{i,2} = 1$ dan $y_{j,4} = 1$ maka $y_{k,3}$ harus bernilai 1 juga.

$$\begin{aligned}
&y_{3_h} + y_{4_h} + y_{5_h} + y_{6_h} + y_{10_h} + y_{11_h} + y_{12_h} \leq \\
&y_{3_{h-1}} + y_{4_{h-1}} + y_{5_{h-1}} + y_{6_{h-1}} + y_{10_{h-1}} + y_{11_{h-1}} + y_{12_{h-1}}
\end{aligned}$$

$\forall h \in H$ dengan $h \geq 3$.

5. Tidak ada langkah pertukaran yang sama seperti pertukaran sebelumnya, misal jika $y_1 = 1$ untuk suatu h maka $y_1 = 0$ untuk $h + 1$.

$$\begin{aligned}y_{3h} &\leq 1 - y_{3h-1} \\y_{4h} &\leq 1 - y_{4h-1} \\y_{5h} &\leq 1 - y_{5h-1} \\y_{6h} &\leq 1 - y_{6h-1} \\y_{10h} &\leq 1 - y_{10h-1} \\y_{11h} &\leq 1 - y_{11h-1} \\y_{12h} &\leq 1 - y_{12h-1}\end{aligned}$$

$\forall h \in H$ dengan $h \geq 2$.

6. Rentang z dapat diketahui secara manual.

$$125 \leq z \leq 190.$$

7. Kendala biner

$$y_{k_h} \in \{0,1\}, k = 3, 4, 5, 6, 10, 11, 12; \forall h \in H \text{ dengan } h \geq 2.$$

APLIKASI MODEL

Untuk aplikasi model yang telah diformulasikan ini akan diterapkan langsung ke salah satu susunan awal yang merupakan susunan terjauh (posisi tersulit) dari 8 -puzzle yaitu:

8	7	6
	4	1
2	5	3

Gambar 9 Salah satu susunan tersulit 8 -puzzle

Gambar 9 di atas merupakan salah satu susunan tersulit (terjauh) yang tentunya dapat mencapai susunan solusi, maka sebenarnya tidak perlu lagi diperiksa susunan ini bisa mencapai susunan solusi atau tidak, tetapi untuk meyakinkan dan memperjelas bahwa susunan ini memiliki solusi maka akan dilakukan pemeriksaan pada susunan ini.

Seperti cara yang sudah dijelaskan dengan sel-sel diurutkan seperti jalur ular maka *placement* solusi adalah 1,2,3,6,5,4,7,8,0 dan susunan tersebut memiliki *placement* 8,7,6,1,4,0,2,5,3, maka *cycle* yang dihasilkan adalah (1,8,5,4,0,3,6)(2,7)

dan karena $(1,8,5,4,0,3,6)(2,7) = (1,8)(1,5)(1,4)(1,0)(1,3)(1,6)(2,7)$ merupakan hasil transposisi sebanyak 7 (ganjil) maka susunan tersebut adalah permutasi ganjil. Susunan tersebut memiliki $n = 3$ (ganjil) dan karena permutasi ganjil maka susunan solusi dapat dicapai oleh susunan tersebut.

Cara lain menentukan bisa tidaknya susunan solusi dicapai adalah dengan menggunakan formulasi model yang telah dibuat (yang berhubungan dengan banyaknya inversi yang terjadi) dan dapat diselesaikan dengan bantuan *software* LINGO 11.0 dengan mengisi $x(1) = 8, x(2) = 7, x(3) = 6, x(4) = 15, x(5) = 4, x(6) = 1, x(7) = 2, x(8) = 5, x(9) = 3$, maka didapatkan hasil $d = 1$ yang artinya susunan awal dapat mencapai susunan solusi.

Langkah selanjutnya adalah memasukkan nilai-nilai tersebut ke formulasi model 1 yaitu $x_{1,1} = 8, x_{2,1} = 7, x_{3,1} = 6, x_{4,1} = 15, x_{5,1} = 4, x_{6,1} = 1, x_{7,1} = 2, x_{8,1} = 5, x_{9,1} = 3$ dengan $M = 100$. Setelah menggunakan bantuan *software* LINGO 11.0 maka telah didapat solusinya yaitu berupa pemilihan pertukaran blok kosong ($y_{i,h}$ yang bernilai 1) di setiap langkah h yang dapat dilihat pada Tabel 1.

Tabel 1 Solusi Model 1 untuk susunan awal *8-puzzle* pada Gambar 9

H	y yang bernilai 1	Pertukaran blok yang terjadi
1	-	-
2	y3	$x_{4,1}$ dengan $x_{5,1}$
3	y4	$x_{5,2}$ dengan $x_{6,2}$
4	y12	$x_{6,3}$ dengan $x_{9,3}$
5	y6	$x_{8,4}$ dengan $x_{9,4}$
6	y5	$x_{7,5}$ dengan $x_{8,5}$
7	y10	$x_{4,6}$ dengan $x_{7,6}$
8	y3	$x_{4,7}$ dengan $x_{5,7}$
9	y4	$x_{5,8}$ dengan $x_{6,8}$
10	y12	$x_{6,9}$ dengan $x_{9,9}$
11	y6	$x_{8,10}$ dengan $x_{9,10}$
12	y11	$x_{5,11}$ dengan $x_{8,11}$
13	y8	$x_{2,12}$ dengan $x_{5,12}$
14	y1	$x_{1,13}$ dengan $x_{2,13}$
15	y7	$x_{1,14}$ dengan $x_{4,14}$
16	y3	$x_{4,15}$ dengan $x_{5,15}$
17	y11	$x_{5,16}$ dengan $x_{8,16}$
18	y6	$x_{8,17}$ dengan $x_{9,17}$
19	y12	$x_{6,18}$ dengan $x_{9,18}$
20	y9	$x_{3,19}$ dengan $x_{6,19}$
21	y2	$x_{2,20}$ dengan $x_{3,20}$
22	y8	$x_{2,21}$ dengan $x_{5,21}$
23	y11	$x_{5,22}$ dengan $x_{8,22}$
24	y6	$x_{8,23}$ dengan $x_{9,23}$
25	y12	$x_{6,23}$ dengan $x_{9,23}$
26	y9	$x_{3,24}$ dengan $x_{6,24}$
27	y2	$x_{2,25}$ dengan $x_{3,25}$
28	y8	$x_{2,26}$ dengan $x_{5,26}$
29	y4	$x_{5,27}$ dengan $x_{6,27}$
30	-	-
31	-	-
32	-	-

Dengan demikian diperoleh susunan baru dengan baris pertama *8-puzzle* seperti pada Gambar 10.

1	2	3
7	8	
4	5	6

Gambar 10 Susunan *8-puzzle* hasil solusi Model 1

Langkah selanjutnya adalah memasukkan nilai-nilai tersebut ke formulasi Model 2 yaitu $x_{4,1} = 7$, $x_{5,1} = 8$, $x_{6,1} = 15$, $x_{7,1} = 4$, $x_{8,1} = 5$, $x_{9,1} = 6$ dengan

$M = 100$. Setelah menggunakan bantuan *software* LINGO 11.0 maka telah didapat solusinya yang dapat dilihat pada Tabel 2.

Tabel 2 Solusi Model 2 untuk susunan *8-puzzle* pada Gambar 10

H	y yang bernilai 1	Pertukaran blok yang terjadi
1	-	-
2	y_4	$x_{5,1}$ dengan $x_{6,1}$
3	y_3	$x_{4,2}$ dengan $x_{5,2}$
4	y_{10}	$x_{4,3}$ dengan $x_{7,3}$
5	y_5	$x_{7,4}$ dengan $x_{8,4}$
6	y_{11}	$x_{5,5}$ dengan $x_{8,5}$
7	y_4	$x_{5,6}$ dengan $x_{6,6}$
8	y_{12}	$x_{6,7}$ dengan $x_{9,7}$
9	y_6	$x_{8,8}$ dengan $x_{9,8}$
10	y_5	$x_{7,9}$ dengan $x_{8,9}$
11	y_{10}	$x_{4,10}$ dengan $x_{7,10}$
12	y_3	$x_{4,11}$ dengan $x_{5,11}$
13	y_{11}	$x_{5,12}$ dengan $x_{8,12}$
14	y_6	$x_{8,13}$ dengan $x_{9,13}$
15	y_{12}	$x_{6,14}$ dengan $x_{9,14}$
16	y_4	$x_{5,15}$ dengan $x_{6,15}$
17	y_3	$x_{4,16}$ dengan $x_{5,16}$
18	y_{10}	$x_{4,17}$ dengan $x_{7,17}$
19	y_5	$x_{7,18}$ dengan $x_{8,18}$
20	y_{11}	$x_{5,19}$ dengan $x_{8,19}$
21	y_4	$x_{5,20}$ dengan $x_{6,20}$
22	y_{12}	$x_{6,21}$ dengan $x_{9,21}$

Susunan solusinya telah dapat dicapai seperti pada Gambar 11.

1	2	3
4	5	6
7	8	

Gambar 11 Susunan solusi yang didapat dari susunan *8-puzzle* pada Gambar 9 melalui Model 1 dan Model 2

SIMPULAN DAN SARAN

Simpulan

Permainan *8-puzzle* dapat diselesaikan dengan formulasi *integer linear programming* (ILP) yang bertujuan untuk mencapai susunan solusi. Telah dapat ditentukan suatu susunan awal dapat mencapai susunan solusi atau tidak dengan melihat ganjil genapnya permutasi suatu susunan awal. Jika susunan awal memiliki posisi blok kosong di pojok kanan bawah (sama dengan posisi blok kosong solusi) dan itu merupakan permutasi genap dari 8 blok (1 sampai 8) di susunan solusi maka susunan awal tersebut memiliki solusi. Sedangkan agar susunan awal memiliki solusi untuk kondisi posisi blok kosong berjarak n langkah dari pojok kanan bawah mensyaratkan n genap dan susunan tersebut merupakan permutasi genap dari 9 blok (1 sampai 8 dan blok kosong) susunan solusi atau n ganjil dan susunan tersebut merupakan permutasi ganjil dari 9 blok dari susunan solusi. Cara lain untuk memeriksa ada tidaknya solusi suatu susunan awal adalah dengan melihat inversi dari 9 blok yang terjadi pada susunan awal, dengan menghitung $t2$ (genap ganjilnya total inversi yang terjadi pada suatu susunan awal) dan membandingkannya dengan $v2$ (genap ganjilnya n) pada model sederhana yang telah dibuat. Susunan awal dapat mencapai susunan solusi jika dan hanya jika $t2$ dan $v2$ bernilai sama. Model tersebut selanjutnya diterapkan dalam *software* LINGO 11.0 untuk mendapatkan solusinya.

Pada langkah berikutnya, diformulasikan Model 1 dan Model 2 yang digunakan untuk menentukan langkah-langkah pertukaran blok kosong dengan salah satu blok terdekatnya yang dapat mengubah suatu susunan awal sehingga pada akhirnya dapat mencapai susunan solusi dari *8-puzzle*. Kedua model ini selanjutnya diterapkan pada *software* LINGO 11.0 yang kemudian didapat solusi optimalnya.

Saran

Karya ilmiah ini telah membahas mengenai cara memodelkan *8-puzzle* dengan *integer linear programming* untuk mencari solusi dari suatu susunan awal dengan terlebih dahulu memeriksa susunan awal tersebut memiliki solusi atau tidak sehingga ruang lingkupnya masih sangat umum, sehingga masih banyak hal lain yang bisa diperdalam lagi seperti salah satunya adalah mencari solusi langkah terpendek sehingga suatu susunan awal dapat mencapai solusi.

DAFTAR PUSTAKA

- Archer AF. 1999. A modern treatment of the 15 puzzle. *American Math Monthly*. 106(9):793-799.doi:10.2307/2589612.
- Brungger A, Marzetta A, Fukuda K, Nievergelt J. 1999. The parallel search bench ZRAM and its applications. *Annals of Operation Research*. 90:45-63.
- Durbin J R.1992. *Modern Algebra: An Introduction*. Ed ke-3. New York (US): John Wiley & Sons.
- Gasser R. 1995. Harnessing computational resources for efficient exhaustive search [disertasi]. Zurich (CH): Swiss Federal Institute of Technology Zurich.
- Grove L. 1983. *Algebra*. New York (US): Dover Pubs.
- Johnson WW, Story WE. 1879. Notes on the "15" puzzle. *American Journal of Mathematics*. 2(4):397–404.doi:10.2307/2369492.
- Lipschutz S. 1991. *Schaum's Outline of Theory and Problems of Linear Algebra*. Ed ke-2. New York (US): McGraw-Hill.
- Winston WL. 2004. *Operations Research: Applications and Algorithms*. Ed ke-4. New York (US): Duxbury.

Lampiran 1 Formulasi dan solusi model untuk memeriksa bisa tidaknya suatu susunan awal (pada Gambar 9) mencapai susunan solusi *8-puzzle* pada LINGO 11.0

model:

sets:

```
j0/1..9/:x,v;
link(j0,j0):w;
```

endsets

!susunan awal 8-puzzle;

```
x(1)=8;x(2)=7;x(3)=6;
```

```
x(4)=15;x(5)=4;x(6)=1;
```

```
x(7)=2;x(8)=5;x(9)=3;
```

!untuk setiap j,k w bernilai 1 jika x(k)<x(j);

```
@for(link(j,k):w(j,k)=@if(x(k)#lt#x(j),1,0));
```

!menghitung total inversi;

```
t=@sum(link(j,k)|j#lt#k:w(j,k));
```

!menentukan posisi blok kosong;

```
@for(j0(i):v(i)=@if(x(i)#eq#15,@mod(i+1,2),0));
```

!menentukan ganjil genapnya jarak blok kosong;

```
v2=@sum(j0(i):v(i));
```

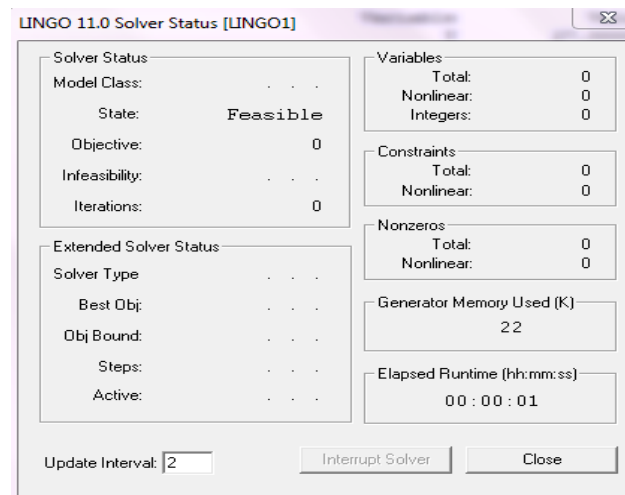
!menentukan ganjil genapnya total inversi;

```
t2=@mod(t,2);
```

!menentukan ada tidaknya solusi pada susunan awal;

```
d=@if(t2#eq#v2,1,0);
```

end



Feasible solution found.
Total solver iterations: 0

Variable	Value
T	27.000000
V2	1.000000
T2	1.000000
D	1.000000
X(1)	8.000000
X(2)	7.000000
X(3)	6.000000
X(4)	15.000000
X(5)	4.000000
X(6)	1.000000
X(7)	2.000000
X(8)	5.000000
X(9)	3.000000
V(1)	0.000000
V(2)	0.000000
V(3)	0.000000
V(4)	1.000000
V(5)	0.000000
V(6)	0.000000
V(7)	0.000000
V(8)	0.000000
V(9)	0.000000
W(1, 1)	0.000000
W(1, 2)	1.000000
W(1, 3)	1.000000
W(1, 4)	0.000000
W(1, 5)	1.000000
W(1, 6)	1.000000
W(1, 7)	1.000000
W(1, 8)	1.000000
W(1, 9)	1.000000
W(2, 1)	0.000000
W(2, 2)	0.000000
W(2, 3)	1.000000
W(2, 4)	0.000000
W(2, 5)	1.000000
W(2, 6)	1.000000
W(2, 7)	1.000000
W(2, 8)	1.000000
W(2, 9)	1.000000
W(3, 1)	0.000000
W(3, 2)	0.000000
W(3, 3)	0.000000
W(3, 4)	0.000000

$W(8, 9)$	1.000000
$W(9, 1)$	0.000000
$W(9, 2)$	0.000000
$W(9, 3)$	0.000000
$W(9, 4)$	0.000000
$W(9, 5)$	0.000000
$W(9, 6)$	1.000000
$W(9, 7)$	1.000000
$W(9, 8)$	0.000000
$W(9, 9)$	0.000000



Lampiran 2 Formulasi dan solusi Model 1 untuk suatu susunan awal (pada Gambar 9) menggunakan LINGO 11.0

model:

```

sets:
posisi/1..32/:x1,x2,x3,x4,x5,x6,x7,x8,x9,
y1,y2,y3,y4,y5,y6,y7,y8,
y9,y10,y11,y12;
endsets
!susunan awal;
x1(1)=8;x2(1)=7;x3(1)=6;
x4(1)=15;x5(1)=4;x6(1)=1;
x7(1)=2;x8(1)=5;x9(1)=3;
!fungsi objektif;
min=z;
z=@sum(posisi(h)|h#eq#@size(hari):3*x1(h)+2*x2(h)+x3(h)
;
@bnd(10,z,68); !rentang nilai z;
!kendala yang menggambarkan pertukaran blok;
@for(posisi(h)|h#ge#2:x1(h)>=x2(h-1)-(1-y1(h))*100);
@for(posisi(h)|h#ge#2:x1(h)<=x2(h-1)+(1-y1(h))*100);
@for(posisi(h)|h#ge#2:x2(h)>=x1(h-1)-(1-y1(h))*100);
@for(posisi(h)|h#ge#2:x2(h)<=x1(h-1)+(1-y1(h))*100);
@for(posisi(h)|h#ge#2:x2(h)>=x3(h-1)-(1-y2(h))*100);
@for(posisi(h)|h#ge#2:x2(h)<=x3(h-1)+(1-y2(h))*100);
@for(posisi(h)|h#ge#2:x3(h)>=x2(h-1)-(1-y2(h))*100);
@for(posisi(h)|h#ge#2:x3(h)<=x2(h-1)+(1-y2(h))*100);
@for(posisi(h)|h#ge#2:x4(h)>=x5(h-1)-(1-y3(h))*100);
@for(posisi(h)|h#ge#2:x4(h)<=x5(h-1)+(1-y3(h))*100);
@for(posisi(h)|h#ge#2:x5(h)>=x4(h-1)-(1-y3(h))*100);
@for(posisi(h)|h#ge#2:x5(h)<=x4(h-1)+(1-y3(h))*100);
@for(posisi(h)|h#ge#2:x5(h)>=x6(h-1)-(1-y4(h))*100);
@for(posisi(h)|h#ge#2:x5(h)<=x6(h-1)+(1-y4(h))*100);
@for(posisi(h)|h#ge#2:x6(h)>=x5(h-1)-(1-y4(h))*100);
@for(posisi(h)|h#ge#2:x6(h)<=x5(h-1)+(1-y4(h))*100);
@for(posisi(h)|h#ge#2:x7(h)>=x8(h-1)-(1-y5(h))*100);
@for(posisi(h)|h#ge#2:x7(h)<=x8(h-1)+(1-y5(h))*100);
@for(posisi(h)|h#ge#2:x8(h)>=x7(h-1)-(1-y5(h))*100);
@for(posisi(h)|h#ge#2:x8(h)<=x7(h-1)+(1-y5(h))*100);
@for(posisi(h)|h#ge#2:x8(h)>=x9(h-1)-(1-y6(h))*100);
@for(posisi(h)|h#ge#2:x8(h)<=x9(h-1)+(1-y6(h))*100);
@for(posisi(h)|h#ge#2:x9(h)>=x8(h-1)-(1-y6(h))*100);
@for(posisi(h)|h#ge#2:x9(h)<=x8(h-1)+(1-y6(h))*100);
@for(posisi(h)|h#ge#2:x1(h)>=x4(h-1)-(1-y7(h))*100);
@for(posisi(h)|h#ge#2:x1(h)<=x4(h-1)+(1-y7(h))*100);
@for(posisi(h)|h#ge#2:x4(h)>=x1(h-1)-(1-y7(h))*100);
@for(posisi(h)|h#ge#2:x4(h)<=x1(h-1)+(1-y7(h))*100);
@for(posisi(h)|h#ge#2:x2(h)>=x5(h-1)-(1-y8(h))*100);

```

```

@for (posisi (h) |h#ge#2:x2 (h) <=x5 (h-1) + (1-y8 (h)) *100) ;
@for (posisi (h) |h#ge#2:x5 (h) >=x2 (h-1) - (1-y8 (h)) *100) ;
@for (posisi (h) |h#ge#2:x5 (h) <=x2 (h-1) + (1-y8 (h)) *100) ;
@for (posisi (h) |h#ge#2:x3 (h) >=x6 (h-1) - (1-y9 (h)) *100) ;
@for (posisi (h) |h#ge#2:x3 (h) <=x6 (h-1) + (1-y9 (h)) *100) ;
@for (posisi (h) |h#ge#2:x6 (h) >=x3 (h-1) - (1-y9 (h)) *100) ;
@for (posisi (h) |h#ge#2:x6 (h) <=x3 (h-1) + (1-y9 (h)) *100) ;
@for (posisi (h) |h#ge#2:x4 (h) >=x7 (h-1) - (1-y10 (h)) *100) ;
@for (posisi (h) |h#ge#2:x4 (h) <=x7 (h-1) + (1-y10 (h)) *100) ;
@for (posisi (h) |h#ge#2:x7 (h) >=x4 (h-1) - (1-y10 (h)) *100) ;
@for (posisi (h) |h#ge#2:x7 (h) <=x4 (h-1) + (1-y10 (h)) *100) ;
@for (posisi (h) |h#ge#2:x5 (h) >=x8 (h-1) - (1-y11 (h)) *100) ;
@for (posisi (h) |h#ge#2:x5 (h) <=x8 (h-1) + (1-y11 (h)) *100) ;
@for (posisi (h) |h#ge#2:x8 (h) >=x5 (h-1) - (1-y11 (h)) *100) ;
@for (posisi (h) |h#ge#2:x8 (h) <=x5 (h-1) + (1-y11 (h)) *100) ;
@for (posisi (h) |h#ge#2:x6 (h) >=x9 (h-1) - (1-y12 (h)) *100) ;
@for (posisi (h) |h#ge#2:x6 (h) <=x9 (h-1) + (1-y12 (h)) *100) ;
@for (posisi (h) |h#ge#2:x9 (h) >=x6 (h-1) - (1-y12 (h)) *100) ;
@for (posisi (h) |h#ge#2:x9 (h) <=x6 (h-1) + (1-y12 (h)) *100) ;

@for (posisi (h) |h#ge#2:x1 (h) >=x1 (h-1) - (y1 (h) +y7 (h)) *
100) ;
@for (posisi (h) |h#ge#2:x1 (h) <=x1 (h-1) + (y1 (h) +y7 (h)) *
100) ;
@for (posisi (h) |h#ge#2:x2 (h) >=x2 (h-1) - (y1 (h) +y2 (h) +
y8 (h)) *100) ;
@for (posisi (h) |h#ge#2:x2 (h) <=x2 (h-1) + (y1 (h) +y2 (h) +
y8 (h)) *100) ;
@for (posisi (h) |h#ge#2:x3 (h) >=x3 (h-1) - (y2 (h) +y9 (h)) *
100) ;
@for (posisi (h) |h#ge#2:x3 (h) <=x3 (h-1) + (y2 (h) +y9 (h)) *
100) ;
@for (posisi (h) |h#ge#2:x4 (h) >=x4 (h-1) - (y3 (h) +y7 (h) +
y10 (h)) *100) ;
@for (posisi (h) |h#ge#2:x4 (h) <=x4 (h-1) + (y3 (h) +y7 (h) +
y10 (h)) *100) ;
@for (posisi (h) |h#ge#2:x5 (h) >=x5 (h-1) - (y3 (h) +y4 (h) +
y8 (h) +y11 (h)) *100) ;
@for (posisi (h) |h#ge#2:x5 (h) <=x5 (h-1) + (y3 (h) +y4 (h) +
y8 (h) +y11 (h)) *100) ;
@for (posisi (h) |h#ge#2:x6 (h) >=x6 (h-1) - (y4 (h) +y9 (h) +
y12 (h)) *100) ;
@for (posisi (h) |h#ge#2:x6 (h) <=x6 (h-1) + (y4 (h) +y9 (h) +
y12 (h)) *100) ;
@for (posisi (h) |h#ge#2:x7 (h) >=x7 (h-1) - (y10 (h) +y5 (h)) *
100) ;
@for (posisi (h) |h#ge#2:x7 (h) <=x7 (h-1) + (y10 (h) +y5 (h)) *
100) ;

```



```

@for (posisi (h) |h#ge#2:x8 (h) >=x8 (h-1) - (y5 (h)+y6 (h) +
y11 (h)) *100);
@for (posisi (h) |h#ge#2:x8 (h) <=x8 (h-1) + (y5 (h)+y6 (h) +
y11 (h)) *100);
@for (posisi (h) |h#ge#2:x9 (h) >=x9 (h-1) - (y6 (h)+y12 (h)) *
100);
@for (posisi (h) |h#ge#2:x9 (h) <=x9 (h-1) + (y6 (h)+y12 (h)) *
100);
!hanya blok kosong yang dapat ditukar dengan blok lain;
@for (posisi (h) |h#ge#2:16*y1 (h) <=x1 (h-1)+x2 (h-1));
@for (posisi (h) |h#ge#2:16*y2 (h) <=x2 (h-1)+x3 (h-1));
@for (posisi (h) |h#ge#2:16*y3 (h) <=x4 (h-1)+x5 (h-1));
@for (posisi (h) |h#ge#2:16*y4 (h) <=x5 (h-1)+x6 (h-1));
@for (posisi (h) |h#ge#2:16*y5 (h) <=x7 (h-1)+x8 (h-1));
@for (posisi (h) |h#ge#2:16*y6 (h) <=x8 (h-1)+x9 (h-1));
@for (posisi (h) |h#ge#2:16*y7 (h) <=x1 (h-1)+x4 (h-1));
@for (posisi (h) |h#ge#2:16*y8 (h) <=x2 (h-1)+x5 (h-1));
@for (posisi (h) |h#ge#2:16*y9 (h) <=x3 (h-1)+x6 (h-1));
@for (posisi (h) |h#ge#2:16*y10 (h) <=x4 (h-1)+x7 (h-1));
@for (posisi (h) |h#ge#2:16*y11 (h) <=x5 (h-1)+x8 (h-1));
@for (posisi (h) |h#ge#2:16*y12 (h) <=x6 (h-1)+x9 (h-1));
!hanya ada 1 pertukaran setiap langkahnya;
@for (posisi (h) |h#ge#2:(y1 (h)+y2 (h)+y3 (h)+y4 (h)+y5 (h) +
y6 (h)+y7 (h)+y8 (h)+y9 (h)+y10 (h)+y11 (h)+y12 (h)) <=1);
!langkah yang diambil harus bersambung;
@for (posisi (h) |h#ge#3:(y1 (h)+y2 (h)+y3 (h)+y4 (h)+y5 (h) +
y6 (h)+y7 (h)+y8 (h)+y9 (h)+y10 (h)+y11 (h)+y12 (h)) <=
(y1 (h-1)+y2 (h-1)+y3 (h-1)+y4 (h-1)+y5 (h-1)+y6 (h-1) +
y7 (h-1)+y8 (h-1)+y9 (h-1)+y10 (h-1)+y11 (h-1)+y12 (h-1)));
!tidak ada y yang dipilih 2 kali berturut-turut;
@for (posisi (h) |h#ge#3:y1 (h) <=1-y1 (h-1));
@for (posisi (h) |h#ge#3:y2 (h) <=1-y2 (h-1));
@for (posisi (h) |h#ge#3:y3 (h) <=1-y3 (h-1));
@for (posisi (h) |h#ge#3:y4 (h) <=1-y4 (h-1));
@for (posisi (h) |h#ge#3:y5 (h) <=1-y5 (h-1));
@for (posisi (h) |h#ge#3:y6 (h) <=1-y6 (h-1));
@for (posisi (h) |h#ge#3:y7 (h) <=1-y7 (h-1));
@for (posisi (h) |h#ge#3:y8 (h) <=1-y8 (h-1));
@for (posisi (h) |h#ge#3:y9 (h) <=1-y9 (h-1));
@for (posisi (h) |h#ge#3:y10 (h) <=1-y10 (h-1));
@for (posisi (h) |h#ge#3:y11 (h) <=1-y11 (h-1));
@for (posisi (h) |h#ge#3:y12 (h) <=1-y12 (h-1));
!y biner dan x berupa nilai integer;
@for (posisi (h) |h#ge#2:@bin (y1 (h)));
@for (posisi (h) |h#ge#2:@bin (y2 (h)));
@for (posisi (h) |h#ge#2:@bin (y3 (h)));
@for (posisi (h) |h#ge#2:@bin (y4 (h)));
@for (posisi (h) |h#ge#2:@bin (y5 (h)));
@for (posisi (h) |h#ge#2:@bin (y6 (h)));

```

```

@for (posisi (h) |h#ge#2:@bin (y7 (h) ) ) ;
@for (posisi (h) |h#ge#2:@bin (y8 (h) ) ) ;
@for (posisi (h) |h#ge#2:@bin (y9 (h) ) ) ;
@for (posisi (h) |h#ge#2:@bin (y10 (h) ) ) ;
@for (posisi (h) |h#ge#2:@bin (y11 (h) ) ) ;
@for (posisi (h) |h#ge#2:@bin (y12 (h) ) ) ;
@for (posisi (h) |h#ge#2:@gin (x1 (h) ) ) ;
@for (posisi (h) |h#ge#2:@gin (x2 (h) ) ) ;
@for (posisi (h) |h#ge#2:@gin (x3 (h) ) ) ;
@for (posisi (h) |h#ge#2:@gin (x4 (h) ) ) ;
@for (posisi (h) |h#ge#2:@gin (x5 (h) ) ) ;
@for (posisi (h) |h#ge#2:@gin (x6 (h) ) ) ;
@for (posisi (h) |h#ge#2:@gin (x7 (h) ) ) ;
@for (posisi (h) |h#ge#2:@gin (x8 (h) ) ) ;
@for (posisi (h) |h#ge#2:@gin (x9 (h) ) ) ;

end
    
```



```

Global optimal solution found.
Objective value:                10.00000
Objective bound:                10.00000
Infeasibilities:                0.000000
Extended solver steps:          48416
Total solver iterations:        2337021
    
```

Variable	Value	Reduced Cost
Z	10.00000	0.000000

X2 (18)	8.000000	0.000000
X2 (19)	8.000000	0.000000
X2 (20)	8.000000	0.000000
X2 (21)	15.00000	0.000000
X2 (22)	3.000000	0.000000
X2 (23)	3.000000	0.000000
X2 (24)	3.000000	0.000000
X2 (25)	3.000000	0.000000
X2 (26)	3.000000	0.000000
X2 (27)	15.00000	0.000000
X2 (28)	2.000000	0.000000
X2 (29)	2.000000	0.000000
X2 (30)	2.000000	0.000000
X2 (31)	2.000000	0.000000
X2 (32)	2.000000	2.000000
X3 (1)	6.000000	0.000000
X3 (2)	6.000000	0.000000
X3 (3)	6.000000	0.000000
X3 (4)	6.000000	0.000000
X3 (5)	6.000000	0.000000
X3 (6)	6.000000	0.000000
X3 (7)	6.000000	0.000000
X3 (8)	6.000000	0.000000
X3 (9)	6.000000	0.000000
X3 (10)	6.000000	0.000000
X3 (11)	6.000000	0.000000
X3 (12)	6.000000	0.000000
X3 (13)	6.000000	0.000000
X3 (14)	6.000000	0.000000
X3 (15)	6.000000	0.000000
X3 (16)	6.000000	0.000000
X3 (17)	6.000000	0.000000
X3 (18)	6.000000	0.000000
X3 (19)	6.000000	0.000000
X3 (20)	15.00000	0.000000
X3 (21)	8.000000	0.000000
X3 (22)	8.000000	0.000000
X3 (23)	8.000000	0.000000
X3 (24)	8.000000	0.000000
X3 (25)	8.000000	0.000000
X3 (26)	15.00000	0.000000
X3 (27)	3.000000	0.000000
X3 (28)	3.000000	0.000000
X3 (29)	3.000000	0.000000
X3 (30)	3.000000	0.000000
X3 (31)	3.000000	0.000000
X3 (32)	3.000000	1.000000
X4 (1)	15.00000	0.000000
X4 (2)	4.000000	0.000000

X5 (20)	3.000000	0.000000
X5 (21)	3.000000	0.000000
X5 (22)	15.00000	0.000000
X5 (23)	2.000000	0.000000
X5 (24)	2.000000	0.000000
X5 (25)	2.000000	0.000000
X5 (26)	2.000000	0.000000
X5 (27)	2.000000	0.000000
X5 (28)	15.00000	0.000000
X5 (29)	8.000000	0.000000
X5 (30)	8.000000	0.000000
X5 (31)	8.000000	0.000000
X5 (32)	8.000000	0.000000
X6 (1)	1.000000	0.000000
X6 (2)	1.000000	0.000000
X6 (3)	15.00000	0.000000
X6 (4)	3.000000	0.000000
X6 (5)	3.000000	0.000000
X6 (6)	3.000000	0.000000
X6 (7)	3.000000	0.000000
X6 (8)	3.000000	0.000000
X6 (9)	15.00000	0.000000
X6 (10)	5.000000	0.000000
X6 (11)	5.000000	0.000000
X6 (12)	5.000000	0.000000
X6 (13)	5.000000	0.000000
X6 (14)	5.000000	0.000000
X6 (15)	5.000000	0.000000
X6 (16)	5.000000	0.000000
X6 (17)	5.000000	0.000000
X6 (18)	5.000000	0.000000
X6 (19)	15.00000	0.000000
X6 (20)	6.000000	0.000000
X6 (21)	6.000000	0.000000
X6 (22)	6.000000	0.000000
X6 (23)	6.000000	0.000000
X6 (24)	6.000000	0.000000
X6 (25)	15.00000	0.000000
X6 (26)	8.000000	0.000000
X6 (27)	8.000000	0.000000
X6 (28)	8.000000	0.000000
X6 (29)	15.00000	0.000000
X6 (30)	15.00000	0.000000
X6 (31)	15.00000	0.000000
X6 (32)	15.00000	0.000000
X7 (1)	2.000000	0.000000
X7 (2)	2.000000	0.000000
X7 (3)	2.000000	0.000000
X7 (4)	2.000000	0.000000



X7 (5)	2.000000	0.000000
X7 (6)	15.000000	0.000000
X7 (7)	4.000000	0.000000
X7 (8)	4.000000	0.000000
X7 (9)	4.000000	0.000000
X7 (10)	4.000000	0.000000
X7 (11)	4.000000	0.000000
X7 (12)	4.000000	0.000000
X7 (13)	4.000000	0.000000
X7 (14)	4.000000	0.000000
X7 (15)	4.000000	0.000000
X7 (16)	4.000000	0.000000
X7 (17)	4.000000	0.000000
X7 (18)	4.000000	0.000000
X7 (19)	4.000000	0.000000
X7 (20)	4.000000	0.000000
X7 (21)	4.000000	0.000000
X7 (22)	4.000000	0.000000
X7 (23)	4.000000	0.000000
X7 (24)	4.000000	0.000000
X7 (25)	4.000000	0.000000
X7 (26)	4.000000	0.000000
X7 (27)	4.000000	0.000000
X7 (28)	4.000000	0.000000
X7 (29)	4.000000	0.000000
X7 (30)	4.000000	0.000000
X7 (31)	4.000000	0.000000
X7 (32)	4.000000	0.000000
X8 (1)	5.000000	0.000000
X8 (2)	5.000000	0.000000
X8 (3)	5.000000	0.000000
X8 (4)	5.000000	0.000000
X8 (5)	15.000000	0.000000
X8 (6)	2.000000	0.000000
X8 (7)	2.000000	0.000000
X8 (8)	2.000000	0.000000
X8 (9)	2.000000	0.000000
X8 (10)	2.000000	0.000000
X8 (11)	15.000000	0.000000
X8 (12)	3.000000	0.000000
X8 (13)	3.000000	0.000000
X8 (14)	3.000000	0.000000
X8 (15)	3.000000	0.000000
X8 (16)	3.000000	0.000000
X8 (17)	15.000000	0.000000
X8 (18)	2.000000	0.000000
X8 (19)	2.000000	0.000000
X8 (20)	2.000000	0.000000
X8 (21)	2.000000	0.000000

X8 (22)	2.000000	0.000000
X8 (23)	15.000000	0.000000
X8 (24)	5.000000	0.000000
X8 (25)	5.000000	0.000000
X8 (26)	5.000000	0.000000
X8 (27)	5.000000	0.000000
X8 (28)	5.000000	0.000000
X8 (29)	5.000000	0.000000
X8 (30)	5.000000	0.000000
X8 (31)	5.000000	0.000000
X8 (32)	5.000000	0.000000
X9 (1)	3.000000	0.000000
X9 (2)	3.000000	0.000000
X9 (3)	3.000000	0.000000
X9 (4)	15.000000	0.000000
X9 (5)	5.000000	0.000000
X9 (6)	5.000000	0.000000
X9 (7)	5.000000	0.000000
X9 (8)	5.000000	0.000000
X9 (9)	5.000000	0.000000
X9 (10)	15.000000	0.000000
X9 (11)	2.000000	0.000000
X9 (12)	2.000000	0.000000
X9 (13)	2.000000	0.000000
X9 (14)	2.000000	0.000000
X9 (15)	2.000000	0.000000
X9 (16)	2.000000	0.000000
X9 (17)	2.000000	0.000000
X9 (18)	15.000000	0.000000
X9 (19)	5.000000	0.000000
X9 (20)	5.000000	0.000000
X9 (21)	5.000000	0.000000
X9 (22)	5.000000	0.000000
X9 (23)	5.000000	0.000000
X9 (24)	15.000000	0.000000
X9 (25)	6.000000	0.000000
X9 (26)	6.000000	0.000000
X9 (27)	6.000000	0.000000
X9 (28)	6.000000	0.000000
X9 (29)	6.000000	0.000000
X9 (30)	6.000000	0.000000
X9 (31)	6.000000	0.000000
X9 (32)	6.000000	0.000000
Y1 (1)	0.000000	0.000000
Y1 (2)	0.000000	0.000000
Y1 (3)	0.000000	0.000000
Y1 (4)	0.000000	0.000000
Y1 (5)	0.000000	0.000000
Y1 (6)	0.000000	0.000000



Y2 (24)	0.000000	0.000000
Y2 (25)	0.000000	0.000000
Y2 (26)	0.000000	0.000000
Y2 (27)	1.000000	0.000000
Y2 (28)	0.000000	0.000000
Y2 (29)	0.000000	0.000000
Y2 (30)	0.000000	0.000000
Y2 (31)	0.000000	0.000000
Y2 (32)	0.000000	0.000000
Y3 (1)	0.000000	0.000000
Y3 (2)	1.000000	0.000000
Y3 (3)	0.000000	0.000000
Y3 (4)	0.000000	0.000000
Y3 (5)	0.000000	0.000000
Y3 (6)	0.000000	0.000000
Y3 (7)	0.000000	0.000000
Y3 (8)	1.000000	0.000000
Y3 (9)	0.000000	0.000000
Y3 (10)	0.000000	0.000000
Y3 (11)	0.000000	0.000000
Y3 (12)	0.000000	0.000000
Y3 (13)	0.000000	0.000000
Y3 (14)	0.000000	0.000000
Y3 (15)	0.000000	0.000000
Y3 (16)	1.000000	0.000000
Y3 (17)	0.000000	0.000000
Y3 (18)	0.000000	0.000000
Y3 (19)	0.000000	0.000000
Y3 (20)	0.000000	0.000000
Y3 (21)	0.000000	0.000000
Y3 (22)	0.000000	0.000000
Y3 (23)	0.000000	0.000000
Y3 (24)	0.000000	0.000000
Y3 (25)	0.000000	0.000000
Y3 (26)	0.000000	0.000000
Y3 (27)	0.000000	0.000000
Y3 (28)	0.000000	0.000000
Y3 (29)	0.000000	0.000000
Y3 (30)	0.000000	0.000000
Y3 (31)	0.000000	0.000000
Y3 (32)	0.000000	0.000000
Y4 (1)	0.000000	0.000000
Y4 (2)	0.000000	0.000000
Y4 (3)	1.000000	0.000000
Y4 (4)	0.000000	0.000000
Y4 (5)	0.000000	0.000000
Y4 (6)	0.000000	0.000000
Y4 (7)	0.000000	0.000000
Y4 (8)	0.000000	0.000000



Y4 (9)	1.000000	0.000000
Y4 (10)	0.000000	0.000000
Y4 (11)	0.000000	0.000000
Y4 (12)	0.000000	0.000000
Y4 (13)	0.000000	0.000000
Y4 (14)	0.000000	0.000000
Y4 (15)	0.000000	0.000000
Y4 (16)	0.000000	0.000000
Y4 (17)	0.000000	0.000000
Y4 (18)	0.000000	0.000000
Y4 (19)	0.000000	0.000000
Y4 (20)	0.000000	0.000000
Y4 (21)	0.000000	0.000000
Y4 (22)	0.000000	0.000000
Y4 (23)	0.000000	0.000000
Y4 (24)	0.000000	0.000000
Y4 (25)	0.000000	0.000000
Y4 (26)	0.000000	0.000000
Y4 (27)	0.000000	0.000000
Y4 (28)	0.000000	0.000000
Y4 (29)	1.000000	0.000000
Y4 (30)	0.000000	0.000000
Y4 (31)	0.000000	0.000000
Y4 (32)	0.000000	0.000000
Y5 (1)	0.000000	0.000000
Y5 (2)	0.000000	0.000000
Y5 (3)	0.000000	0.000000
Y5 (4)	0.000000	0.000000
Y5 (5)	0.000000	0.000000
Y5 (6)	1.000000	0.000000
Y5 (7)	0.000000	0.000000
Y5 (8)	0.000000	0.000000
Y5 (9)	0.000000	0.000000
Y5 (10)	0.000000	0.000000
Y5 (11)	0.000000	0.000000
Y5 (12)	0.000000	0.000000
Y5 (13)	0.000000	0.000000
Y5 (14)	0.000000	0.000000
Y5 (15)	0.000000	0.000000
Y5 (16)	0.000000	0.000000
Y5 (17)	0.000000	0.000000
Y5 (18)	0.000000	0.000000
Y5 (19)	0.000000	0.000000
Y5 (20)	0.000000	0.000000
Y5 (21)	0.000000	0.000000
Y5 (22)	0.000000	0.000000
Y5 (23)	0.000000	0.000000
Y5 (24)	0.000000	0.000000
Y5 (25)	0.000000	0.000000

Y5 (26)	0.000000	0.000000
Y5 (27)	0.000000	0.000000
Y5 (28)	0.000000	0.000000
Y5 (29)	0.000000	0.000000
Y5 (30)	0.000000	0.000000
Y5 (31)	0.000000	0.000000
Y5 (32)	0.000000	0.000000
Y6 (1)	0.000000	0.000000
Y6 (2)	0.000000	0.000000
Y6 (3)	0.000000	0.000000
Y6 (4)	0.000000	0.000000
Y6 (5)	1.000000	0.000000
Y6 (6)	0.000000	0.000000
Y6 (7)	0.000000	0.000000
Y6 (8)	0.000000	0.000000
Y6 (9)	0.000000	0.000000
Y6 (10)	0.000000	0.000000
Y6 (11)	1.000000	0.000000
Y6 (12)	0.000000	0.000000
Y6 (13)	0.000000	0.000000
Y6 (14)	0.000000	0.000000
Y6 (15)	0.000000	0.000000
Y6 (16)	0.000000	0.000000
Y6 (17)	0.000000	0.000000
Y6 (18)	1.000000	0.000000
Y6 (19)	0.000000	0.000000
Y6 (20)	0.000000	0.000000
Y6 (21)	0.000000	0.000000
Y6 (22)	0.000000	0.000000
Y6 (23)	0.000000	0.000000
Y6 (24)	1.000000	0.000000
Y6 (25)	0.000000	0.000000
Y6 (26)	0.000000	0.000000
Y6 (27)	0.000000	0.000000
Y6 (28)	0.000000	0.000000
Y6 (29)	0.000000	0.000000
Y6 (30)	0.000000	0.000000
Y6 (31)	0.000000	0.000000
Y6 (32)	0.000000	0.000000
Y7 (1)	0.000000	0.000000
Y7 (2)	0.000000	0.000000
Y7 (3)	0.000000	0.000000
Y7 (4)	0.000000	0.000000
Y7 (5)	0.000000	0.000000
Y7 (6)	0.000000	0.000000
Y7 (7)	0.000000	0.000000
Y7 (8)	0.000000	0.000000
Y7 (9)	0.000000	0.000000
Y7 (10)	0.000000	0.000000



Y7 (11)	0.000000	0.000000
Y7 (12)	0.000000	0.000000
Y7 (13)	0.000000	0.000000
Y7 (14)	0.000000	0.000000
Y7 (15)	1.000000	0.000000
Y7 (16)	0.000000	0.000000
Y7 (17)	0.000000	0.000000
Y7 (18)	0.000000	0.000000
Y7 (19)	0.000000	0.000000
Y7 (20)	0.000000	0.000000
Y7 (21)	0.000000	0.000000
Y7 (22)	0.000000	0.000000
Y7 (23)	0.000000	0.000000
Y7 (24)	0.000000	0.000000
Y7 (25)	0.000000	0.000000
Y7 (26)	0.000000	0.000000
Y7 (27)	0.000000	0.000000
Y7 (28)	0.000000	0.000000
Y7 (29)	0.000000	0.000000
Y7 (30)	0.000000	0.000000
Y7 (31)	0.000000	0.000000
Y7 (32)	0.000000	0.000000
Y8 (1)	0.000000	0.000000
Y8 (2)	0.000000	0.000000
Y8 (3)	0.000000	0.000000
Y8 (4)	0.000000	0.000000
Y8 (5)	0.000000	0.000000
Y8 (6)	0.000000	0.000000
Y8 (7)	0.000000	0.000000
Y8 (8)	0.000000	0.000000
Y8 (9)	0.000000	0.000000
Y8 (10)	0.000000	0.000000
Y8 (11)	0.000000	0.000000
Y8 (12)	0.000000	0.000000
Y8 (13)	1.000000	0.000000
Y8 (14)	0.000000	0.000000
Y8 (15)	0.000000	0.000000
Y8 (16)	0.000000	0.000000
Y8 (17)	0.000000	0.000000
Y8 (18)	0.000000	0.000000
Y8 (19)	0.000000	0.000000
Y8 (20)	0.000000	0.000000
Y8 (21)	0.000000	0.000000
Y8 (22)	1.000000	0.000000
Y8 (23)	0.000000	0.000000
Y8 (24)	0.000000	0.000000
Y8 (25)	0.000000	0.000000
Y8 (26)	0.000000	0.000000
Y8 (27)	0.000000	0.000000

Halaman ini adalah bagian dari dokumen resmi yang diterbitkan oleh IPB University dan merupakan hak milik dari IPB University. Semua hak cipta dan hak lainnya dilindungi undang-undang. Penggunaan yang tidak sah akan dikenakan sanksi hukum yang berlaku. IPB University tidak bertanggung jawab atas kerugian yang timbul akibat penggunaan yang tidak sah. IPB University tidak bertanggung jawab atas kerugian yang timbul akibat penggunaan yang tidak sah. IPB University tidak bertanggung jawab atas kerugian yang timbul akibat penggunaan yang tidak sah.

Y8 (28)	1.000000	0.000000
Y8 (29)	0.000000	0.000000
Y8 (30)	0.000000	0.000000
Y8 (31)	0.000000	0.000000
Y8 (32)	0.000000	0.000000
Y9 (1)	0.000000	0.000000
Y9 (2)	0.000000	0.000000
Y9 (3)	0.000000	0.000000
Y9 (4)	0.000000	0.000000
Y9 (5)	0.000000	0.000000
Y9 (6)	0.000000	0.000000
Y9 (7)	0.000000	0.000000
Y9 (8)	0.000000	0.000000
Y9 (9)	0.000000	0.000000
Y9 (10)	0.000000	0.000000
Y9 (11)	0.000000	0.000000
Y9 (12)	0.000000	0.000000
Y9 (13)	0.000000	0.000000
Y9 (14)	0.000000	0.000000
Y9 (15)	0.000000	0.000000
Y9 (16)	0.000000	0.000000
Y9 (17)	0.000000	0.000000
Y9 (18)	0.000000	0.000000
Y9 (19)	0.000000	0.000000
Y9 (20)	1.000000	0.000000
Y9 (21)	0.000000	0.000000
Y9 (22)	0.000000	0.000000
Y9 (23)	0.000000	0.000000
Y9 (24)	0.000000	0.000000
Y9 (25)	0.000000	0.000000
Y9 (26)	1.000000	0.000000
Y9 (27)	0.000000	0.000000
Y9 (28)	0.000000	0.000000
Y9 (29)	0.000000	0.000000
Y9 (30)	0.000000	0.000000
Y9 (31)	0.000000	0.000000
Y9 (32)	0.000000	0.000000
Y10 (1)	0.000000	0.000000
Y10 (2)	0.000000	0.000000
Y10 (3)	0.000000	0.000000
Y10 (4)	0.000000	0.000000
Y10 (5)	0.000000	0.000000
Y10 (6)	0.000000	0.000000
Y10 (7)	1.000000	0.000000
Y10 (8)	0.000000	0.000000
Y10 (9)	0.000000	0.000000
Y10 (10)	0.000000	0.000000
Y10 (11)	0.000000	0.000000
Y10 (12)	0.000000	0.000000

Y11 (30)	0.000000	0.000000
Y11 (31)	0.000000	0.000000
Y11 (32)	0.000000	0.000000
Y12 (1)	0.000000	0.000000
Y12 (2)	0.000000	0.000000
Y12 (3)	0.000000	0.000000
Y12 (4)	1.000000	0.000000
Y12 (5)	0.000000	0.000000
Y12 (6)	0.000000	0.000000
Y12 (7)	0.000000	0.000000
Y12 (8)	0.000000	0.000000
Y12 (9)	0.000000	0.000000
Y12 (10)	1.000000	0.000000
Y12 (11)	0.000000	0.000000
Y12 (12)	0.000000	0.000000
Y12 (13)	0.000000	0.000000
Y12 (14)	0.000000	0.000000
Y12 (15)	0.000000	0.000000
Y12 (16)	0.000000	0.000000
Y12 (17)	0.000000	0.000000
Y12 (18)	0.000000	0.000000
Y12 (19)	1.000000	0.000000
Y12 (20)	0.000000	0.000000
Y12 (21)	0.000000	0.000000
Y12 (22)	0.000000	0.000000
Y12 (23)	0.000000	0.000000
Y12 (24)	0.000000	0.000000
Y12 (25)	1.000000	0.000000
Y12 (26)	0.000000	0.000000
Y12 (27)	0.000000	0.000000
Y12 (28)	0.000000	0.000000
Y12 (29)	0.000000	0.000000
Y12 (30)	0.000000	0.000000
Y12 (31)	0.000000	0.000000
Y12 (32)	0.000000	0.000000

Lampiran 3 Ilustrasi perubahan susunan berdasarkan solusi Model 1

<table border="1"><tr><td>8</td><td>7</td><td>6</td></tr><tr><td></td><td>4</td><td>1</td></tr><tr><td>2</td><td>5</td><td>3</td></tr></table>	8	7	6		4	1	2	5	3	<table border="1"><tr><td>8</td><td>7</td><td>6</td></tr><tr><td>4</td><td></td><td>1</td></tr><tr><td>2</td><td>5</td><td>3</td></tr></table> y ³	8	7	6	4		1	2	5	3	<table border="1"><tr><td>8</td><td>7</td><td>6</td></tr><tr><td>4</td><td>1</td><td></td></tr><tr><td>2</td><td>5</td><td>3</td></tr></table> y ⁴	8	7	6	4	1		2	5	3	<table border="1"><tr><td>8</td><td>7</td><td>6</td></tr><tr><td>4</td><td>1</td><td>3</td></tr><tr><td>2</td><td>5</td><td></td></tr></table> y ¹²	8	7	6	4	1	3	2	5	
8	7	6																																					
	4	1																																					
2	5	3																																					
8	7	6																																					
4		1																																					
2	5	3																																					
8	7	6																																					
4	1																																						
2	5	3																																					
8	7	6																																					
4	1	3																																					
2	5																																						
<table border="1"><tr><td>8</td><td>7</td><td>6</td></tr><tr><td>4</td><td>1</td><td>3</td></tr><tr><td>2</td><td></td><td>5</td></tr></table> y ⁶	8	7	6	4	1	3	2		5	<table border="1"><tr><td>8</td><td>7</td><td>6</td></tr><tr><td>4</td><td>1</td><td>3</td></tr><tr><td></td><td>2</td><td>5</td></tr></table> y ⁵	8	7	6	4	1	3		2	5	<table border="1"><tr><td>8</td><td>7</td><td>6</td></tr><tr><td></td><td>1</td><td>3</td></tr><tr><td>4</td><td>2</td><td>5</td></tr></table> y ¹⁰	8	7	6		1	3	4	2	5	<table border="1"><tr><td>8</td><td>7</td><td>6</td></tr><tr><td>1</td><td></td><td>3</td></tr><tr><td>4</td><td>2</td><td>5</td></tr></table> y ³	8	7	6	1		3	4	2	5
8	7	6																																					
4	1	3																																					
2		5																																					
8	7	6																																					
4	1	3																																					
	2	5																																					
8	7	6																																					
	1	3																																					
4	2	5																																					
8	7	6																																					
1		3																																					
4	2	5																																					
<table border="1"><tr><td>8</td><td>7</td><td>6</td></tr><tr><td>1</td><td>3</td><td></td></tr><tr><td>4</td><td>2</td><td>5</td></tr></table> y ⁴	8	7	6	1	3		4	2	5	<table border="1"><tr><td>8</td><td>7</td><td>6</td></tr><tr><td>1</td><td>3</td><td>5</td></tr><tr><td>4</td><td>2</td><td></td></tr></table> y ¹²	8	7	6	1	3	5	4	2		<table border="1"><tr><td>8</td><td>7</td><td>6</td></tr><tr><td>1</td><td>3</td><td>5</td></tr><tr><td>4</td><td></td><td>2</td></tr></table> y ⁶	8	7	6	1	3	5	4		2	<table border="1"><tr><td>8</td><td>7</td><td>6</td></tr><tr><td>1</td><td></td><td>5</td></tr><tr><td>4</td><td>3</td><td>2</td></tr></table> y ¹¹	8	7	6	1		5	4	3	2
8	7	6																																					
1	3																																						
4	2	5																																					
8	7	6																																					
1	3	5																																					
4	2																																						
8	7	6																																					
1	3	5																																					
4		2																																					
8	7	6																																					
1		5																																					
4	3	2																																					
<table border="1"><tr><td>8</td><td></td><td>6</td></tr><tr><td>1</td><td>7</td><td>5</td></tr><tr><td>4</td><td>3</td><td>2</td></tr></table> y ⁸	8		6	1	7	5	4	3	2	<table border="1"><tr><td></td><td>8</td><td>6</td></tr><tr><td>1</td><td>7</td><td>5</td></tr><tr><td>4</td><td>3</td><td>2</td></tr></table> y ¹		8	6	1	7	5	4	3	2	<table border="1"><tr><td>1</td><td>8</td><td>6</td></tr><tr><td></td><td>7</td><td>5</td></tr><tr><td>4</td><td>3</td><td>2</td></tr></table> y ⁷	1	8	6		7	5	4	3	2	<table border="1"><tr><td>1</td><td>8</td><td>6</td></tr><tr><td>7</td><td></td><td>5</td></tr><tr><td>4</td><td>3</td><td>2</td></tr></table> y ³	1	8	6	7		5	4	3	2
8		6																																					
1	7	5																																					
4	3	2																																					
	8	6																																					
1	7	5																																					
4	3	2																																					
1	8	6																																					
	7	5																																					
4	3	2																																					
1	8	6																																					
7		5																																					
4	3	2																																					
<table border="1"><tr><td>1</td><td>8</td><td>6</td></tr><tr><td>7</td><td>3</td><td>5</td></tr><tr><td>4</td><td></td><td>2</td></tr></table> y ¹¹	1	8	6	7	3	5	4		2	<table border="1"><tr><td>1</td><td>8</td><td>6</td></tr><tr><td>7</td><td>3</td><td>5</td></tr><tr><td>4</td><td>2</td><td></td></tr></table> y ⁶	1	8	6	7	3	5	4	2		<table border="1"><tr><td>1</td><td>8</td><td>6</td></tr><tr><td>7</td><td>3</td><td></td></tr><tr><td>4</td><td>2</td><td>5</td></tr></table> y ¹²	1	8	6	7	3		4	2	5	<table border="1"><tr><td>1</td><td>8</td><td></td></tr><tr><td>7</td><td>3</td><td>6</td></tr><tr><td>4</td><td>2</td><td>5</td></tr></table> y ⁹	1	8		7	3	6	4	2	5
1	8	6																																					
7	3	5																																					
4		2																																					
1	8	6																																					
7	3	5																																					
4	2																																						
1	8	6																																					
7	3																																						
4	2	5																																					
1	8																																						
7	3	6																																					
4	2	5																																					
<table border="1"><tr><td>1</td><td></td><td>8</td></tr><tr><td>7</td><td>3</td><td>6</td></tr><tr><td>4</td><td>2</td><td>5</td></tr></table> y ²	1		8	7	3	6	4	2	5	<table border="1"><tr><td>1</td><td>3</td><td>8</td></tr><tr><td>7</td><td></td><td>6</td></tr><tr><td>4</td><td>2</td><td>5</td></tr></table> y ⁸	1	3	8	7		6	4	2	5	<table border="1"><tr><td>1</td><td>3</td><td>8</td></tr><tr><td>7</td><td>2</td><td>6</td></tr><tr><td>4</td><td></td><td>5</td></tr></table> y ¹¹	1	3	8	7	2	6	4		5	<table border="1"><tr><td>1</td><td>3</td><td>8</td></tr><tr><td>7</td><td>2</td><td>6</td></tr><tr><td>4</td><td>5</td><td></td></tr></table> y ⁶	1	3	8	7	2	6	4	5	
1		8																																					
7	3	6																																					
4	2	5																																					
1	3	8																																					
7		6																																					
4	2	5																																					
1	3	8																																					
7	2	6																																					
4		5																																					
1	3	8																																					
7	2	6																																					
4	5																																						
<table border="1"><tr><td>1</td><td>3</td><td>8</td></tr><tr><td>7</td><td>2</td><td></td></tr><tr><td>4</td><td>5</td><td>6</td></tr></table> y ¹²	1	3	8	7	2		4	5	6	<table border="1"><tr><td>1</td><td>3</td><td></td></tr><tr><td>7</td><td>2</td><td>8</td></tr><tr><td>4</td><td>5</td><td>6</td></tr></table> y ⁹	1	3		7	2	8	4	5	6	<table border="1"><tr><td>1</td><td></td><td>3</td></tr><tr><td>7</td><td>2</td><td>8</td></tr><tr><td>4</td><td>5</td><td>6</td></tr></table> y ²	1		3	7	2	8	4	5	6	<table border="1"><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>7</td><td></td><td>8</td></tr><tr><td>4</td><td>5</td><td>6</td></tr></table> y ⁸	1	2	3	7		8	4	5	6
1	3	8																																					
7	2																																						
4	5	6																																					
1	3																																						
7	2	8																																					
4	5	6																																					
1		3																																					
7	2	8																																					
4	5	6																																					
1	2	3																																					
7		8																																					
4	5	6																																					

glick cpn nur ipb university

IPB University

1	2	3
7	8	
4	5	6

v4

@Hack cipu mitr IPB University

IPB University



IPB University
— Bogor, Indonesia

Halaman 1 dari 1 | 10/10/2024

1. Diambil sebagai bagian dari penelitian yang dilakukan oleh mahasiswa dan dosen di IPB University.
2. Diperoleh sebagai bagian dari penelitian yang dilakukan oleh mahasiswa dan dosen di IPB University.

Lampiran 4 Formulasi dan solusi Model 2 untuk suatu susunan solusi 8-puzzle dari solusi Model 1 (Gambar 10) menggunakan LINGO 11.0

```

model:
sets:
posisi/1..22/:x4,x5,x6,x7,x8,x9,
y3,y4,y5,y6,y10,y11,y12;
endsets

x4(1)=7;x5(1)=8;x6(1)=15;
x7(1)=4;x8(1)=5;x9(1)=6;

max=z;
z=@sum(posisi(h)|h#eq#@size(hari):x4(h)+2*x5(h)+
3*x6(h)+4*x7(h)+5*x8(h)+6*x9(h));
@bnd(125,z,190);

@for(posisi(h)|h#ge#2:x4(h)>=x5(h-1)-(1-y3(h))*100);
@for(posisi(h)|h#ge#2:x4(h)<=x5(h-1)+(1-y3(h))*100);
@for(posisi(h)|h#ge#2:x5(h)>=x4(h-1)-(1-y3(h))*100);
@for(posisi(h)|h#ge#2:x5(h)<=x4(h-1)+(1-y3(h))*100);
@for(posisi(h)|h#ge#2:x5(h)>=x6(h-1)-(1-y4(h))*100);
@for(posisi(h)|h#ge#2:x5(h)<=x6(h-1)+(1-y4(h))*100);
@for(posisi(h)|h#ge#2:x6(h)>=x5(h-1)-(1-y4(h))*100);
@for(posisi(h)|h#ge#2:x6(h)<=x5(h-1)+(1-y4(h))*100);
@for(posisi(h)|h#ge#2:x7(h)>=x8(h-1)-(1-y5(h))*100);
@for(posisi(h)|h#ge#2:x7(h)<=x8(h-1)+(1-y5(h))*100);
@for(posisi(h)|h#ge#2:x8(h)>=x7(h-1)-(1-y5(h))*100);
@for(posisi(h)|h#ge#2:x8(h)<=x7(h-1)+(1-y5(h))*100);
@for(posisi(h)|h#ge#2:x8(h)>=x9(h-1)-(1-y6(h))*100);
@for(posisi(h)|h#ge#2:x8(h)<=x9(h-1)+(1-y6(h))*100);
@for(posisi(h)|h#ge#2:x9(h)>=x8(h-1)-(1-y6(h))*100);
@for(posisi(h)|h#ge#2:x9(h)<=x8(h-1)+(1-y6(h))*100);
@for(posisi(h)|h#ge#2:x1(h)>=x4(h-1)-(1-y7(h))*100);
@for(posisi(h)|h#ge#2:x1(h)<=x4(h-1)+(1-y7(h))*100);
@for(posisi(h)|h#ge#2:x4(h)>=x7(h-1)-(1-y10(h))*100);
@for(posisi(h)|h#ge#2:x4(h)<=x7(h-1)+(1-y10(h))*100);
@for(posisi(h)|h#ge#2:x7(h)>=x4(h-1)-(1-y10(h))*100);
@for(posisi(h)|h#ge#2:x7(h)<=x4(h-1)+(1-y10(h))*100);
@for(posisi(h)|h#ge#2:x5(h)>=x8(h-1)-(1-y11(h))*100);
@for(posisi(h)|h#ge#2:x5(h)<=x8(h-1)+(1-y11(h))*100);
@for(posisi(h)|h#ge#2:x8(h)>=x5(h-1)-(1-y11(h))*100);
@for(posisi(h)|h#ge#2:x8(h)<=x5(h-1)+(1-y11(h))*100);
@for(posisi(h)|h#ge#2:x6(h)>=x9(h-1)-(1-y12(h))*100);
@for(posisi(h)|h#ge#2:x6(h)<=x9(h-1)+(1-y12(h))*100);
@for(posisi(h)|h#ge#2:x9(h)>=x6(h-1)-(1-y12(h))*100);
@for(posisi(h)|h#ge#2:x9(h)<=x6(h-1)+(1-y12(h))*100);

```



```

@for (posisi (h) |h#ge#2:x4 (h) >=x4 (h-1) - (y3 (h)+y10 (h)) *
100);
@for (posisi (h) |h#ge#2:x4 (h) <=x4 (h-1) + (y3 (h)+y10 (h)) *
100);
@for (posisi (h) |h#ge#2:x5 (h) >=x5 (h-1) - (y3 (h)+y4 (h) +
y11 (h)) *100);
@for (posisi (h) |h#ge#2:x5 (h) <=x5 (h-1) + (y3 (h)+y4 (h) +
y11 (h)) *100);
@for (posisi (h) |h#ge#2:x6 (h) >=x6 (h-1) - (y4 (h)+y12 (h)) *
100);
@for (posisi (h) |h#ge#2:x6 (h) <=x6 (h-1) + (y4 (h)+y12 (h)) *
100);
@for (posisi (h) |h#ge#2:x7 (h) >=x7 (h-1) - (y10 (h)+y5 (h)) *
100);
@for (posisi (h) |h#ge#2:x7 (h) <=x7 (h-1) + (y10 (h)+y5 (h)) *
100);
@for (posisi (h) |h#ge#2:x8 (h) >=x8 (h-1) - (y5 (h)+y6 (h) +
y11 (h)) *100);
@for (posisi (h) |h#ge#2:x8 (h) <=x8 (h-1) + (y5 (h)+y6 (h) +
y11 (h)) *100);
@for (posisi (h) |h#ge#2:x9 (h) >=x9 (h-1) - (y6 (h)+y12 (h)) *
100);
@for (posisi (h) |h#ge#2:x9 (h) <=x9 (h-1) + (y6 (h)+y12 (h)) *
100);

@for (posisi (h) |h#ge#2:16*y3 (h) <=x4 (h-1)+x5 (h-1));
@for (posisi (h) |h#ge#2:16*y4 (h) <=x5 (h-1)+x6 (h-1));
@for (posisi (h) |h#ge#2:16*y5 (h) <=x7 (h-1)+x8 (h-1));
@for (posisi (h) |h#ge#2:16*y6 (h) <=x8 (h-1)+x9 (h-1));
@for (posisi (h) |h#ge#2:16*y10 (h) <=x4 (h-1)+x7 (h-1));
@for (posisi (h) |h#ge#2:16*y11 (h) <=x5 (h-1)+x8 (h-1));
@for (posisi (h) |h#ge#2:16*y12 (h) <=x6 (h-1)+x9 (h-1));

@for (posisi (h) |h#ge#2: (y3 (h)+y4 (h)+y5 (h)+y6 (h)+y10 (h) +
y11 (h)+y12 (h)) <=1);

@for (posisi (h) |h#ge#3: (+y3 (h)+y4 (h)+y5 (h)+y6 (h)+y10 (h) +
y11 (h)+y12 (h)) <=
(y3 (h-1)+y4 (h-1)+y5 (h-1)+y6 (h-1)+y10 (h-1)+y11 (h-1) +
y12 (h-1)));

@for (posisi (h) |h#ge#3:y3 (h) <=1-y3 (h-1));
@for (posisi (h) |h#ge#3:y4 (h) <=1-y4 (h-1));
@for (posisi (h) |h#ge#3:y5 (h) <=1-y5 (h-1));
@for (posisi (h) |h#ge#3:y6 (h) <=1-y6 (h-1));
@for (posisi (h) |h#ge#3:y10 (h) <=1-y10 (h-1));
@for (posisi (h) |h#ge#3:y11 (h) <=1-y11 (h-1));
@for (posisi (h) |h#ge#3:y12 (h) <=1-y12 (h-1));

```



```

@for (posisi (h) | h#ge#2: @bin (y3 (h) ) );
@for (posisi (h) | h#ge#2: @bin (y4 (h) ) );
@for (posisi (h) | h#ge#2: @bin (y5 (h) ) );
@for (posisi (h) | h#ge#2: @bin (y6 (h) ) );
@for (posisi (h) | h#ge#2: @bin (y10 (h) ) );
@for (posisi (h) | h#ge#2: @bin (y11 (h) ) );
@for (posisi (h) | h#ge#2: @bin (y12 (h) ) );
@for (posisi (h) | h#ge#2: @gin (x4 (h) ) );
@for (posisi (h) | h#ge#2: @gin (x5 (h) ) );
@for (posisi (h) | h#ge#2: @gin (x6 (h) ) );
@for (posisi (h) | h#ge#2: @gin (x7 (h) ) );
@for (posisi (h) | h#ge#2: @gin (x8 (h) ) );
@for (posisi (h) | h#ge#2: @gin (x9 (h) ) );

end

```

The screenshot shows the LINGO 11.0 Solver Status dialog box. The title bar reads 'LINGO 11.0 Solver Status [barupuzzle2x3baru]'. The dialog is divided into several sections:

- Solver Status:**
 - Model Class: ILP
 - State: Global Opt
 - Objective: 190
 - Infeasibility: 0
 - Iterations: 36592759
- Variables:**
 - Total: 281
 - Nonlinear: 0
 - Integers: 273
- Constraints:**
 - Total: 1170
 - Nonlinear: 0
- Nonzeros:**
 - Total: 3958
 - Nonlinear: 0
- Extended Solver Status:**
 - Solver Type: B-and-B
 - Best Obj: 190
 - Obj Bound: 190
 - Steps: 344501
 - Active: 0
- Generator Memory Used (K):** 188
- Elapsed Runtime (hh:mm:ss):** 04:39:45

At the bottom, there is an 'Update Interval' set to 2, and buttons for 'Interrupt Solver' and 'Close'.

```

Objective value:                190.0000
Objective bound:                190.0000
Infeasibilities:                0.000000
Extended solver steps:         344501
Total solver iterations:       36592759

```

Variable	Value	Reduced Cost
Z	190.0000	0.000000
X4 (1)	7.000000	0.000000
X4 (2)	7.000000	0.000000
X4 (3)	15.00000	0.000000

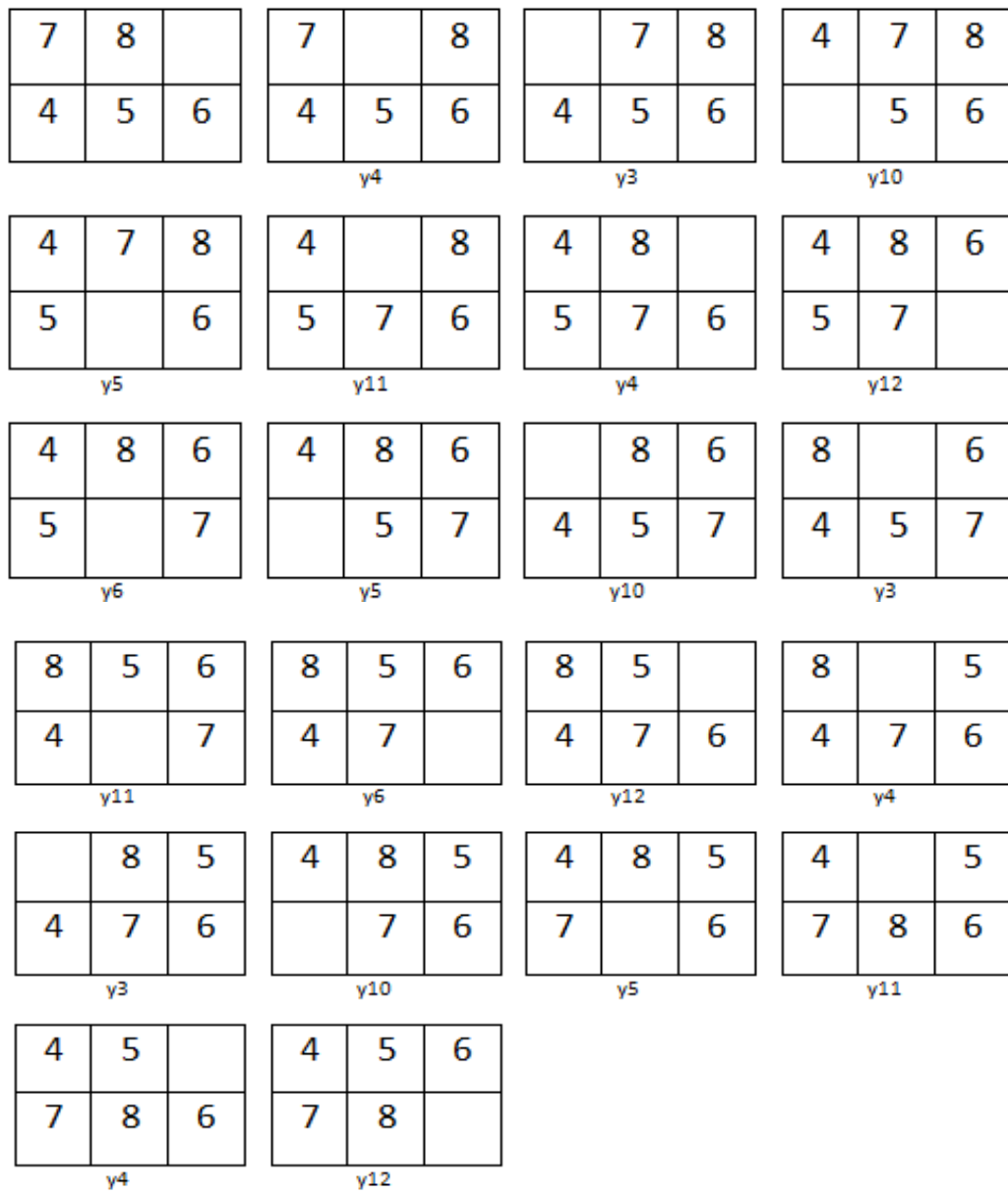
X4 (4)	4.000000	0.000000
X4 (5)	4.000000	0.000000
X4 (6)	4.000000	0.000000
X4 (7)	4.000000	0.000000
X4 (8)	4.000000	0.000000
X4 (9)	4.000000	0.000000
X4 (10)	4.000000	0.000000
X4 (11)	15.00000	0.000000
X4 (12)	8.000000	0.000000
X4 (13)	8.000000	0.000000
X4 (14)	8.000000	0.000000
X4 (15)	8.000000	0.000000
X4 (16)	8.000000	0.000000
X4 (17)	15.00000	0.000000
X4 (18)	4.000000	0.000000
X4 (19)	4.000000	0.000000
X4 (20)	4.000000	0.000000
X4 (21)	4.000000	0.000000
X4 (22)	4.000000	-1.000000
X5 (1)	8.000000	0.000000
X5 (2)	15.00000	0.000000
X5 (3)	7.000000	0.000000
X5 (4)	7.000000	0.000000
X5 (5)	7.000000	0.000000
X5 (6)	15.00000	0.000000
X5 (7)	8.000000	0.000000
X5 (8)	8.000000	0.000000
X5 (9)	8.000000	0.000000
X5 (10)	8.000000	0.000000
X5 (11)	8.000000	0.000000
X5 (12)	15.00000	0.000000
X5 (13)	5.000000	0.000000
X5 (14)	5.000000	0.000000
X5 (15)	5.000000	0.000000
X5 (16)	15.00000	0.000000
X5 (17)	8.000000	0.000000
X5 (18)	8.000000	0.000000
X5 (19)	8.000000	0.000000
X5 (20)	15.00000	0.000000
X5 (21)	5.000000	0.000000
X5 (22)	5.000000	-2.000000
X6 (1)	15.00000	0.000000
X6 (2)	8.000000	0.000000
X6 (3)	8.000000	0.000000
X6 (4)	8.000000	0.000000
X6 (5)	8.000000	0.000000
X6 (6)	8.000000	0.000000
X6 (7)	15.00000	0.000000
X6 (8)	6.000000	0.000000

X8 (14)	7.000000	0.000000
X8 (15)	7.000000	0.000000
X8 (16)	7.000000	0.000000
X8 (17)	7.000000	0.000000
X8 (18)	7.000000	0.000000
X8 (19)	15.00000	0.000000
X8 (20)	8.000000	0.000000
X8 (21)	8.000000	0.000000
X8 (22)	8.000000	-5.000000
X9 (1)	6.000000	0.000000
X9 (2)	6.000000	0.000000
X9 (3)	6.000000	0.000000
X9 (4)	6.000000	0.000000
X9 (5)	6.000000	0.000000
X9 (6)	6.000000	0.000000
X9 (7)	6.000000	0.000000
X9 (8)	15.00000	0.000000
X9 (9)	7.000000	0.000000
X9 (10)	7.000000	0.000000
X9 (11)	7.000000	0.000000
X9 (12)	7.000000	0.000000
X9 (13)	7.000000	0.000000
X9 (14)	15.00000	0.000000
X9 (15)	6.000000	0.000000
X9 (16)	6.000000	0.000000
X9 (17)	6.000000	0.000000
X9 (18)	6.000000	0.000000
X9 (19)	6.000000	0.000000
X9 (20)	6.000000	0.000000
X9 (21)	6.000000	0.000000
X9 (22)	15.00000	-6.000000
Y3 (1)	0.000000	0.000000
Y3 (2)	0.000000	0.000000
Y3 (3)	1.000000	0.000000
Y3 (4)	0.000000	0.000000
Y3 (5)	0.000000	0.000000
Y3 (6)	0.000000	0.000000
Y3 (7)	0.000000	0.000000
Y3 (8)	0.000000	0.000000
Y3 (9)	0.000000	0.000000
Y3 (10)	0.000000	0.000000
Y3 (11)	0.000000	0.000000
Y3 (12)	1.000000	0.000000
Y3 (13)	0.000000	0.000000
Y3 (14)	0.000000	0.000000
Y3 (15)	0.000000	0.000000
Y3 (16)	0.000000	0.000000
Y3 (17)	1.000000	0.000000
Y3 (18)	0.000000	0.000000

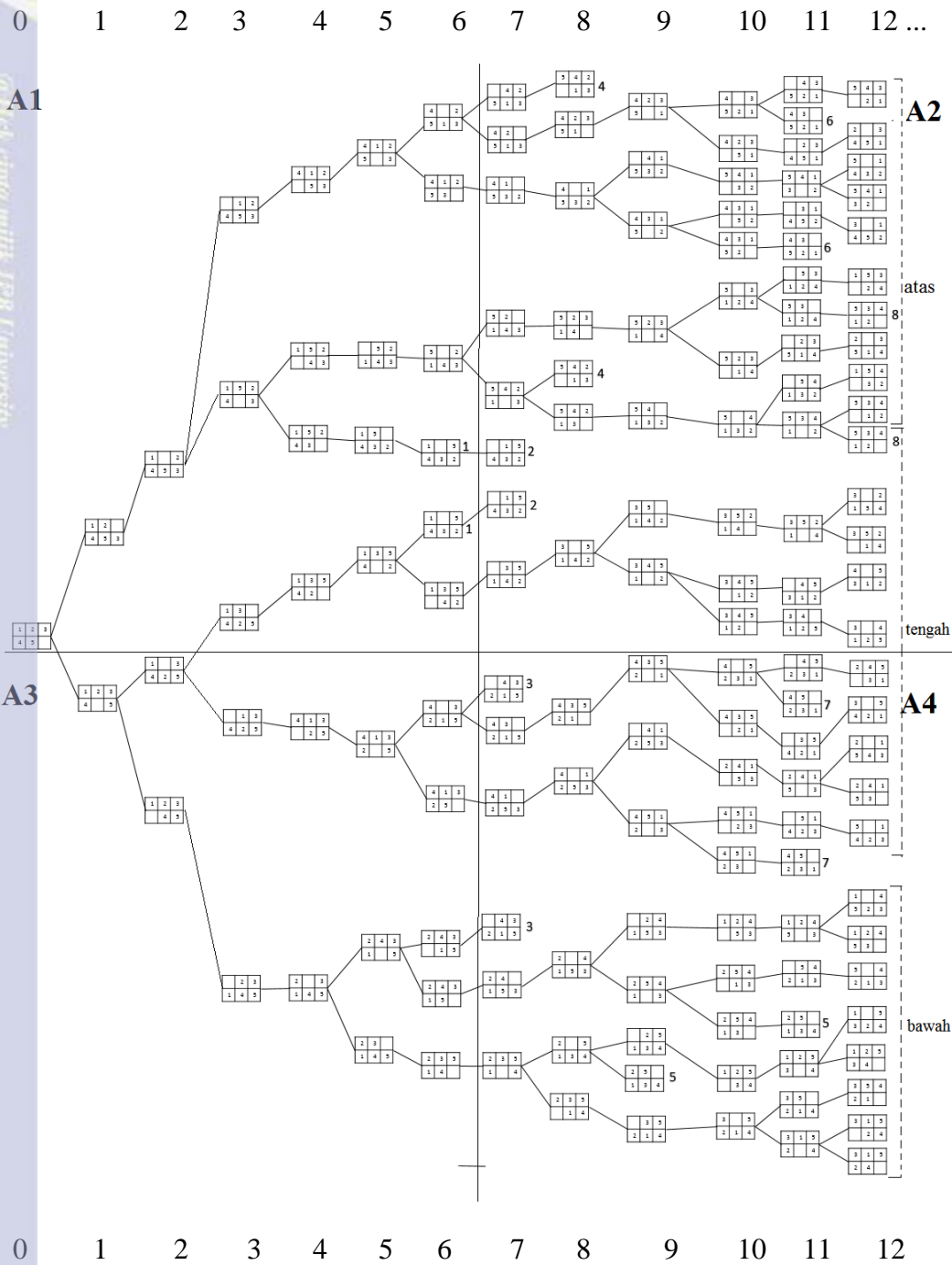
Y11 (7)	0.000000	0.000000
Y11 (8)	0.000000	0.000000
Y11 (9)	0.000000	0.000000
Y11 (10)	0.000000	0.000000
Y11 (11)	0.000000	0.000000
Y11 (12)	0.000000	0.000000
Y11 (13)	1.000000	0.000000
Y11 (14)	0.000000	0.000000
Y11 (15)	0.000000	0.000000
Y11 (16)	0.000000	0.000000
Y11 (17)	0.000000	0.000000
Y11 (18)	0.000000	0.000000
Y11 (19)	0.000000	0.000000
Y11 (20)	1.000000	0.000000
Y11 (21)	0.000000	0.000000
Y11 (22)	0.000000	0.000000
Y12 (1)	0.000000	0.000000
Y12 (2)	0.000000	0.000000
Y12 (3)	0.000000	0.000000
Y12 (4)	0.000000	0.000000
Y12 (5)	0.000000	0.000000
Y12 (6)	0.000000	0.000000
Y12 (7)	0.000000	0.000000
Y12 (8)	1.000000	0.000000
Y12 (9)	0.000000	0.000000
Y12 (10)	0.000000	0.000000
Y12 (11)	0.000000	0.000000
Y12 (12)	0.000000	0.000000
Y12 (13)	0.000000	0.000000
Y12 (14)	0.000000	0.000000
Y12 (15)	1.000000	0.000000
Y12 (16)	0.000000	0.000000
Y12 (17)	0.000000	0.000000
Y12 (18)	0.000000	0.000000
Y12 (19)	0.000000	0.000000
Y12 (20)	0.000000	0.000000
Y12 (21)	0.000000	0.000000
Y12 (22)	1.000000	0.000000



Lampiran 5 Ilustrasi perubahan *puzzle* berdasarkan solusi Model 2



Lampiran 6 Susunan tersulit puzzle 2 x 3 dapat dicapai dengan 21 langkah dari susunan solusi



0 1 2 3 4 5 6 7 8 9 10 11 12

A1

A2

A3

A4

atas

tengah

bawah

Click cipta milik IPB University

IPB University



Halaman ini adalah hak cipta milik IPB University dan tidak boleh disebarluaskan atau diperjualbelikan kembali.

1. Dilindungi sebagai bagian dari koleksi digital IPB University dan tidak boleh disebarluaskan atau diperjualbelikan kembali.

2. Diperbolehkan untuk mengutip dan menyalin sebagian atau seluruhnya dengan syarat harus mencantumkan sumber.

3. Diperbolehkan untuk mengutip dan menyalin sebagian atau seluruhnya dengan syarat harus mencantumkan sumber.

4. Diperbolehkan untuk mengutip dan menyalin sebagian atau seluruhnya dengan syarat harus mencantumkan sumber.

5. Diperbolehkan untuk mengutip dan menyalin sebagian atau seluruhnya dengan syarat harus mencantumkan sumber.

6. Diperbolehkan untuk mengutip dan menyalin sebagian atau seluruhnya dengan syarat harus mencantumkan sumber.

7. Diperbolehkan untuk mengutip dan menyalin sebagian atau seluruhnya dengan syarat harus mencantumkan sumber.

8. Diperbolehkan untuk mengutip dan menyalin sebagian atau seluruhnya dengan syarat harus mencantumkan sumber.

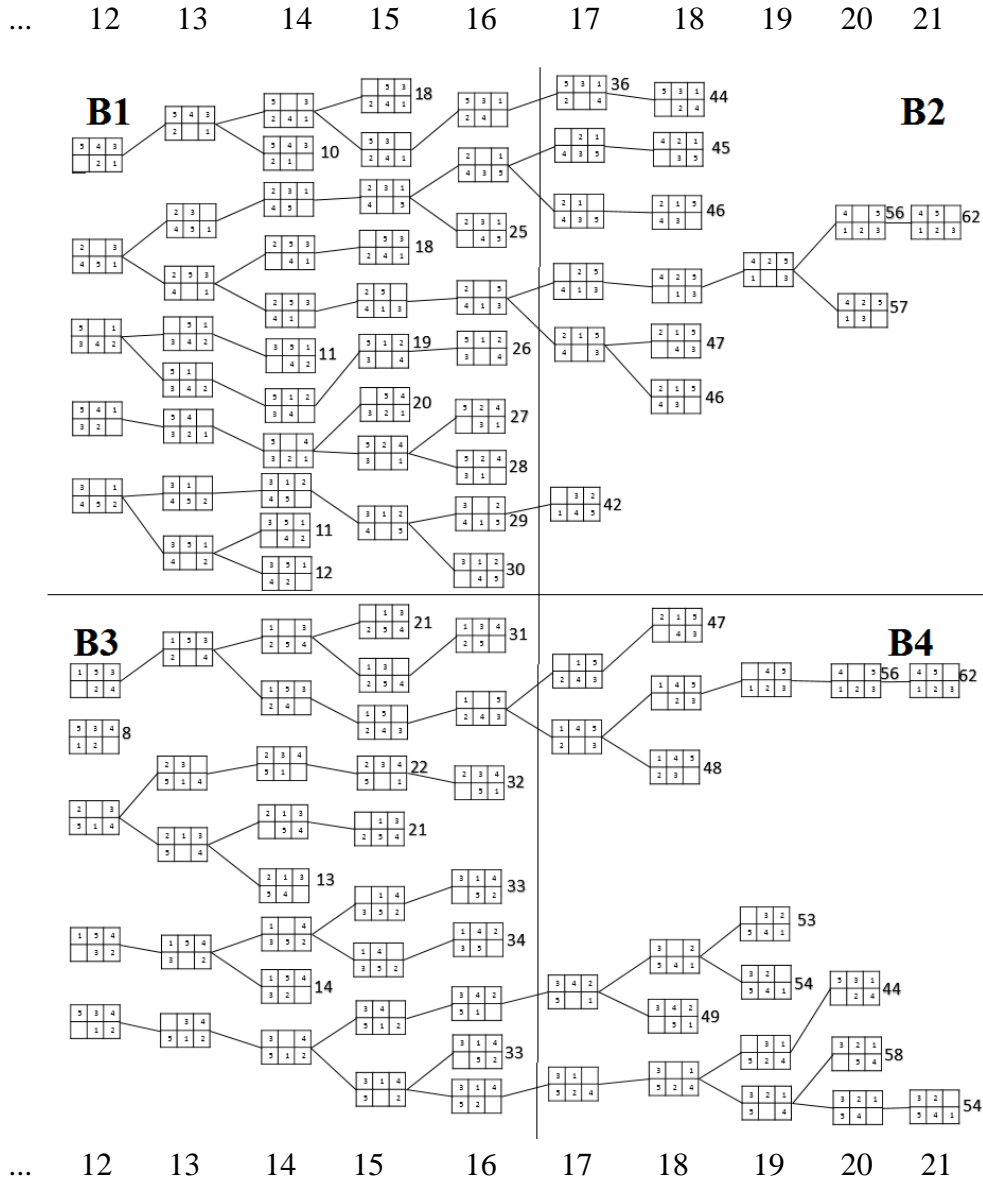
9. Diperbolehkan untuk mengutip dan menyalin sebagian atau seluruhnya dengan syarat harus mencantumkan sumber.

10. Diperbolehkan untuk mengutip dan menyalin sebagian atau seluruhnya dengan syarat harus mencantumkan sumber.

11. Diperbolehkan untuk mengutip dan menyalin sebagian atau seluruhnya dengan syarat harus mencantumkan sumber.

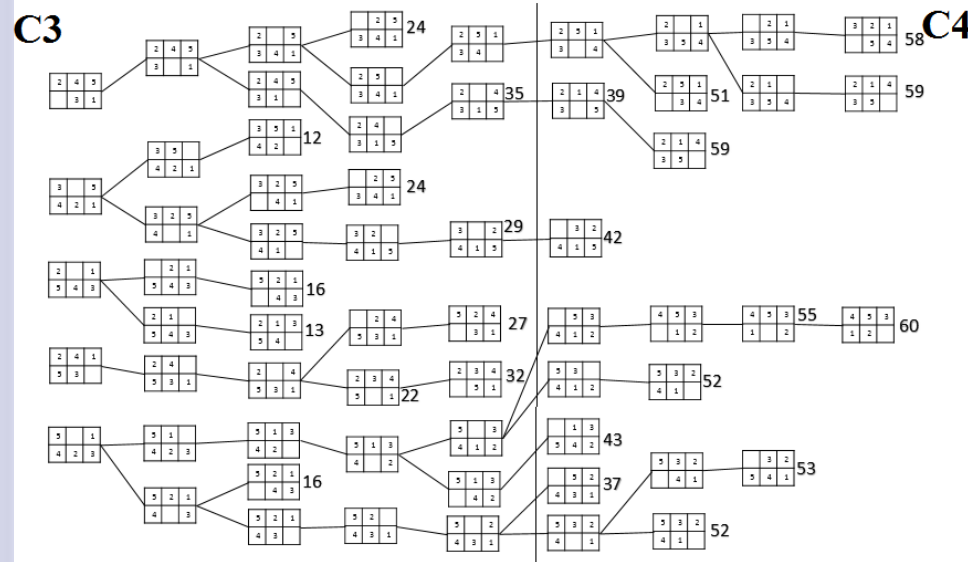
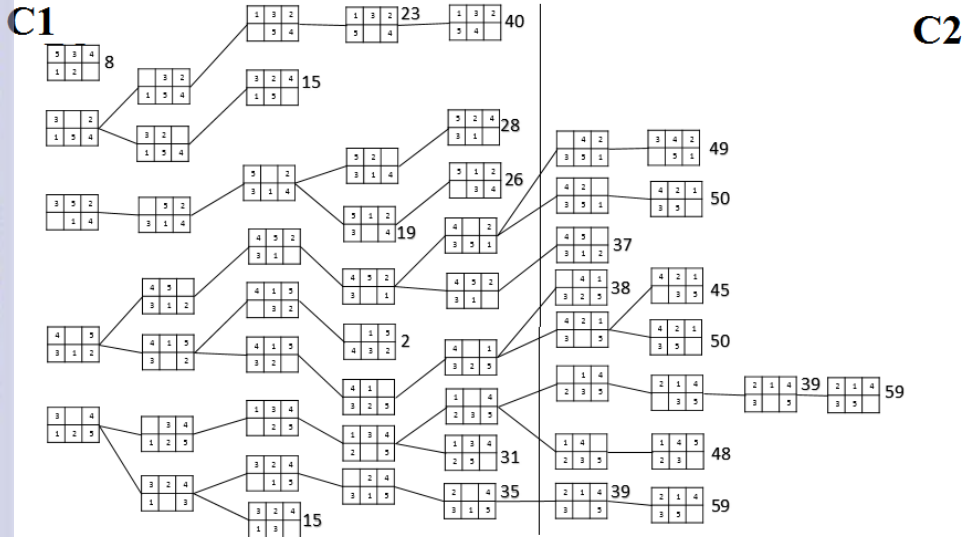
12. Diperbolehkan untuk mengutip dan menyalin sebagian atau seluruhnya dengan syarat harus mencantumkan sumber.

Lanjutan bagian atas halaman 56



Lanjutan bagian tengah halaman 56

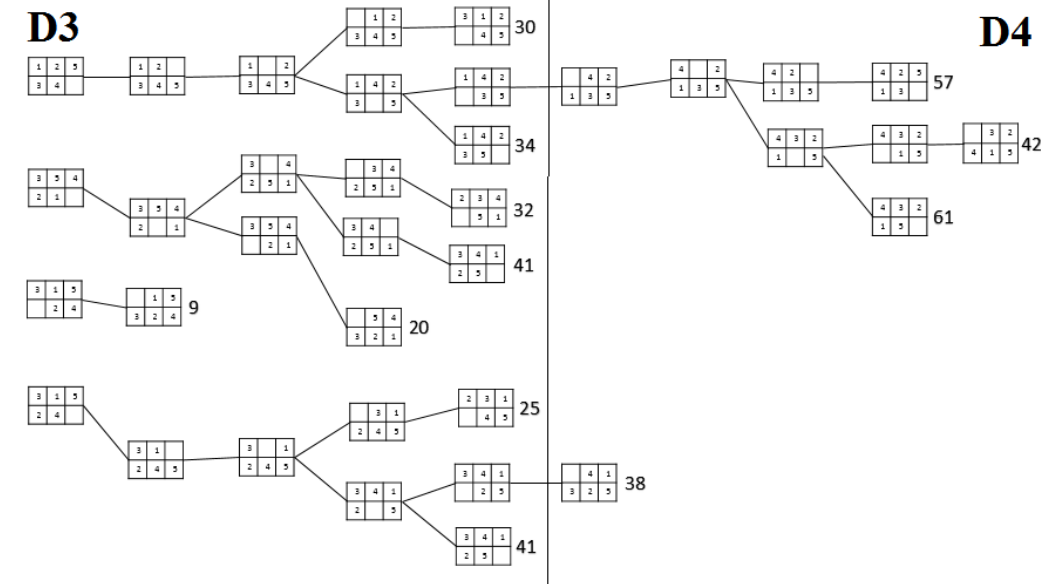
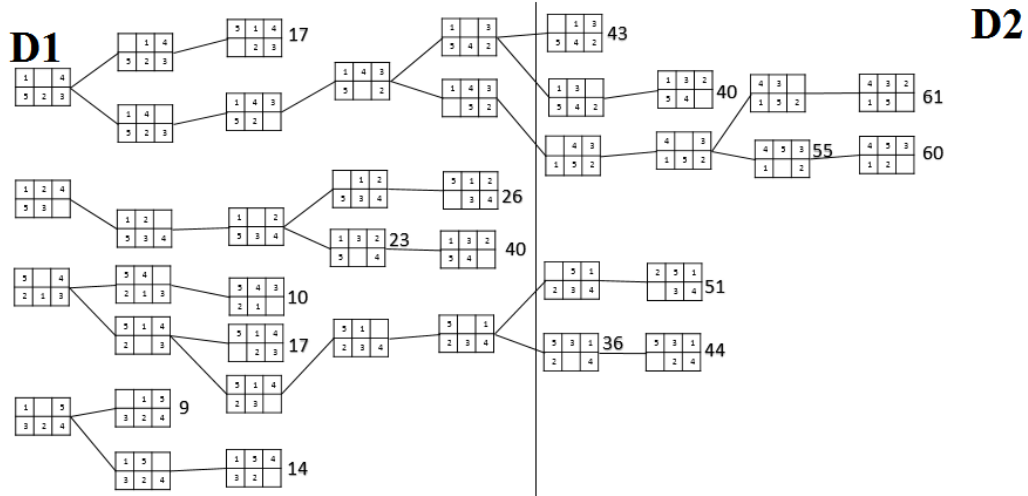
... 12 13 14 15 16 17 18 19 20 21



... 12 13 14 15 16 17 18 19 20 21

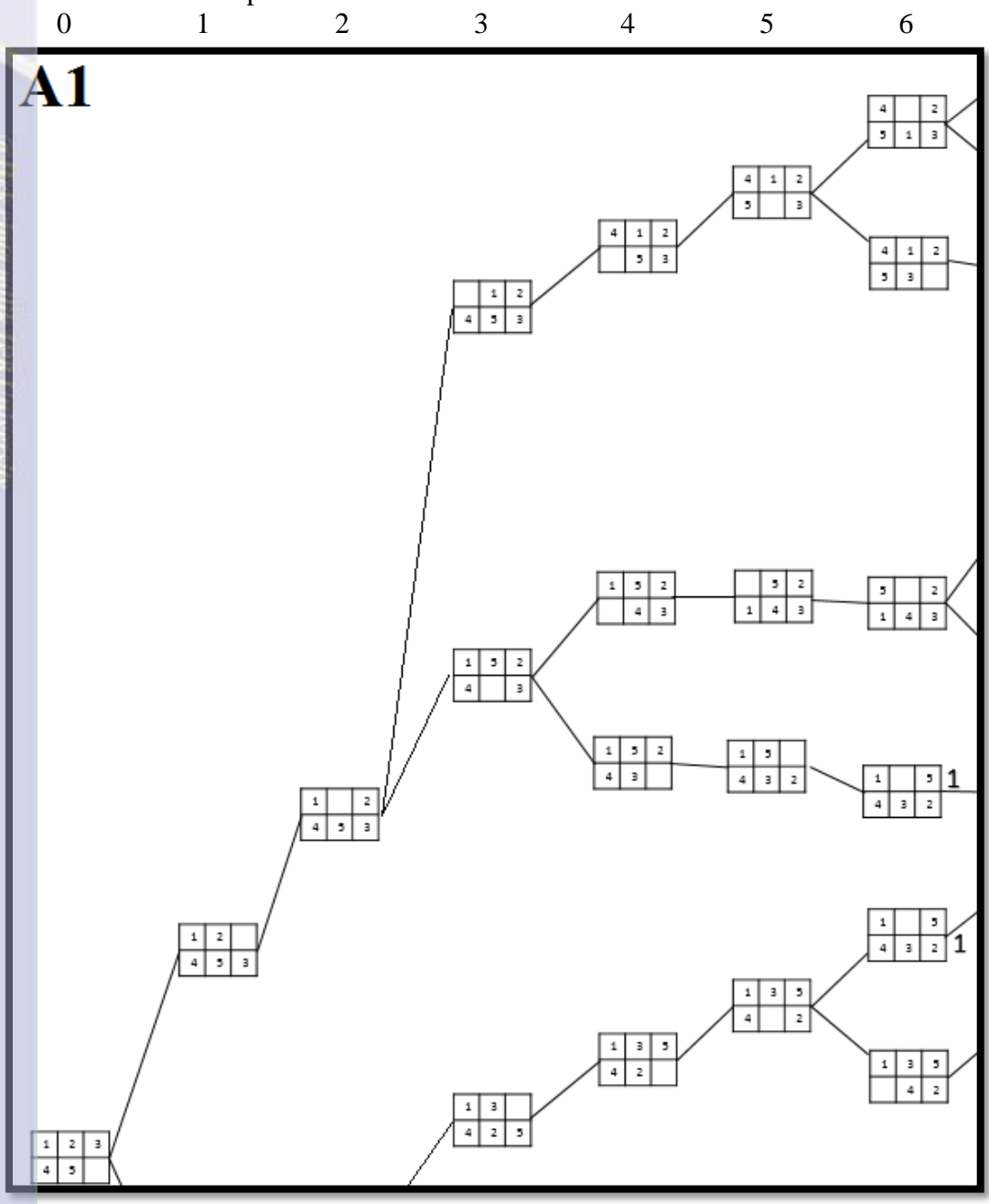
Lanjutan bagian bawah halaman 56

... 12 13 14 15 16 17 18 19 20 21



...12 13 14 15 16 17 18 19 20 21

Perbesaran dari A1 pada halaman 56



Halaman 60 (Penerjemah: Unsur-unsur)

1. Diambil sebagai bagian dari jurnal karya tulis yang diterbitkan dan dipublikasikan.

2. Pengantar jurnal untuk kepentingan pendidikan, penelitian, pengabdian masyarakat, dan tujuan lain yang mulia.

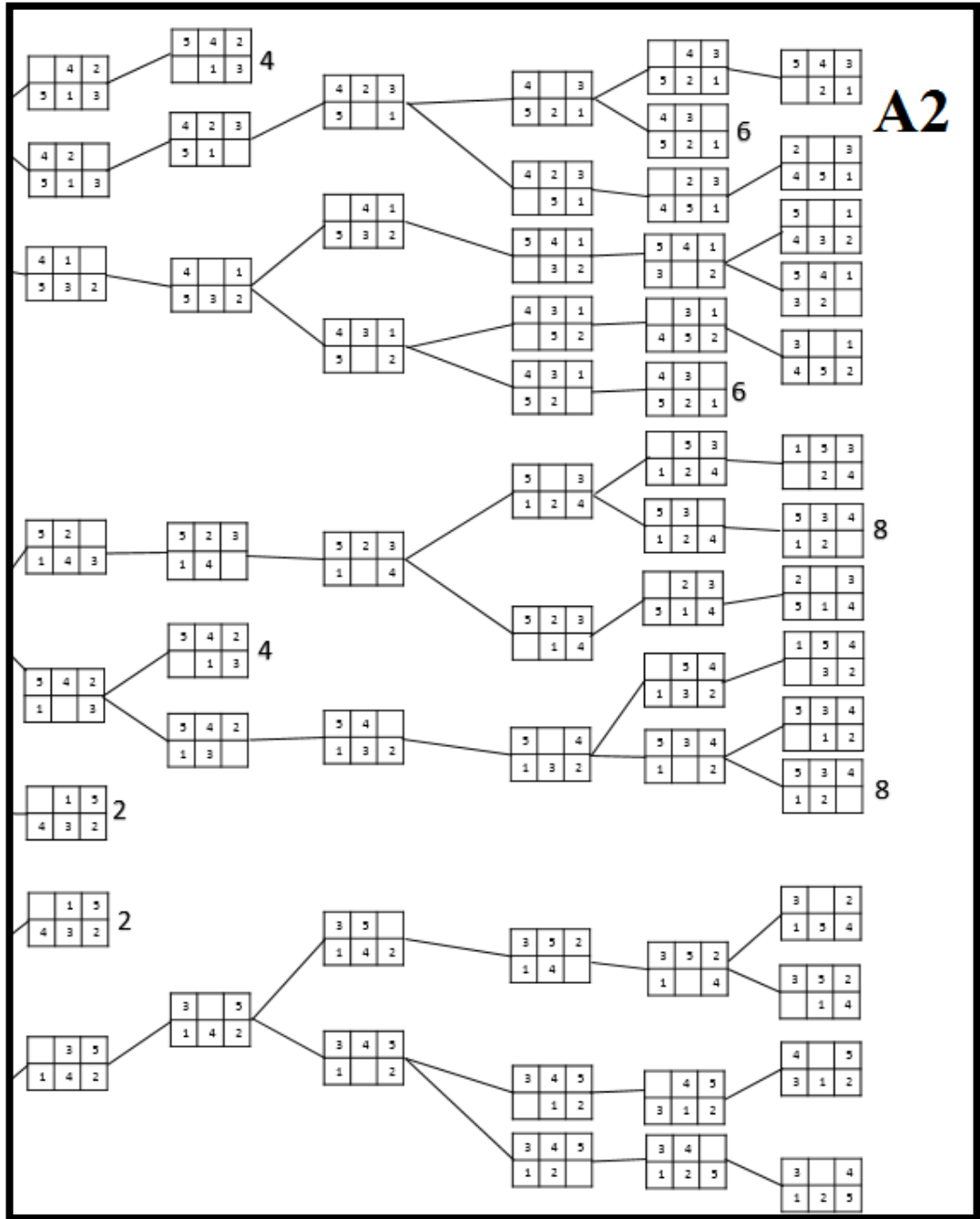
3. Pengantar jurnal tidak merugikan kepentingan yang wajar IPB University.

4. Diambil, mengutip, dan menerjemahkan sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin IPB University.

IPB University

Perbesaran dari A2 pada halaman 56

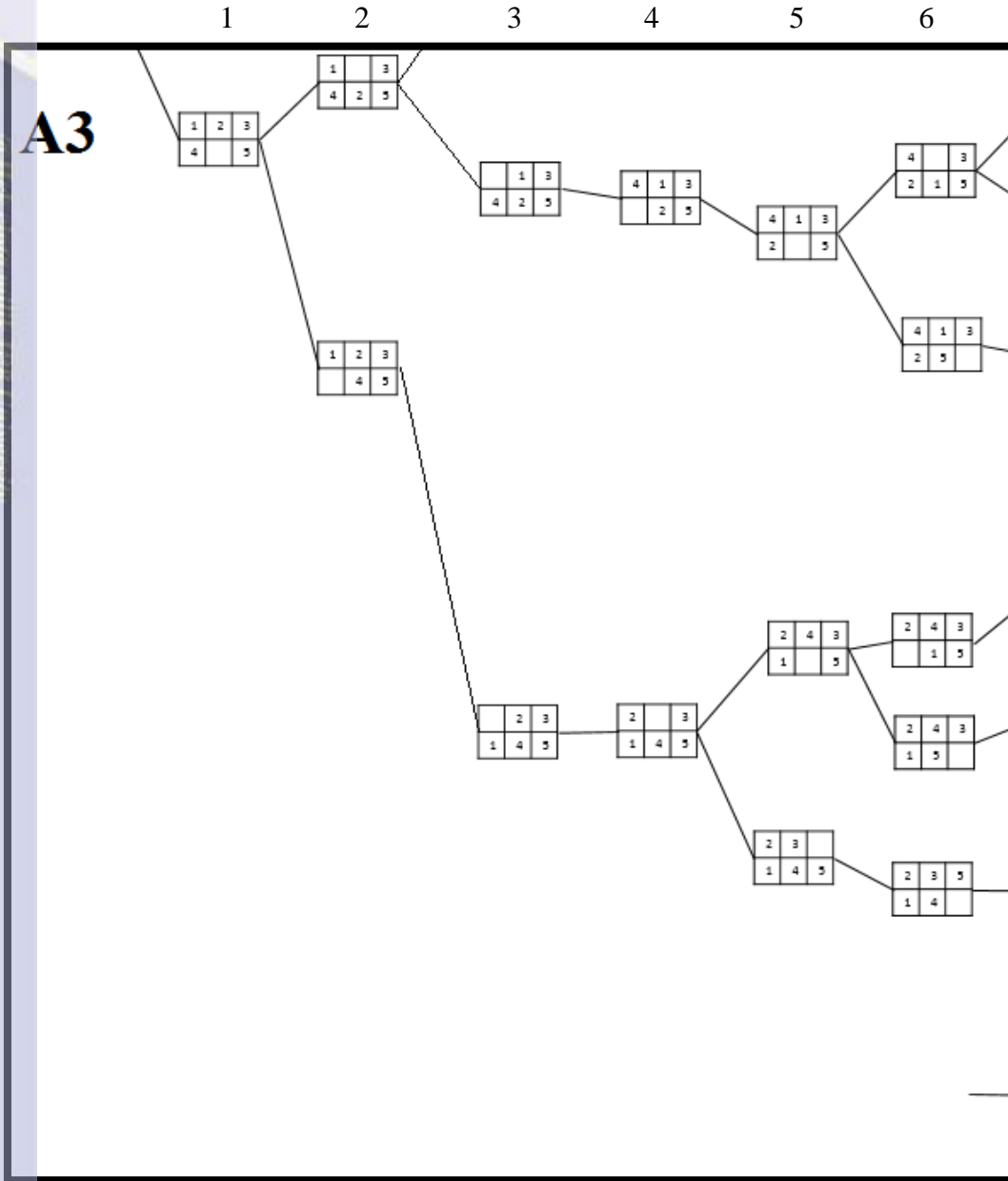
7 8 9 10 11 12



A2

Halaman 61 dari 61 halaman

Perbesaran dari A3 pada halaman 56

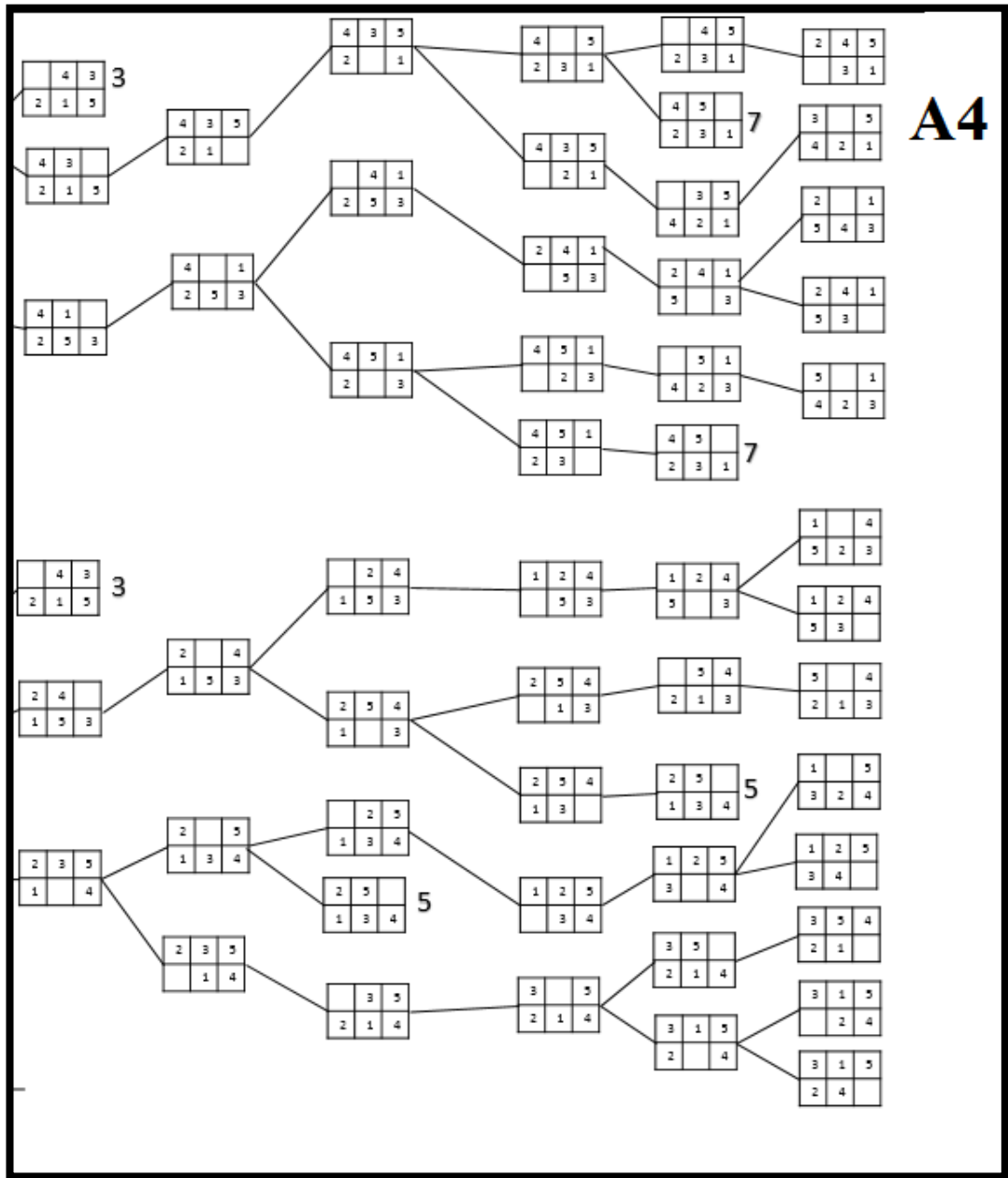


Gedung Sate IPB University

Halaman ini adalah...
1. Diambil sebagai...
2. Diambil sebagai...
3. Diambil sebagai...
4. Diambil sebagai...
5. Diambil sebagai...
6. Diambil sebagai...

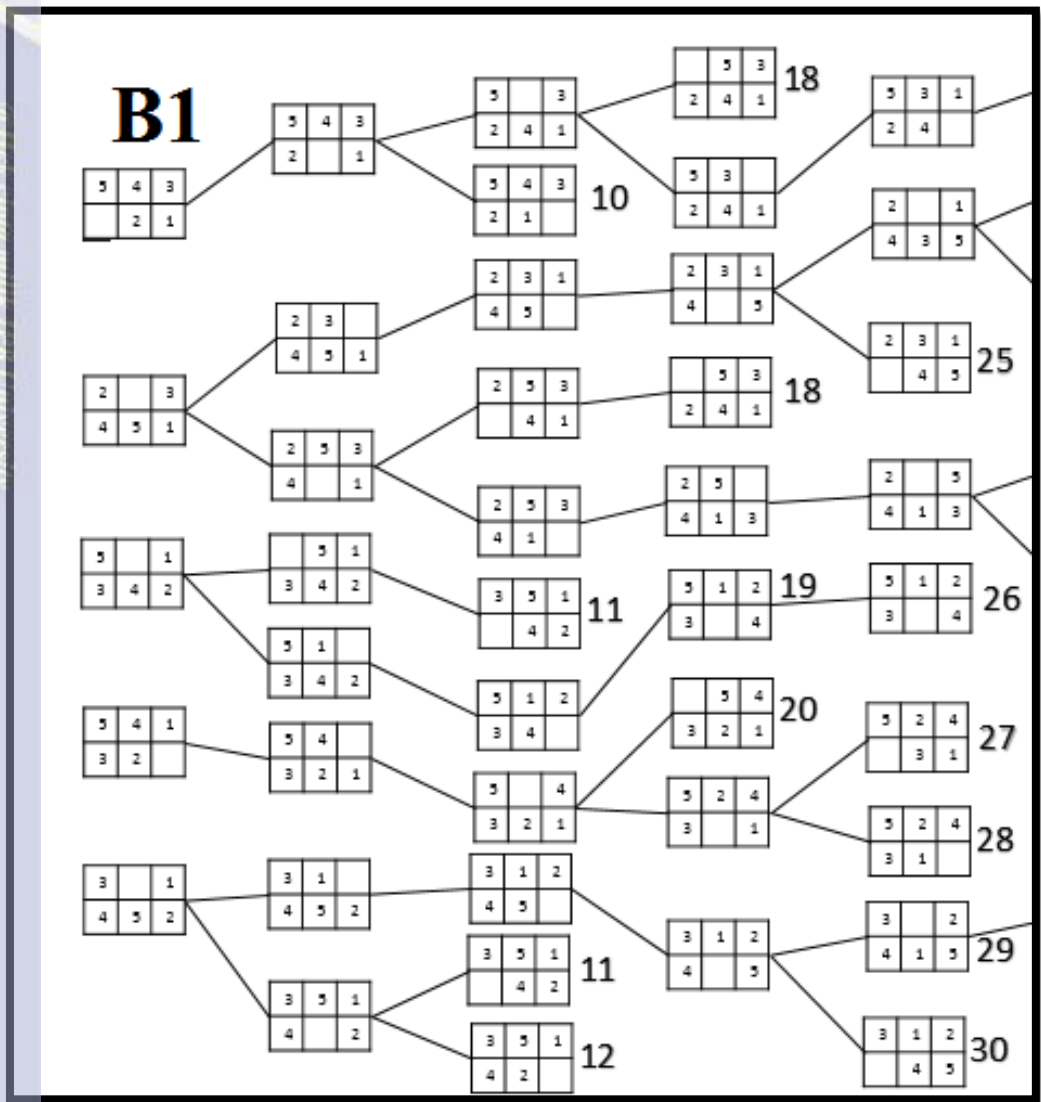
Perbesaran dari A4 pada halaman 56

7 8 9 10 11 12



Perbesaran dari B1 pada halaman 57

12 13 14 15 16



Halaman Perbesaran (Perbesar) dari B1 pada halaman 57

1. Diambil dari beberapa sumber yang ada di internet.

2. Perbesar dari B1 pada halaman 57.

3. Perbesar dari B1 pada halaman 57.

4. Perbesar dari B1 pada halaman 57.

5. Perbesar dari B1 pada halaman 57.

6. Perbesar dari B1 pada halaman 57.

7. Perbesar dari B1 pada halaman 57.

8. Perbesar dari B1 pada halaman 57.

9. Perbesar dari B1 pada halaman 57.

10. Perbesar dari B1 pada halaman 57.

11. Perbesar dari B1 pada halaman 57.

12. Perbesar dari B1 pada halaman 57.

Perbesaran dari B2 pada halaman 57

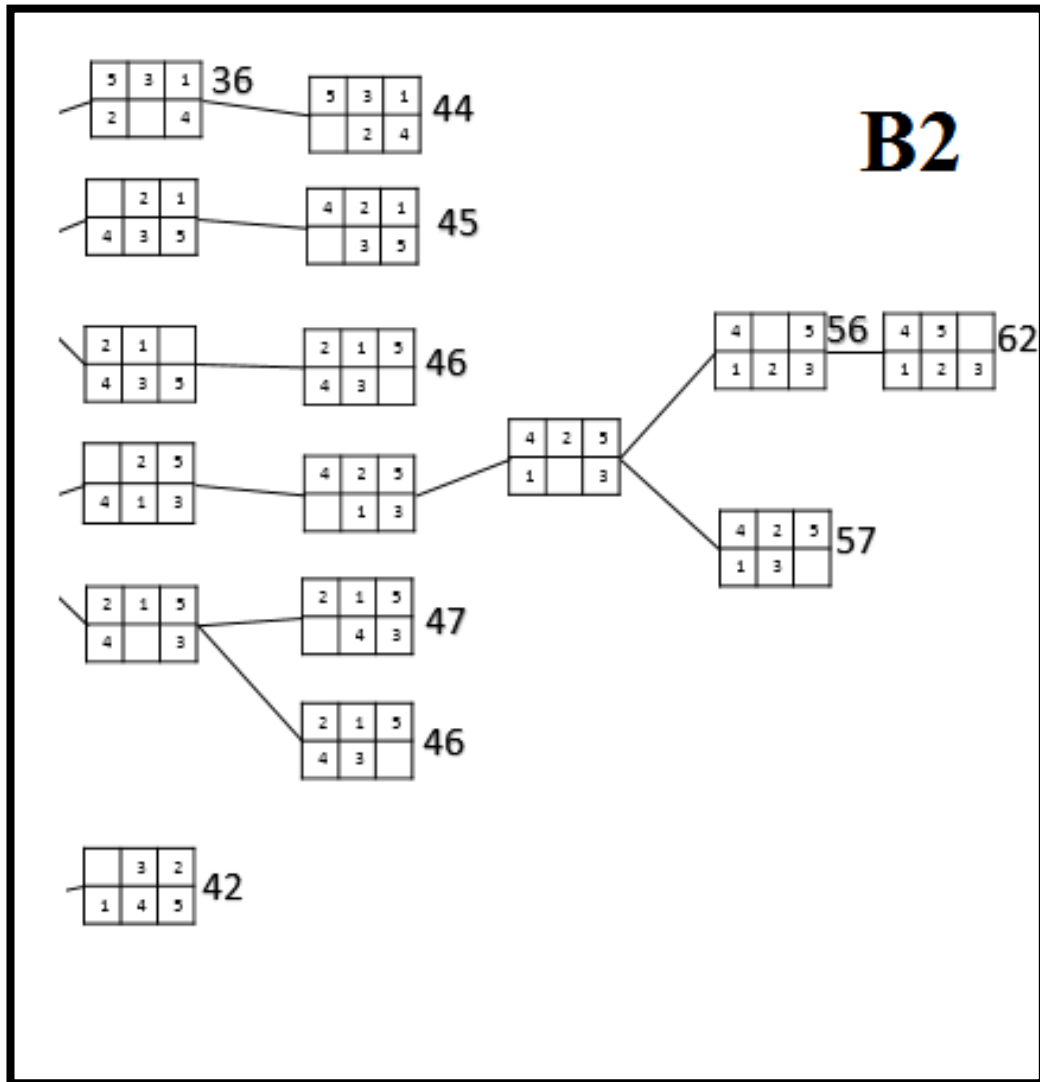
17

18

19

20

21



Perbesaran dari B3 pada halaman 57

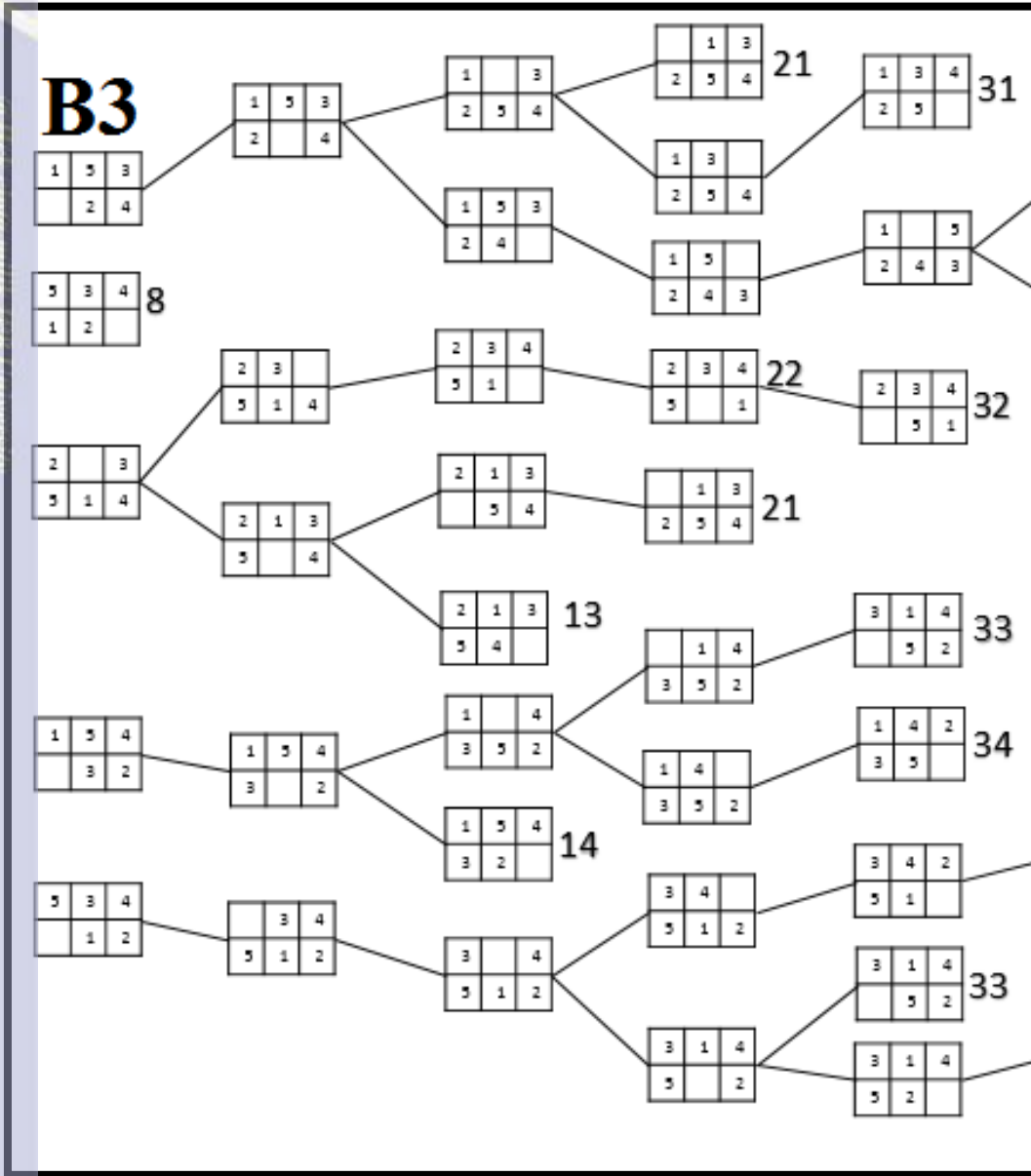
12

13

14

15

16



Gates, Sipin, and ...

1. Diambil sebagai ...
 2. Diambil sebagai ...
 3. Diambil sebagai ...
 4. Diambil sebagai ...
 5. Diambil sebagai ...
 6. Diambil sebagai ...
 7. Diambil sebagai ...
 8. Diambil sebagai ...
 9. Diambil sebagai ...
 10. Diambil sebagai ...

Perbesaran dari B4 pada halaman 57

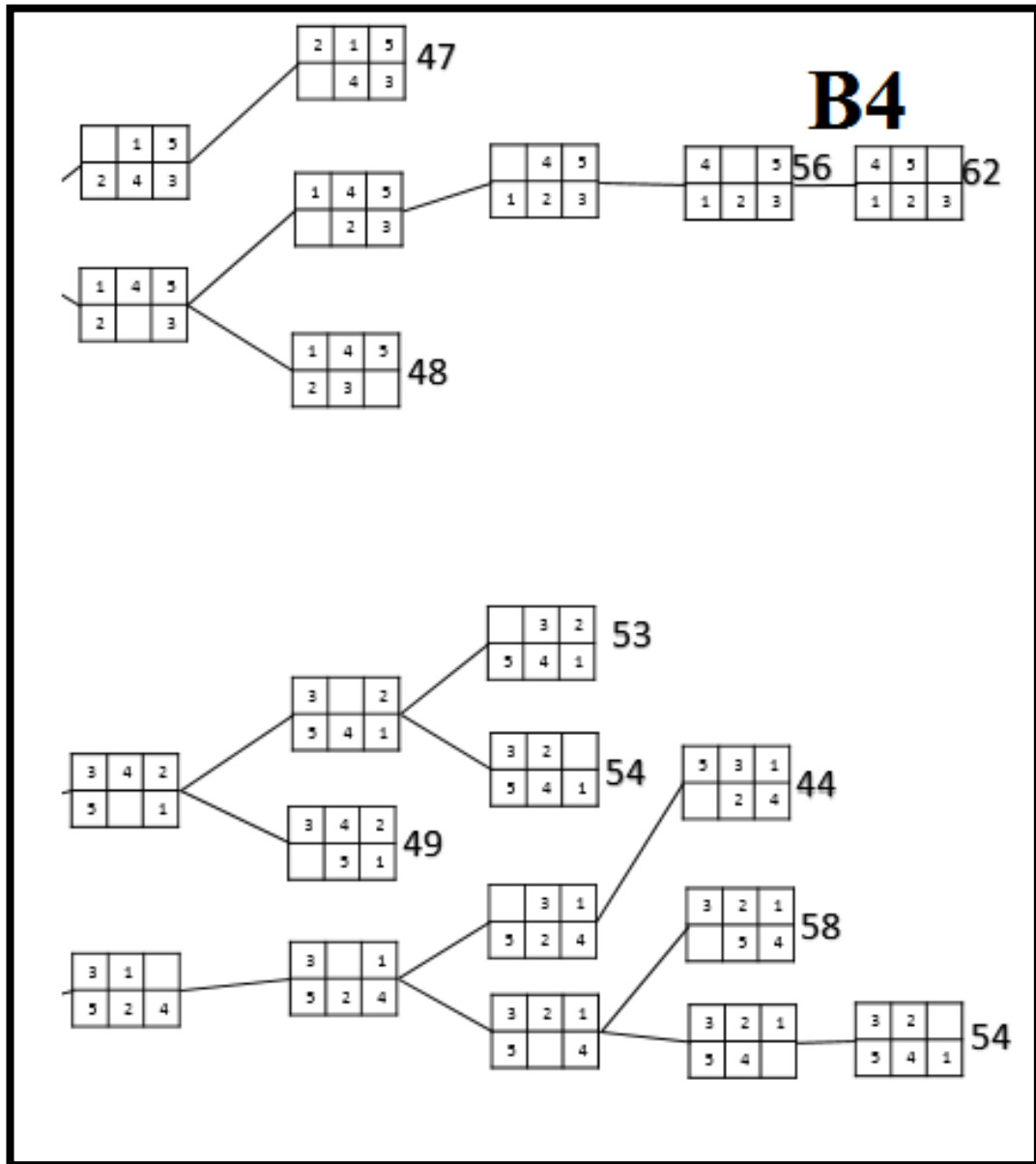
17

18

19

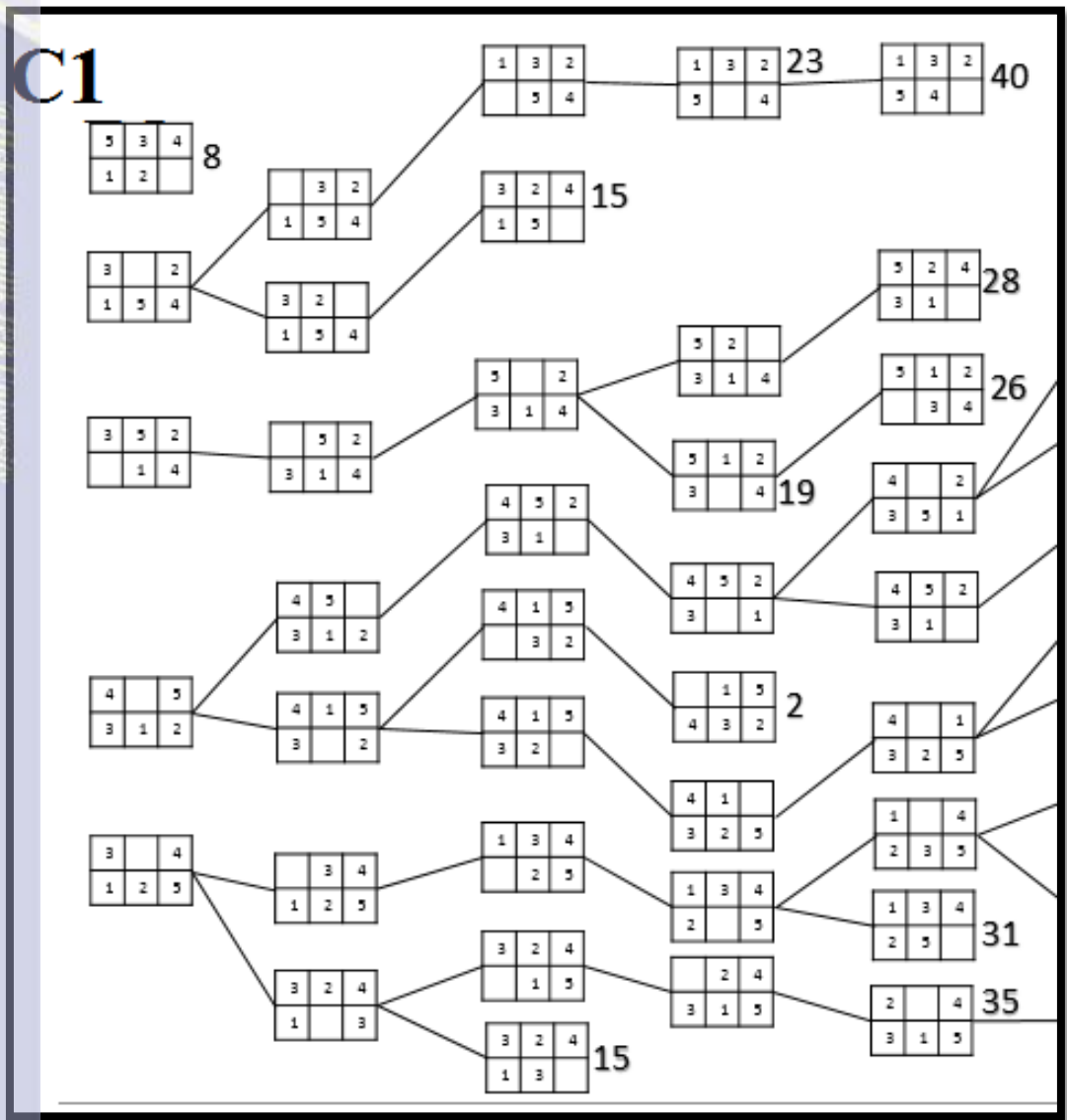
20

21



Perbesaran dari C1 pada halaman 58

12 13 14 15 16



Halaman ini diterbitkan oleh IPB University sebagai bagian dari upaya meningkatkan dan memperluas aksesibilitas informasi. Penggunaan informasi yang terdapat di sini adalah untuk tujuan akademik dan penelitian. Penggunaan informasi yang terdapat di sini untuk tujuan komersial atau untuk melanggar hukum adalah dilarang. IPB University tidak bertanggung jawab atas kerugian yang timbul akibat penggunaan informasi yang terdapat di sini. IPB University.

Perbesaran dari C2 pada halaman 58

17

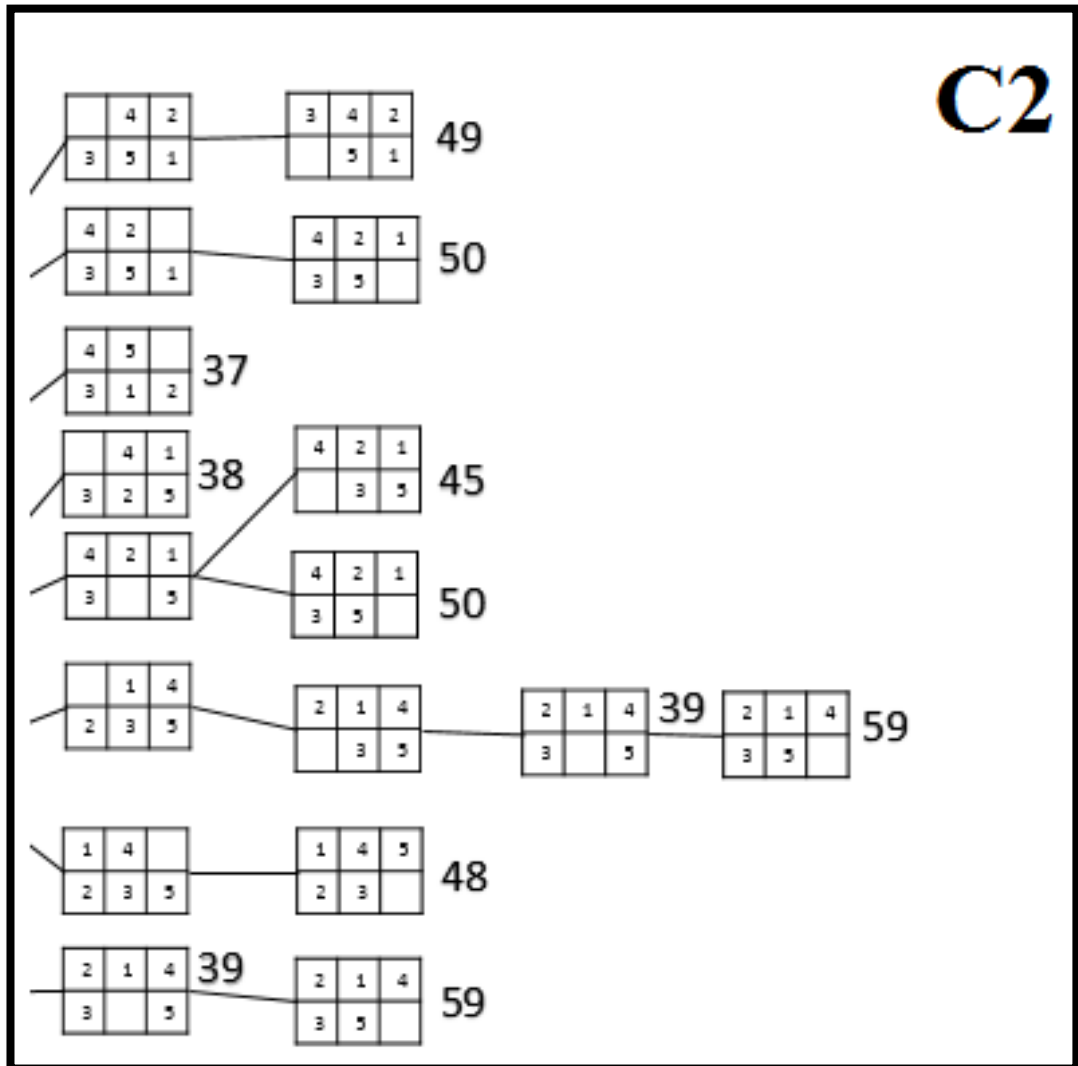
18

19

20

21

C2



Perbesaran dari C3 pada halaman 58

12

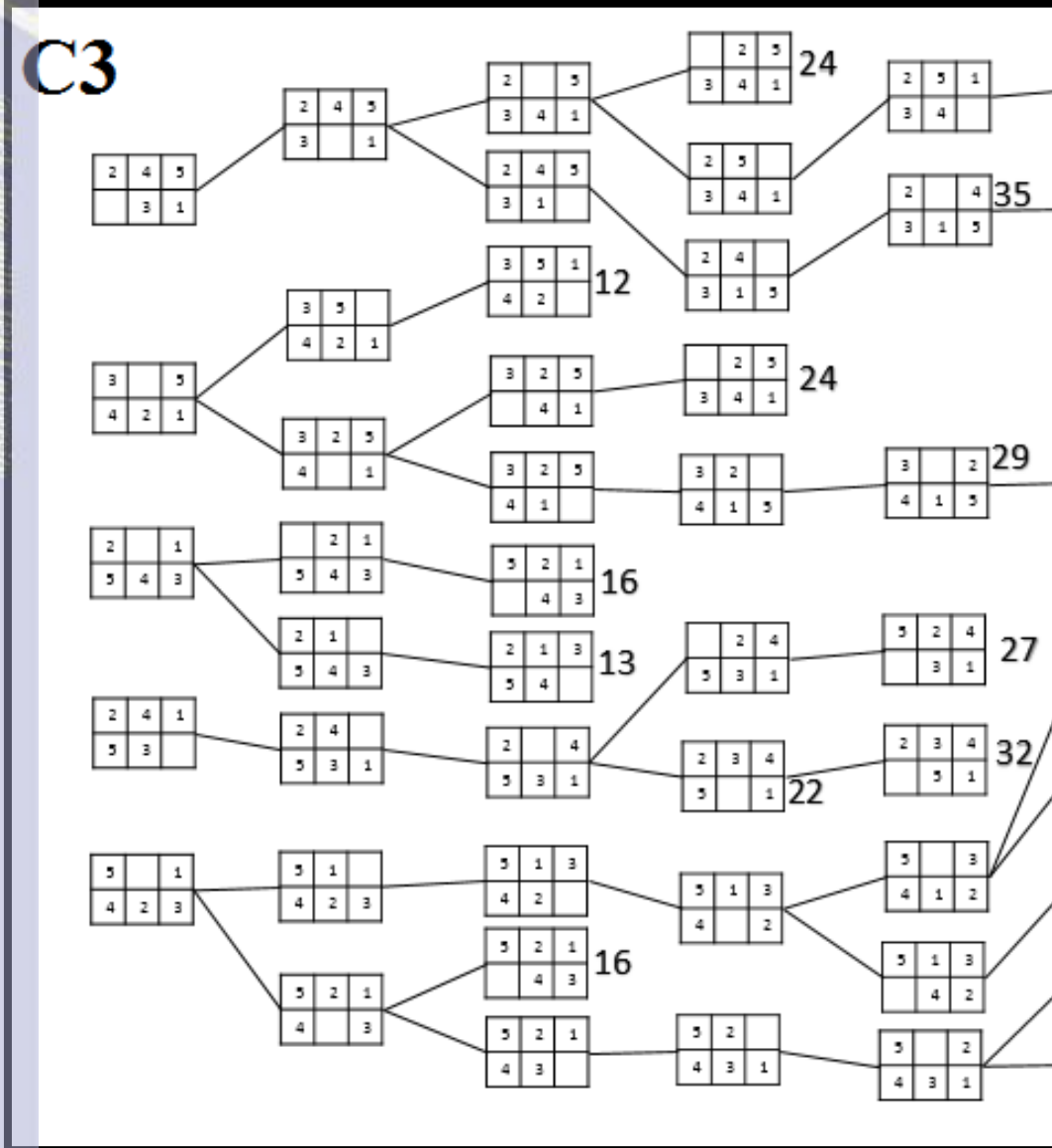
13

14

15

16

C3



© 2013 Institut Pertanian Bogor

Halaman ini adalah...
1. Diambil...
2. Diambil...

Perbesaran dari C4 pada halaman 58

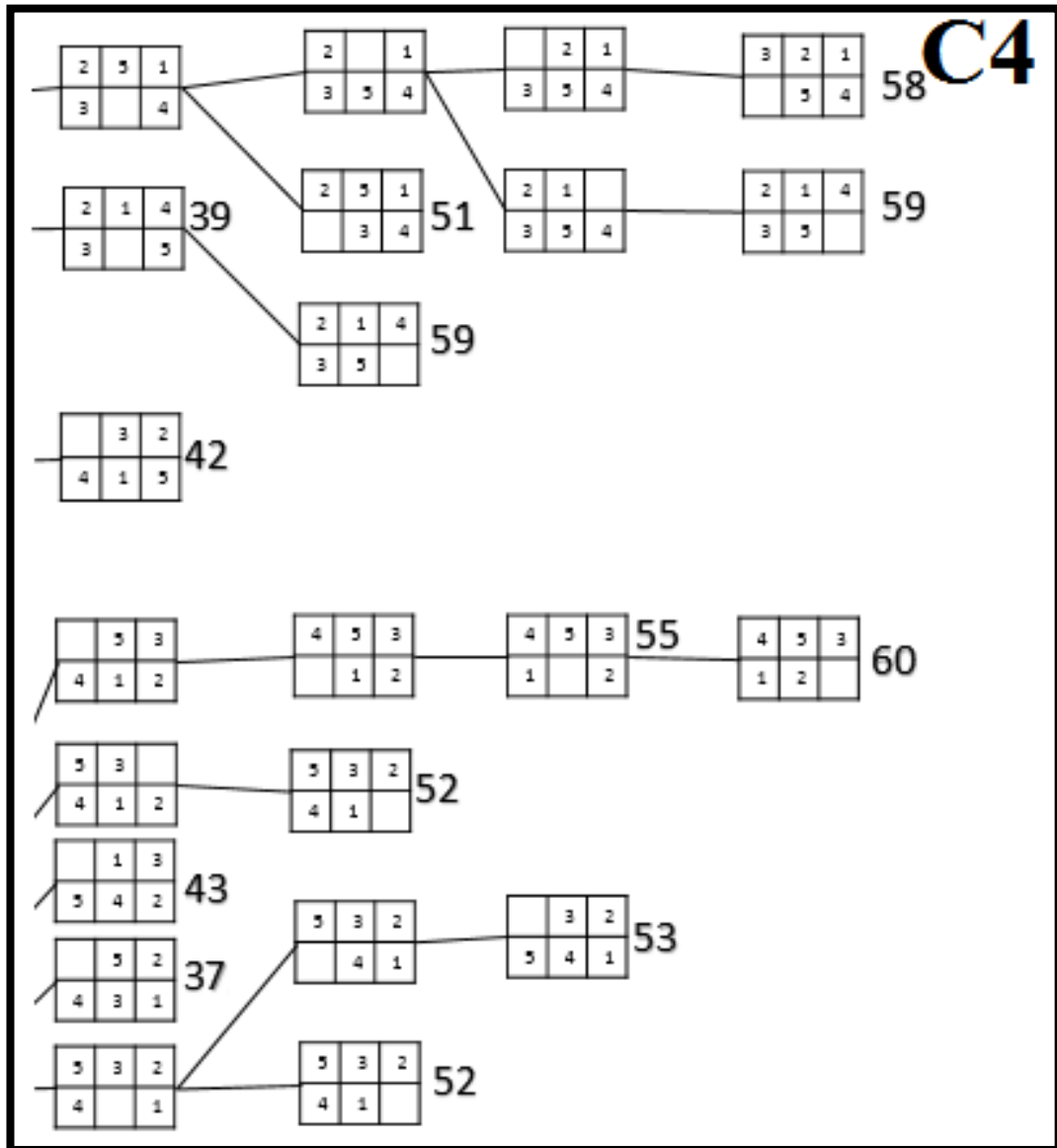
17

18

19

20

21



Perbesaran dari D1 pada halaman 59

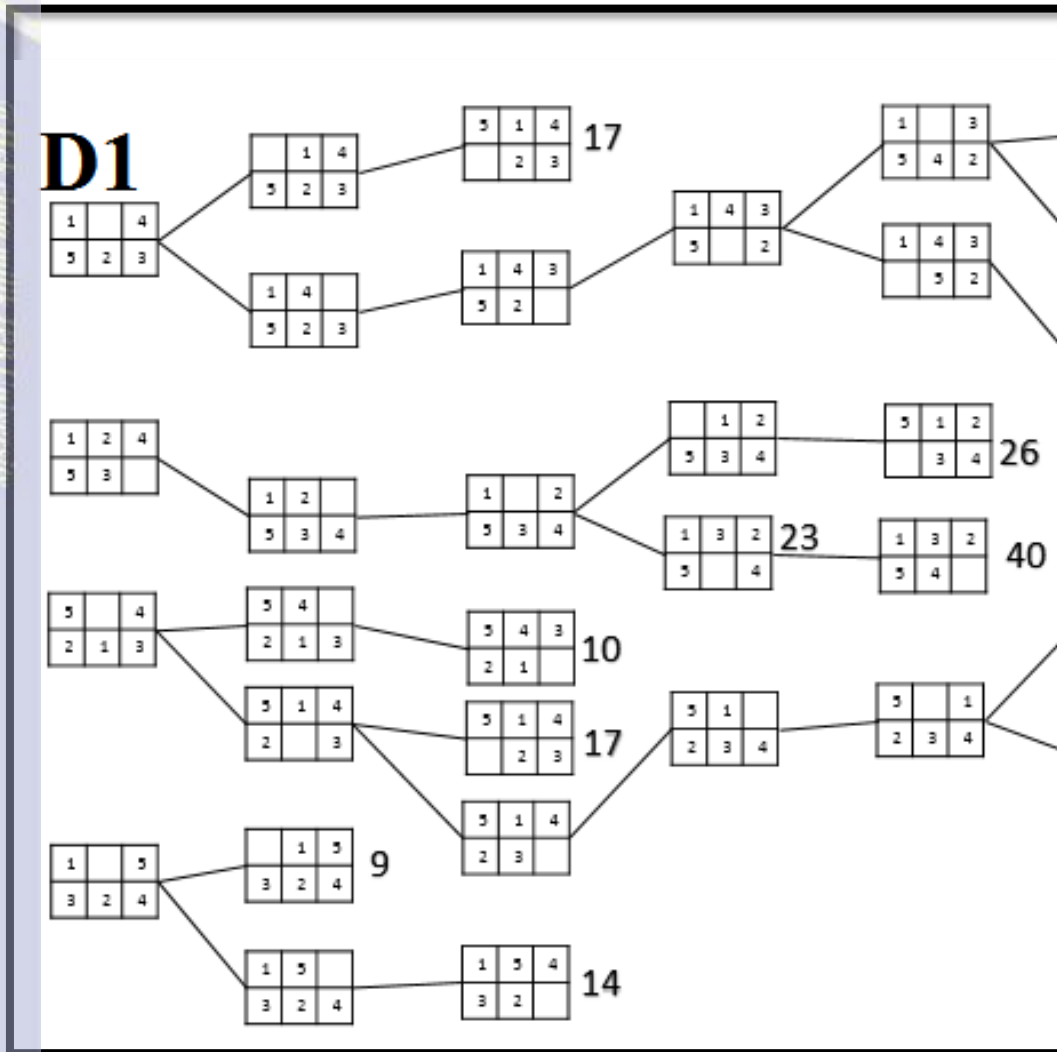
12

13

14

15

16



Hubungan antara D1 dan D2

Halaman ini merupakan bagian dari buku yang berjudul "Hubungan antara D1 dan D2" yang diterbitkan oleh IPB University. Buku ini membahas tentang hubungan antara D1 dan D2 dalam konteks tertentu. Untuk informasi lebih lanjut, silakan kunjungi website IPB University.

Perbesaran dari D2 pada halaman 59

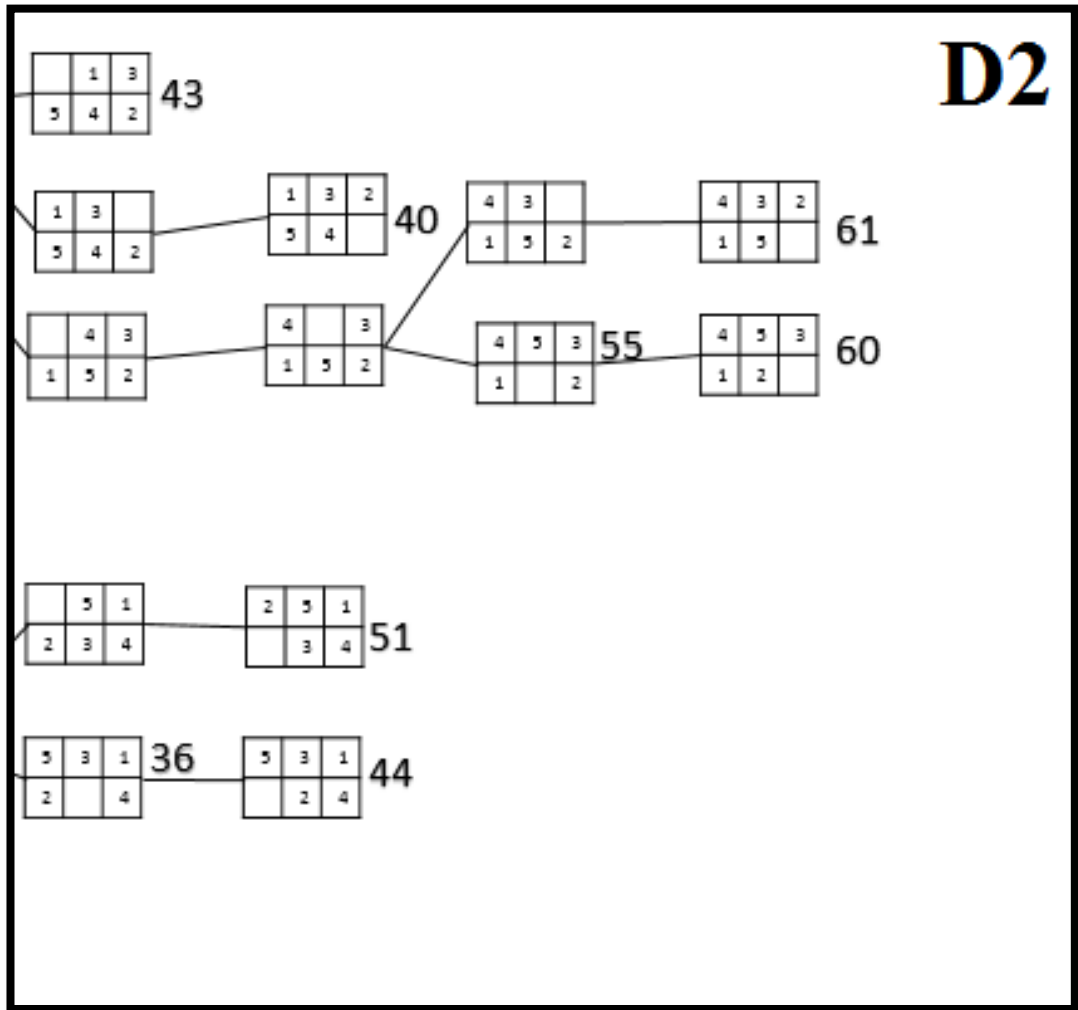
17

18

19

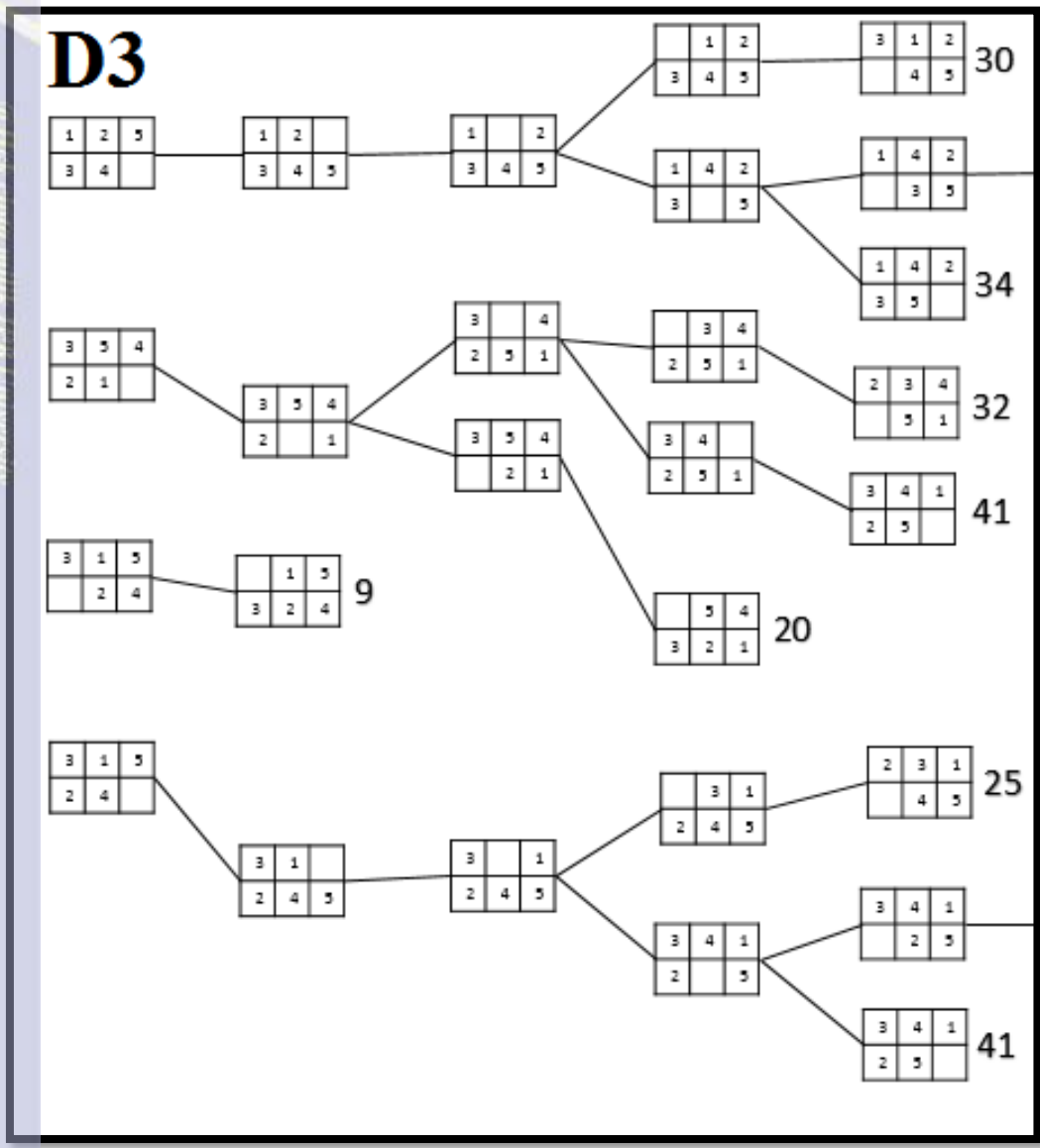
20

21



Perbesaran dari D3 pada halaman 59

12 13 14 15 16



Halaman Perbesaran (Unsur-unsur)
 1. Diambil sebagai bagian dari sebuah karya tulis yang bertujuan untuk dan memperoleh manfaat
 2. Pengutipan harus mencantumkan sumber dan tidak boleh diubah-ubah
 3. Pengutipan harus mencantumkan sumber dan tidak boleh diubah-ubah
 4. Pengutipan harus mencantumkan sumber dan tidak boleh diubah-ubah
 5. Pengutipan harus mencantumkan sumber dan tidak boleh diubah-ubah
 6. Pengutipan harus mencantumkan sumber dan tidak boleh diubah-ubah
 7. Pengutipan harus mencantumkan sumber dan tidak boleh diubah-ubah
 8. Pengutipan harus mencantumkan sumber dan tidak boleh diubah-ubah
 9. Pengutipan harus mencantumkan sumber dan tidak boleh diubah-ubah
 10. Pengutipan harus mencantumkan sumber dan tidak boleh diubah-ubah

Perbesaran dari D4 pada halaman 59

17

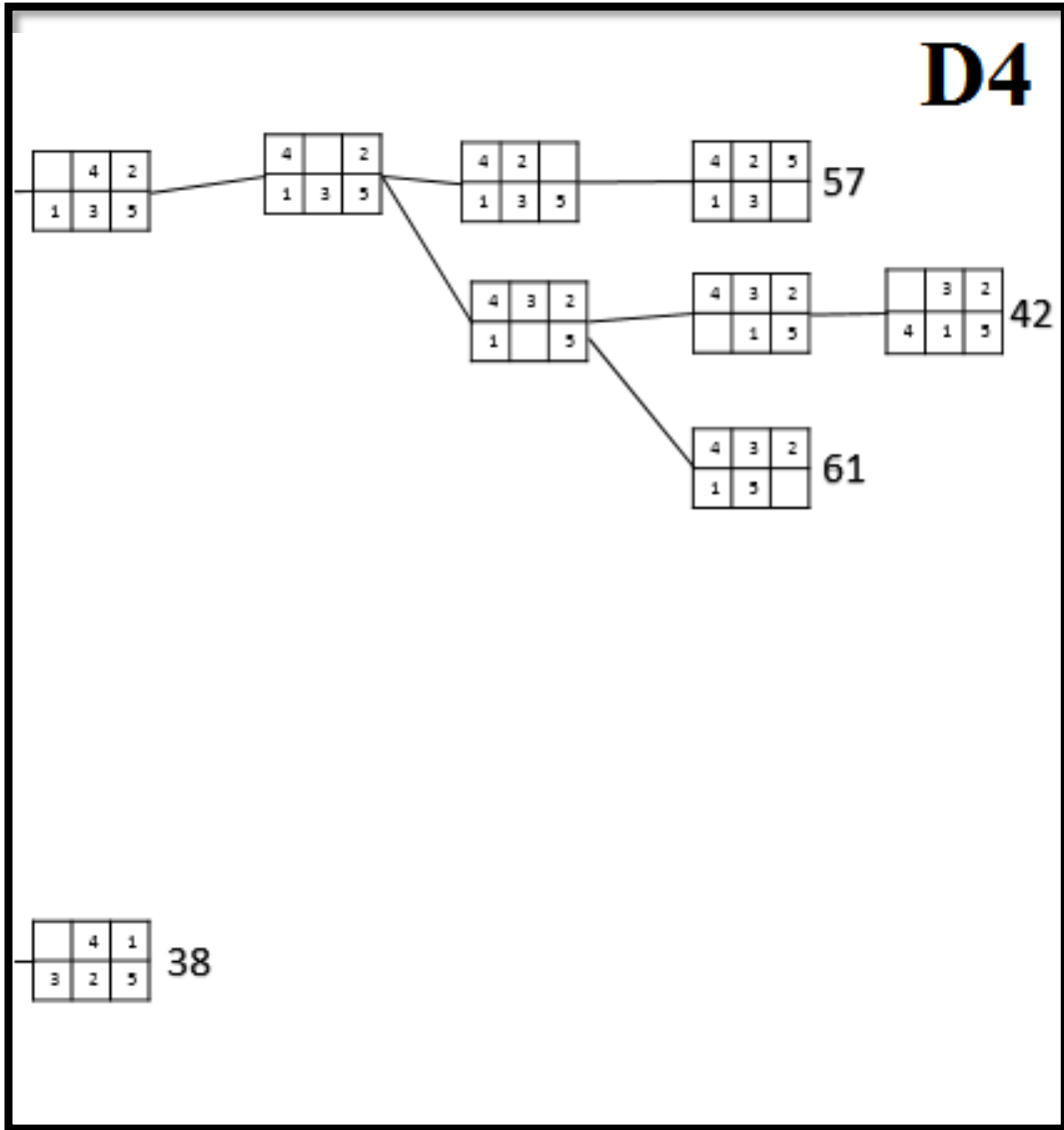
18

19

20

21

D4





RIWAYAT HIDUP

Penulis dilahirkan di Bogor pada tanggal 6 Mei 1991. Penulis adalah putra ke-2 dari empat bersaudara dari ayah Ngadimin dan Ibu Warngiatun. Penulis menyelesaikan pendidikan taman kanak-kanak di TK Al-Muttaqin Kabupaten Bogor pada tahun 1997 dan melanjutkan pendidikan di Sekolah Dasar Negeri Kartika Sejahtera Kabupaten Bogor dan lulus pada tahun 2003. Setelah itu penulis melanjutkan pendidikan di SMP Negeri 14 Depok dan lulus pada tahun 2006 kemudian melanjutkan ke SMA Negeri 5 Bogor dan lulus pada tahun 2009.

Tahun 2009 penulis diterima sebagai mahasiswa di Institut Pertanian Bogor (IPB) melalui jalur PMDK di Departemen Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam.

Selama menempuh pendidikan, penulis juga pernah aktif di beberapa organisasi, yaitu sebagai anggota Ithri (Rohis) SMAN 5 Bogor dan anggota Lembaga Dakwah Fakultas MIPA (Serum G).