



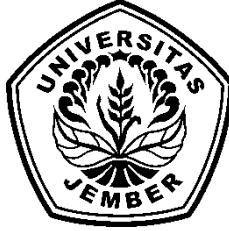
**SIMULASI ARUS LALU LINTAS
DENGAN *CELLULAR AUTOMATA***

SKRIPSI

Oleh

**Abduh Riski
NIM 071810101005**

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2011**



**SIMULASI ARUS LALU LINTAS
DENGAN *CELLULAR AUTOMATA***

SKRIPSI

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Matematika (S1)
dan mencapai gelar Sarjana Sains

Oleh

**Abduh Riski
NIM 071810101005**

**JURUSAN MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS JEMBER
2011**

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Ibunda Ririn Angraini dan almarhum ayahanda Muniri yang tercinta;
2. Guru-guru penulis sejak sekolah dasar sampai dengan perguruan tinggi;
3. “Tulang rusuk” yang selalu memberi penulis semangat;
4. Almamater Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember; SMA Nurul Jadid; MTs Nurul Jadid; SD Negeri 1 Sumber Ketempa.

MOTTO

Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya.
(terjemahan Surat *Al-Baqarah* Ayat 286) *)

*) Departemen Agama Republik Indonesia. 2005. *Al-Qur'an dan Terjemahannya*. Bandung: CV Penerbit J-ART.

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

nama : Abduh Riski

NIM : 071810101005

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Simulasi Arus Lalu Lintas dengan *Cellular Automata*” adalah benar-benar hasil karya sendiri, kecuali kutipan yang telah saya sebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak mana pun serta bersedia mendapat sanksi akademik jika ternyata di kemudian hari pernyataan ini tidak benar.

Jember, 22 Nopember 2011

Yang menyatakan,

Abduh Riski
NIM 071810101005

SKRIPSI

**SIMULASI ARUS LALU LINTAS
DENGAN *CELLULAR AUTOMATA***

Oleh

Abduh Riski
NIM 071810101005

Pembimbing

Dosen Pembimbing Utama : Drs. Moh. Hasan, M.Sc., Ph.D.

Dosen Pembimbing Anggota : Kiswara Agung Santoso, M.Kom.

PENGESAHAN

Skripsi berjudul “Simulasi Arus Lalu Lintas dengan *Cellular Automata*” telah diuji dan disahkan pada:

hari, tanggal :

tempat : Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember

Tim Penguji:

Ketua,

Sekretaris,

Drs. Moh. Hasan, MSc., Ph.D.
NIP 196404041988021001

Kiswara Agung Santoso, M.Kom.
NIP 197209071998031003

Penguji I,

Penguji II,

Kosala Dwidja Purnomo, S.Si., M.Si.
NIP 196908281998021001

Yuliani Setia Dewi, S.Si., M.Si.
NIP 197407162000032001

Mengesahkan
Dekan,

Prof. Drs. Kusno, DEA, M.Sc., Ph.D.
NIP 196101081986021001

RINGKASAN

Simulasi Arus Lalu Lintas dengan *Cellular Automata*; Abduh Riski, 071810101005; 2011: 52 halaman; Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Salah satu alternatif untuk dapat menghasilkan kondisi arus lalu lintas yang aman dan efisien bagi pengguna jalan adalah dengan melakukan pemodelan arus lalu lintas, dan melakukan simulasi yang dapat digunakan untuk menghasilkan rekayasa lalu lintas dengan mudah. Beberapa pemodelan yang pernah dilakukan sebelumnya adalah pemodelan arus lalu lintas dengan Persamaan Diferensial Biasa dan Persamaan Diferensial Parsial. Arus lalu lintas juga pernah dimodelkan dengan *Cellular Automata*. Model arus lalu lintas yang pernah dihasilkan menggunakan *Cellular Automata* sebelumnya antara lain: model *Binary State*, model Nagel-Schreckenberg atau disebut juga model *Multi-Speed*, model *Two-Lane*, dan model *Three-Lane*. Tetapi, semua model arus lalu lintas tersebut mengasumsikan bahwa arus lalu lintas yang dimodelkan hanya berisi kendaraan roda empat. Padahal pada kenyataannya keberadaan kendaraan roda dua sangatlah berpengaruh terhadap kondisi arus lalu lintas, terlebih lagi kondisi arus lalu lintas di Indonesia. Sehingga model-model tersebut kurang begitu dapat merepresentasikan kondisi arus lalu lintas di Indonesia yang sebenarnya. Oleh karena itu, perlu dilakukan pemodelan terhadap arus lalu lintas yang berisi kendaraan roda dua dan kendaraan roda empat. Tujuan dari tugas akhir ini adalah untuk memodelkan arus lalu lintas yang berisi kendaraan roda dua dan kendaraan roda empat menggunakan *Cellular Automata* dan menganalisis relasi antar parameter arus lalu lintas yang terjadi.

Pemodelan dilakukan dalam beberapa langkah berikut; pertama adalah merumuskan tahapan-tahapan yang digunakan untuk memperbarui status sel-sel *Cellular Automata*; kedua, membuat model *Cellular Automata* dari arus lalu lintas yang akan dimodelkan; ketiga adalah membuat algoritma dan program simulasi dari

model arus lalu lintas yang telah diperoleh menggunakan *software Matlab 7.8.0* kemudian melakukan simulasi dan menganalisisnya.

Pemodelan arus lalu lintas yang dilakukan menghasikan model *Cellular Automata* 5-tupel (L, Q, N, δ, C_0) . Dimana L adalah geometri *Cellular Automata*; Q adalah himpunan berhingga status sel *Cellular Automata*; N adalah himpunan sel tetangga; δ adalah fungsi transisi, dimana fungsi transisi yang sama diterapkan pada setiap sel secara berurutan; C_0 adalah status awal sel yang dibangkitkan secara acak.

Cellular Automata dapat digunakan untuk memodelkan arus lalu lintas yang berisi kendaraan roda dua dan kendaraan roda empat dengan baik. Dari simulasi yang dilakukan juga menghasilkan grafik relasi antar parameter arus lalu lintas yaitu kecepatan, kepadatan, dan arus. Relasi arus – kepadatan adalah parabolik, arus bergerak naik seiring bertambahnya kepadatan dan arus akan terus menurun setelah kepadatan tertentu. Relasi kecepatan – kepadatan adalah eksponensial, kecepatan akan terus menurun seiring dengan bertambahnya kepadatan. Relasi kecepatan – arus adalah parabolik, semakin besar kecepatan akan menyebabkan arus terus bergerak naik dan kemudian arus akan terus bergerak turun setelah kecepatan tertentu. Relasi varians kecepatan – kepadatan adalah parabolik, varians kecepatan terus bertambah seiring bertambahnya kepadatan dan varians kecepatan akan terus berkurang setelah kepadatan tertentu.

PRAKATA

Puji syukur ke hadirat Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Simulasi Arus Lalu Lintas dengan *Cellular Automata*”. Skripsi ini disusun untuk memenuhi salah satu syarat untuk menyelesaikan pendidikan strata satu (S1) pada Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Drs. Moh. Hasan, M.Sc., Ph.D. selaku Dosen Pembimbing Utama dan Kiswara Agung Santoso, M.Kom. selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi ini;
2. Kosala Dwidja Purnomo, S.Si., M.Si. selaku Dosen Penguji I dan Yuliani Setia Dewi, S.Si., M.Si. selaku Dosen Penguji II yang telah memberikan kritik dan saran demi kesempurnaan skripsi ini;
3. Refi Ainurrofiq, Aullya Mahadipa, Vina Pramudya Hapsari, Veni Pramudya Hapsari, semua teman-teman angkatan 2007 dan semua teman-teman Jurusan Matematika yang telah memberikan dukungan dan bantuan untuk menyelesaikan tugas akhir ini. Pertanyaan “kapan seminar?” yang sering dilontarkan memberi motivasi tersendiri kepada penulis untuk segera menyelesaikan skripsi ini;
4. semua pihak yang tidak dapat disebutkan satu per satu.

Penulis menerima segala kritik dan saran dari semua pihak demi kesempurnaan skripsi ini. Akhirnya penulis berharap semoga skripsi ini dapat bermanfaat.

Jember, Nopember 2011

Penulis

DAFTAR ISI

| | Halaman |
|--|---------|
| HALAMAN JUDUL | i |
| HALAMAN PERSEMBAHAN | iii |
| HALAMAN MOTTO | iv |
| HALAMAN PERNYATAAN | v |
| HALAMAN PEMBIMBINGAN | vi |
| HALAMAN PENGESAHAN | vii |
| RINGKASAN | viii |
| PRAKATA | x |
| DAFTAR ISI | xi |
| DAFTAR GAMBAR | xiii |
| DAFTAR LAMPIRAN | xiv |
| BAB 1. PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah | 3 |
| 1.3 Tujuan | 4 |
| 1.4 Manfaat | 4 |
| BAB 2. TINJAUAN PUSTAKA | 5 |
| 2.1 Arus Lalu Lintas | 5 |
| 2.2 Cellular Automata | 7 |
| 2.2.1 Definisi <i>Cellular Automata</i> | 7 |
| 2.2.2 Karakteristik <i>Cellular Automata</i> | 12 |
| BAB 3. METODE PENELITIAN | 13 |
| 3.1 Karakteristik Arus Lalu Lintas | 13 |
| 3.2 Langkah Penyelesaian | 14 |

| | |
|--|----|
| BAB 4. HASIL DAN PEMBAHASAN | 17 |
| 4.1 Hasil | 17 |
| 4.1.1 Perumusan Tahapan Perubahan Status Sel | 17 |
| 4.1.2 Pembuatan Model <i>Cellular Automata</i> | 22 |
| 4.1.3 Pembuatan Algoritma dan Program | 24 |
| 4.2 Pembahasan | 38 |
| BAB 5. PENUTUP | 50 |
| 5.1 Kesimpulan | 50 |
| 5.2 Saran | 51 |
| DAFTAR PUSTAKA | |

DAFTAR GAMBAR

| | Halaman |
|---|---------|
| 1.1 Ilustrasi arus lalu lintas yang akan dimodelkan | 4 |
| 2.1 Geometri <i>Cellular Automata</i> | 9 |
| 2.2 <i>Neighbourhood Cellular Automata</i> | 10 |
| 2.3 <i>Cellular Automata</i> dua dimensi | 11 |
| 3.1 Diagram metode penelitian | 13 |
| 4.1 Ilustrasi model arus lalu lintas dengan <i>Cellular Automata</i> | 21 |
| 4.2 Diagram alir model arus lalu lintas dengan <i>Cellular Automata</i> | 25 |
| 4.3 Diagram alir tahap penentuan posisi awal kendaraan | 26 |
| 4.4 Diagram alir tahap pindah lajur | 27 |
| 4.5 Diagram alir tahap pindah lajur (lanjutan) | 28 |
| 4.6 Diagram alir tahap percepatan | 29 |
| 4.7 Diagram alir tahap perlambatan | 30 |
| 4.8 Diagram alir tahap pindah sel | 31 |
| 4.9 GUI program RodaDua | 35 |
| 4.10 GUI <i>output</i> program RodaDua | 37 |
| 4.11 GUI <i>output</i> akhir program RodaDua | 37 |
| 4.12 Perubahan sel model arus lalu lintas dengan <i>Cellular Automata</i> | 39 |
| 4.13 Contoh kasus kendaraan berkurang kecepatannya dan berhenti | 40 |
| 4.14 Contoh kasus tahap pindah lajur | 41 |
| 4.15 Contoh kasus tahap perlambatan | 42 |
| 4.16 Diagram fundamental arus lalu lintas | 43 |
| 4.17 Grafik relasi arus dan kepadatan | 44 |
| 4.18 Grafik relasi arus dan kepadatan Bellemans | 45 |
| 4.19 Grafik relasi kecepatan dan kepadatan | 46 |
| 4.20 Grafik relasi kecepatan dan kepadatan Bellemans | 47 |
| 4.21 Grafik relasi arus dan kecepatan | 48 |
| 4.22 Grafik relasi varians kecepatan dan kepadatan | 49 |

DAFTAR LAMPIRAN

| | Halaman |
|---|---------|
| A. <i>Listing</i> Program GUI RodaDua | 53 |
| B. <i>Listing</i> Program Tahap Penentuan Posisi Awal Kendaraan | 61 |
| C. <i>Listing</i> Program Tahap Pindah Lajur | 63 |
| D. <i>Listing</i> Program Tahap Percepatan | 67 |
| E. <i>Listing</i> Program Tahap Perlambatan | 68 |
| F. <i>Listing</i> Program Tahap Pindah Sel | 69 |
| G. Tabel Hasil Simulasi | 70 |

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Hampir setiap hari, jalan raya penuh dilalui pengemudi kendaraan bermotor untuk mencapai tempat tujuannya masing-masing. Jalan raya pada umumnya dapat berupa jalan besar yang terdiri atas banyak lajur ataupun juga berupa jalan yang hanya terdiri atas satu lajur saja, yang tentunya dilewati oleh kendaraan bermotor, misalnya mobil dan sepeda motor. Banyak kejadian atau interaksi yang terjadi di jalan raya yang kemudian interaksi tersebut disebut dengan arus lalu lintas. Bentuk interaksi antar kendaraan antara lain: sebuah kendaraan berjalan mendahului kendaraan yang lain dengan melakukan perpindahan lajur ke sebelah kanannya, pengereman, atau kejadian lain yang sering terjadi dalam berlalu lintas di jalan raya. Dalam arus lalu lintas juga terdapat parameter yang berpengaruh terhadap kondisi arus lalu lintas yang terjadi. Parameter tersebut adalah kecepatan, kepadatan, dan arus (Roess *et al.*, 2004). Dari parameter tersebut kemudian arus lalu lintas kendaraan di jalan raya dapat juga disebut sebagai suatu sistem *self organizing* karena kecepatan kendaraan di jalan raya pada suatu saat dapat dianggap hanya dipengaruhi oleh kecepatan kendaraan itu dan keadaan jalan di sekitarnya pada saat sebelumnya.

Tingginya tingkat mobilitas manusia menyebabkan semakin dibutuhkannya keberadaan jalan raya sebagai media sarana transportasi yang dapat memenuhi keperluan banyak pengguna jalan raya, tentunya hal tersebut harus disertai oleh kondisi jalan raya yang aman dan efisien. Salah satu alternatif untuk dapat menghasilkan kondisi jalan raya yang dimaksud tersebut adalah dengan melakukan pemodelan arus lalu lintas, dan melakukan simulasi yang dapat digunakan untuk menghasilkan rekayasa lalu lintas dengan mudah. Pemodelan arus lalu lintas dapat bersifat makroskopik atau mikroskopik. Pemodelan arus lalu lintas secara makroskopik adalah pemodelan arus lalu lintas yang terjadi pada sejumlah besar

kendaraan pada suatu ruas jalan, dimana pada pemodelan ini dapat dilihat parameter seperti kepadatan, kecepatan dan arus. Pemodelan arus lalu lintas secara mikroskopik adalah pemodelan yang memodelkan perilaku pengemudi dalam berinteraksi dengan kendaraan lain di depannya pada suatu jalan, seperti pemodelan dengan *car following* model dimana pada model ini dapat dilihat parameter seperti posisi dan kecepatan individual.

Pemodelan arus lalu lintas menggunakan Persamaan Diferensial Parsial (PDP) oleh Sudradjat *et al.* (2008) dan pemodelan arus lalu lintas menggunakan Model Makroskopik dan Mikroskopik dalam Persamaan Diferensial Biasa (PDB) oleh Susiliawati (2009) merupakan contoh dari pemodelan arus lalu lintas yang telah dilakukan sebelumnya. Pemodelan menggunakan PDP dan PDB sebenarnya telah dapat menghasilkan persamaan diferensial yang dapat memodelkan arus lalu lintas secara mikroskopik dan makroskopik. Akan tetapi, pemodelan yang telah dilakukan menggunakan kedua metode tersebut terbatas untuk arus lalu lintas yang di dalamnya tidak terdapat kendaraan yang saling mendahului kendaraan lainnya, serta pemodelan yang dilakukan hanya untuk arus lalu lintas yang berisi kendaraan sejenis (dalam hal ini adalah kendaraan roda empat).

Arus lalu lintas juga dapat dimodelkan dengan *Cellular Automata* (CA), karena arus lalu lintas adalah suatu sistem *self organizing* yang sesuai dengan karakteristik *Cellular Automata* dimana status suatu sel pada suatu saat hanya ditentukan oleh status sel itu sendiri dan status sel-sel tetangganya pada saat sebelumnya. Nagel dan Schreckenberg merupakan Matematikawan dan Fisikawan Jerman yang memelopori pemodelan arus lalu lintas menggunakan *Cellular Automata*. Pada tahun 1992 kedua peneliti ini merumuskan suatu model sederhana dari arus lalu lintas yang kemudian dikenal sebagai model Nagel-Schreckenberg. Model ini telah diterapkan untuk menyimulasikan arus lalu lintas di jalan perkotaan dan jalan tol Jerman.

Beberapa model arus lalu lintas dengan *Cellular Automata* yang telah dimodelkan sebelumnya diantaranya adalah: model *Binary State* (Syahid, 2003), model Nagel-Schreckenberg atau disebut juga model *Multi-Speed* (Nagel dan

Schreckenberg, 1992), model *Two-Lane* (Syalim, 2003), dan model *Three-Lane* (Daoudia dan Moussa, 2003). Model *Binary State* mengasumsikan bahwa kendaraan dalam arus lalu lintas yang dimodelkan memiliki kecepatan konstan, sedangkan dalam model *Multi-Speed* mengasumsikan kendaraan dalam arus lalu lintas yang dimodelkan memiliki kecepatan yang beragam. Kedua model tersebut memodelkan arus lalu lintas pada jalan yang terdiri dari satu lajur searah dan tidak terdapat kendaraan yang saling mendahului dalam arus lalu lintas yang dimodelkan. Sedangkan model *Two-Lane* dan *Three-Lane* merupakan pengembangan dari model *Multi-Speed*. Dalam kedua model tersebut, arus lalu lintas yang dimodelkan terdapat pada jalan yang terdiri dari dua lajur searah dan tiga lajur searah. Serta kendaraan yang terdapat di dalamnya dapat saling mendahului kendaraan lainnya. Tetapi, dalam keempat model arus lalu lintas dengan *Cellular Automata* tersebut diasumsikan bahwa arus lalu lintas yang dimodelkan hanya berisi kendaraan roda empat. Padahal pada kenyataannya keberadaan kendaraan roda dua sangatlah berpengaruh terhadap kondisi arus lalu lintas, terlebih lagi kondisi arus lalu lintas di Indonesia. Sehingga keempat model tersebut kurang begitu dapat merepresentasikan kondisi arus lalu lintas di Indonesia yang sebenarnya.

1.2 Rumusan Masalah

Permasalahan yang akan dibahas dalam tugas akhir ini adalah bagaimana memodelkan arus lalu lintas yang berisi kendaraan roda dua dan kendaraan roda empat (selanjutnya dalam skripsi ini disebut sebagai arus lalu lintas saja) dengan menggunakan *Cellular Automata*, kemudian menyimulasikannya dan menganalisis relasi yang terjadi antar parameter arus lalu lintas. Arus lalu lintas yang dimaksud adalah arus lalu lintas satu arah pada jalan lurus dengan lebar jalan yang memungkinkan setiap kendaraan mendahului kendaraan lainnya. Kondisi arus lalu lintas yang akan dimodelkan dapat diilustrasikan dalam Gambar 1.1.

BAB 2. TINJAUAN PUSTAKA

2.1 Arus Lalu Lintas

Arus lalu lintas terbentuk oleh individual pengemudi dan kendaraan, yang berinteraksi dengan cara yang unik antara satu dengan yang lain dan dengan elemen-elemen jalan serta lingkungannya. Karena persepsi dan kemampuan individu pengemudi mempunyai sifat yang berbeda, maka perilaku kendaraan di dalam arus lalu lintas tidak dapat diseragamkan. Arus lalu lintas akan mengalami perbedaan karakteristik akibat dari perbedaan perilaku pengemudi dikarenakan oleh karakteristiknya yang bervariasi baik berdasarkan lokasi maupun waktu.

Arus lalu lintas dapat dikelompokkan menjadi dua bagian, yaitu: arus lalu lintas tidak terganggu dan arus lalu lintas terganggu. Arus lalu lintas tidak terganggu adalah arus lalu lintas yang hanya ditentukan oleh interaksi kendaraan dengan kendaraan lainnya dan interaksi kendaraan dengan jalan, seperti arus lalu lintas kendaraan di jalan tol. Sedangkan arus lalu lintas terganggu adalah arus lalu lintas yang ditentukan oleh alat atau cara dari luar arus lalu lintas misalnya lampu lalu lintas atau marka jalan. Interaksi kendaraan dengan kendaraan lainnya dan interaksi kendaraan dengan jalan mempunyai peranan kedua dalam menentukan arus lalu lintas terganggu (Roess *et al.*, 2004).

Dalam rangka untuk mengerti tentang keberagaman karakteristik arus lalu lintas, perlu didefinisikan parameter yang penting dan terukur. Arus lalu lintas dipengaruhi oleh banyak parameter, tetapi secara garis besar dalam tulisan ini terdapat tiga parameter arus lalu lintas, yaitu kecepatan, kepadatan, dan arus.

a. Kecepatan

Kecepatan adalah jarak tempuh per-satuan waktu, secara matematis ditulis:

$$v = \frac{s}{t}, \quad (2.1)$$

dimana: v = kecepatan, s = jarak tempuh, t = waktu tempuh. Satuan dari kecepatan adalah km/jam atau m/s.

Terdapat dua macam kecepatan yaitu kecepatan individual dan kecepatan rata-rata. Kecepatan individual adalah kecepatan sebuah kendaraan di suatu titik (jalan) pada saat t , ditulis $v(s, t)$. Sedangkan kecepatan rata-rata diperoleh dengan merata-ratakan kecepatan individual semua kendaraan dalam suatu jalan.

b. Kepadatan

Kepadatan adalah perbandingan antara jumlah kendaraan yang ada dalam suatu ruas jalan dengan panjang ruas jalan tersebut. Kepadatan dilambangkan dengan ρ dan dirumuskan sebagai berikut:

$$\rho = \frac{n}{\sum_{i=1}^n (L_i + gap_i)}, \quad (2.2)$$

dimana: n = banyak kendaraan, L_i = panjang kendaraan ke- i , gap_i = jarak antar kendaraan ke- i dan kendaraan di depannya (Roess *et al.*, 2004). Satuan dari kepadatan adalah kendaraan/km atau smp/km (smp = satuan mobil penumpang).

Jika diasumsikan panjang setiap kendaraan dan jarak antar kendaraan adalah sama, misalkan panjang masing-masing kendaraan adalah $L = L_1 = L_2 = \dots = L_n$ dan jarak antar kendaraan adalah $gap = gap_1 = gap_2 = \dots = gap_n$, maka persamaan (2.2) menjadi:

$$\rho = \frac{1}{L + gap}. \quad (2.3)$$

Kepadatan tinggi menunjukkan jarak antar kendaraan cukup dekat atau $gap \approx 0$, kepadatan rendah berarti jarak antar kendaraan cukup jauh.

c. Arus

Arus adalah jumlah kendaraan yang melalui suatu jalan tiap interval waktu (Roess *et al.*, 2004). Karena ρ adalah jumlah kendaraan tiap panjang jalan pada suatu jalan tertentu, sedangkan jarak yang ditempuh adalah $v \times t$, maka jumlah kendaraan yang melewati jalan tersebut dalam waktu t adalah $\rho \times v \times t$. Jadi arus adalah

$$q = \rho \times v. \quad (2.4)$$

Satuan dari arus adalah kendaraan/jam atau smp/jam. Misalkan banyak kendaraan pada suatu jalan tertentu selama 15 menit adalah 100 kendaraan roda empat maka arus lalu lintas pada jalan tersebut adalah $100 \times 4 = 400$ smp/jam.

Karena arus merupakan hasil perkalian kepadatan dan kecepatan, maka nilai arus akan sama dengan 0 bila $\rho = 0$ atau $v = 0$. Dapat juga dinyatakan bahwa q akan maksimum pada kombinasi optimum antara ρ dan v . Sebagai ilustrasi, tinjau keadaan berikut:

- a. Kemacetan arus lalu lintas terjadi pada saat dimana ρ sangat tinggi dan v sangat rendah, kombinasi ini menghasilkan q sangat rendah.
- b. Sebaliknya, q sangat tinggi terjadi bila ρ sangat rendah sehingga pengemudi dapat mencapai kecepatan maksimum tanpa khawatir keberadaan kendaraan lain di jalan.

2.2 Cellular Automata

Cellular Automata pertama kali diperkenalkan oleh Von Neumann pada tahun 1948 yang digunakan untuk memodelkan evolusi biologis *self-reproduction* secara sederhana. *Cellular Automata* telah digunakan secara luas sebagai model untuk menyelesaikan sistem yang kompleks. Secara praktis, *Cellular Automata* berkembang ketika pada tahun 1970 Conway membuat *Game Of Life* yang mampu memodelkan kehidupan nyata secara sederhana (Ilachinski, 2001).

2.2.1 Definisi Cellular Automata

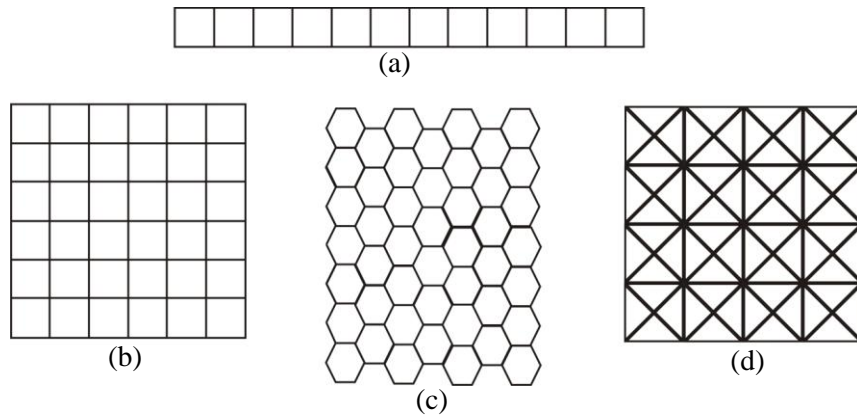
Cellular Automata adalah suatu model dari sistem fisik dimana ruang dan waktu bersifat diskrit, dan interaksi yang terjadi bersifat lokal. Disamping kesederhanaan strukturnya, *Cellular Automata* juga dapat digunakan untuk memperlihatkan fenomena kompleks yang dinamis dan dapat menerangkan berbagai macam proses dan sistem fisik (Benjaafar *et al.*, 1997).

Cellular Automata terdiri dari sel-sel yang seragam dan tetap dalam n -dimensi *array* ($n = 1,2,3, \dots$). Biasanya dipergunakan *Cellular Automata* satu atau dua dimensi karena kesulitan dalam memvisualisasikan model dengan dimensi yang lebih tinggi. Sel-sel tersebut merupakan representasi dari sistem fisik dan masing-masing sel memiliki nilai atau yang disebut dengan keadaan (status). Sel-sel tersebut akan berinteraksi secara lokal dengan sel-sel tetangganya. Status tiap sel akan diperbarui secara serentak pada tiap *time-step* berdasarkan status sel itu sendiri dan status sel-sel tetangganya. Fungsi yang digunakan untuk menghitung status dari sel pada *time-step* selanjutnya ditentukan oleh suatu aturan-aturan yang disebut fungsi transisi. Biasanya fungsi transisi yang sama diterapkan pada semua sel *Cellular Automata*.

Secara umum menurut Maerivoet dan Moor (2005) setidaknya terdapat empat unsur pembentuk *Cellular Automata*, sedangkan menurut Syalim (2003) terdapat unsur pembentuk *Cellular Automata* yang kelima yaitu "status awal sel", sehingga unsur-unsur pembentuk *Cellular Automata* adalah sebagai berikut:

a. Geometri

Geometri adalah bentuk sel serta bentuk sistem yang disusun oleh sel-sel tersebut. Geometri *Cellular Automata* terdiri atas dimensi *Cellular Automata* tersebut dalam n -dimensi *array* ($n = 1,2,3, \dots$), dan bentuk geometri dari masing-masing sel penyusunnya. Pada Gambar 2.1 adalah beberapa contoh bentuk geometri sel *Cellular Automata* yang sering digunakan. Tetapi, bentuk geometri sel *Cellular Automata* yang biasa digunakan untuk merepresentasikan jalan dalam simulasi arus lalu lintas adalah sel segi empat satu dimensi atau sel segi empat dua dimensi seperti pada Gambar 2.1 (a) dan Gambar 2.1 (b).



(a) segi empat satu dimensi;
 (b) segi empat dua dimensi; (c) segi enam dua dimensi; (d) segi tiga dua dimensi

Gambar 2.1 Geometri *Cellular Automata*

b. Status Sel

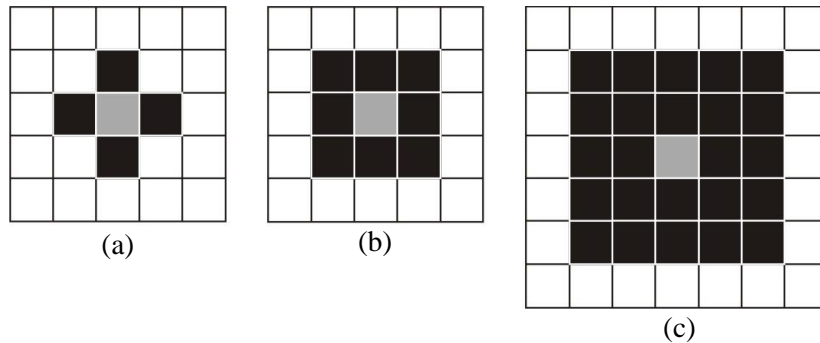
Status sel adalah keadaan sel yang dapat dimiliki oleh masing-masing sel *Cellular Automata*. Status sel dapat berupa angka maupun sifat tertentu. Misalkan jika masing-masing sel merepresentasikan bagian jalan raya yang berukuran cukup untuk satu buah kendaraan, maka status sel dapat berupa keberadaan kendaraan di bagian jalan tersebut (jalan terisi oleh kendaraan atau jalan kosong), yang dapat dilambangkan dengan bilangan 0 untuk jalan dalam keadaan kosong dan 1 untuk jalan yang terisi kendaraan.

c. *Neighbourhood*

Neighbourhood adalah sel-sel yang dapat mempengaruhi status suatu sel *Cellular Automata*. Umumnya *neighbourhood* suatu sel hanya meliputi sel-sel yang berada di sekitarnya.

Dimungkinkan terdapat banyak definisi tentang struktur dari *neighbourhood*. Namun, secara umum terdapat dua macam *neighbourhood* yang telah dikenal dalam *Cellular Automata* dua dimensi, yakni Von Neumann *neighborhood* dan Moore *neighborhood*. Pada model Von Neumann *neighborhood* (Gambar 2.2 (a)) sebuah sel (berwarna abu-abu) mempunyai empat sel tetangga (berwarna hitam) yang berada pada arah utara, timur, selatan, dan barat. Sedangkan model Moore *neighborhood* dengan jari-jari *neighbourhood* (dilambangkan dengan r) sama

dengan 1 dan mempunyai delapan sel tetangga dengan tambahan di arah barat laut, tenggara, barat daya, dan timur laut (Gambar 2.2 (b)).



(a) Von Neumann *neighborhood*; (b) Moore *neighborhood* dengan $r = 1$;
(c) Moore *neighbourhood* dengan $r = 2$

Gambar 2.2 *Neighbourhood Cellular Automata*

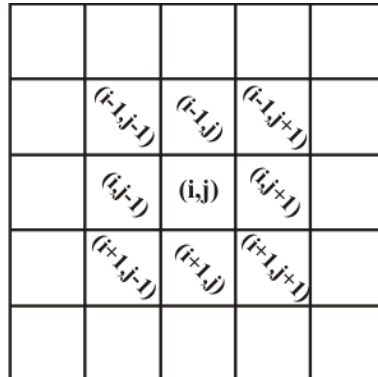
d. Fungsi Transisi

Fungsi transisi adalah aturan yang menentukan bagaimana status suatu sel berubah berdasarkan status sel tersebut dan status sel tetangganya sekarang. Fungsi transisi diterapkan terhadap semua sel secara serentak (paralel). Pada umumnya fungsi transisi yang sama diterapkan kepada semua sel.

Status dari suatu sel untuk *time-step* $t + 1$ dihitung berdasarkan suatu fungsi transisi δ dari status sel tersebut dan status dari sel-sel tetangganya pada *time-step* t . Sebagai ilustrasi, misalkan terdapat sebuah *Cellular Automata* dua dimensi seperti pada Gambar 2.3. Pada contoh ini sel (i, j) mempunyai delapan sel tetangga yang berdekatan. Fungsi transisi *Cellular Automata* yang akan digunakan untuk menghitung sel (i, j) pada *time-step* $t + 1$ merupakan fungsi dari status sel (i, j) itu sendiri dan status sel-sel tetangganya pada *time-step* t . Jadi,

$$C_{i,j}^{t+1} = \delta(C_{i-1,j-1}^t, C_{i-1,j}^t, C_{i-1,j+1}^t, C_{i,j-1}^t, C_{i,j}^t, C_{i,j+1}^t, C_{i+1,j-1}^t, C_{i+1,j}^t, C_{i+1,j+1}^t) \quad (2.5)$$

dimana $C_{i,j}^t$ dan $C_{i,j}^{t+1}$ adalah status sel (i, j) pada *time-step* t dan $t + 1$ (Ilachinski, 2001).



Gambar 2.3 *Cellular Automata* dua dimensi

e. Status Awal Sel

Status awal sel adalah status yang dimiliki oleh masing-masing sel pada saat sistem mulai berjalan. Dalam simulasi atau pemodelan menggunakan *Cellular Automata* status awal sel dapat dibangkitkan secara random.

Seperti pada penjelasan di atas bahwa sebuah *Cellular Automata* terdiri dari lima unsur pembentuk. Sehingga *Cellular Automata* dapat dituliskan sebagai 5-pasangan berurutan (tupel) (L, Q, N, δ, C_0) , dimana:

- L : Adalah geometri *Cellular Automata* yaitu bentuk sel serta bentuk sistem yang disusun oleh sel-sel tersebut dengan d adalah dimensi dari bentuk geometri tersebut. Pada Gambar 2.1 (b), (c), dan (d) merupakan *Cellular Automata* dua dimensi sehingga $d = 2$.
- Q : Himpunan berhingga status sel.
- N : Himpunan sel yang mempengaruhi status suatu sel pada langkah berikutnya dengan n adalah jumlah sel tetangga (*neighbours*) yang mempengaruhi sel tersebut dan r adalah jari-jari *neighbourhood*. Pada Gambar 2.2 (a) mempunyai nilai $n = 4$ dan $r = 1$.
- δ : Fungsi transisi yang diterapkan pada sistem untuk setiap *time-step*. Pada tiap *time-step* fungsi diterapkan serentak pada semua sel. *Input* fungsi ini

adalah status dirinya sendiri dan status sel tetangganya. Sehingga fungsi transisi dapat didefinisikan sebagai:

$$\delta: Q^{n+1} \rightarrow Q$$

Dimana Q adalah status suatu sel pada *time-step* ke- $t + 1$ dan Q^{n+1} adalah himpunan status sel tersebut dan status sel-sel tetangganya pada *time-step* ke- t .

C_0 : Kondisi awal sel, kondisi awal adalah himpunan yang berisi status tiap-tiap sel dari *Cellular Automata* pada *time-step* ke-0 atau pada waktu sistem mulai berjalan.

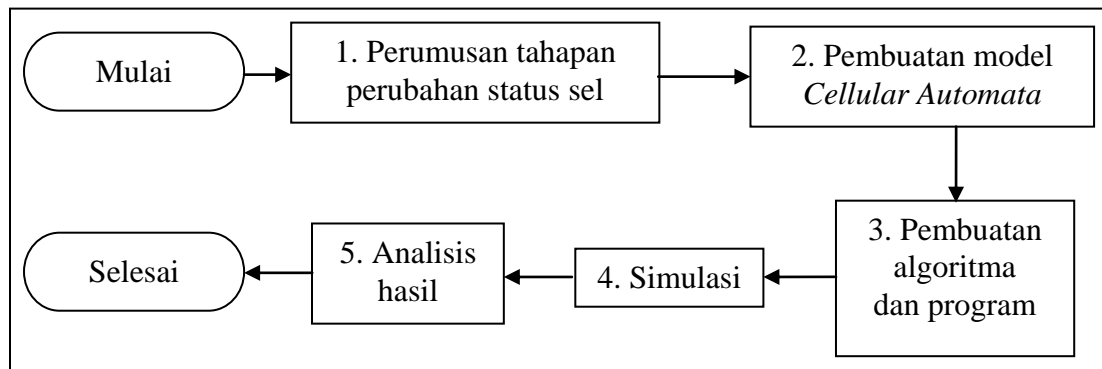
2.2.2 Karakteristik *Cellular Automata*

Cellular Automata memiliki karakteristik-karakteristik umum yang menggambarkan sistem *Cellular Automata*. Menurut Ilachinski (2001), *Cellular Automata* memiliki beberapa karakteristik umum, diantaranya:

- a. Bentuk geometri sel diskrit, bentuk geometri sistem *Cellular Automata* terdiri dari n -dimensi *array* ($n = 1, 2, 3, \dots$).
- b. Homogenitas, semua sel dalam *Cellular Automata* adalah ekuivalen.
- c. Status sel diskrit, status setiap sel berupa salah satu dari himpunan terbatas keadaan diskrit.
- d. Interaksi lokal, setiap sel hanya berinteraksi dengan sel-sel yang berada pada lingkungan lokal (sel-sel tetangganya).
- e. Dinamik diskrit, status setiap sel diperbarui dalam setiap satuan waktu diskrit menurut sebuah fungsi transisi yang telah didefinisikan dengan mempertimbangkan status sel-sel tetangganya.

BAB 3. METODE PENELITIAN

Secara umum langkah-langkah yang akan dilakukan untuk menyelesaikan permasalahan pada subbab 1.2 terdiri dari beberapa langkah; pertama adalah merumuskan tahapan-tahapan yang digunakan untuk memperbarui status sel-sel *Cellular Automata*; kedua, membuat model *Cellular Automata* dari arus lalu lintas yang akan dimodelkan; ketiga adalah membuat algoritma dan program simulasi dari model arus lalu lintas yang telah diperoleh menggunakan *software Matlab 7.8.0* kemudian melakukan simulasi dan menganalisisnya. Secara skematik, langkah-langkah tersebut dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram metode penelitian

3.1 Karakteristik Arus Lalu Lintas

Arus lalu lintas yang akan dimodelkan adalah arus lalu lintas seperti telah dijelaskan dalam Bab 1. Arus lalu lintas tersebut adalah arus lalu lintas yang juga memperhitungkan keberadaan kendaraan roda dua, sehingga arus lalu lintas yang akan dimodelkan adalah arus lalu lintas yang berisi kendaraan roda empat (dalam hal ini adalah mobil) dan roda dua. Arus lalu lintas yang akan dimodelkan dapat diilustrasikan seperti pada Gambar 1.1 yaitu arus lalu lintas tersebut terjadi pada jalan

lurus satu arah, dan jalan tersebut mempunyai lebar yang cukup untuk dilalui dua kendaraan roda empat atau empat kendaraan roda dua secara bersamaan.

Kendaraan dalam arus lalu lintas yang akan dimodelkan dapat mendahului kendaraan lainnya dengan berpindah ke lajur di sebelahnya. Hal ini terjadi jika sebuah kendaraan berjalan lebih cepat daripada kendaraan di depannya dan berjarak cukup dekat antar keduanya, tentunya juga bila pengemudi kendaraan tersebut berkeinginan untuk mendahului kendaraan di depannya. Syarat lain yang harus dipenuhi sebelum sebuah kendaraan berpindah lajur untuk mendahului kendaraan lainnya adalah terdapat kondisi aman. Kondisi aman terpenuhi jika pada lajur sebelah di depan kendaraan tersebut tidak terdapat kendaraan lain sehingga kendaraan tersebut dapat mempertahankan atau menambah kecepatannya jika berpindah lajur, serta pada lajur sebelah di belakang kendaraan tersebut tidak terdapat kendaraan lainnya yang berjarak dekat dan memiliki kecepatan yang lebih tinggi. Namun, bila tidak memungkinkan untuk berpindah lajur, haruslah pengemudi kendaraan tersebut menurunkan kecepatan kendaraannya hingga tidak akan terjadi tabrakan dengan kendaraan di depannya.

Karakteristik lain dari arus lalu lintas yang akan dimodelkan adalah kecepatan dari setiap kendaraan dapat terus bertambah hingga mencapai kecepatan maksimal yang telah ditentukan. Hal ini terjadi bila kecepatan sebuah kendaraan kurang dari kecepatan maksimal dan kurang dari *gap* kendaraan tersebut dengan kendaraan di depannya. Namun dalam hal ini terdapat unsur probabilistik yang dapat menyebabkan pengemudi kendaraan tetap mempertahankan atau menurunkan kecepatan kendaraannya.

3.2 Langkah Penyelesaian

1. Merumuskan Tahapan Perubahan Status Sel

Tahapan perubahan status sel diperoleh dengan merubah karakteristik arus lalu lintas menjadi langkah-langkah sistematis, sehingga lebih mudah untuk

memahami langkah-langkah memperbarui posisi kendaraan dalam arus lalu lintas yang dimodelkan.

2. Membuat Model *Cellular Automata*

Pada langkah ini dibuat model *Cellular Automata* dari arus lalu lintas sebagai 5-tupel (L, Q, N, δ, C_0) , dimana:

L : Geometri sel.

Q : Himpunan berhingga status sel.

N : Himpunan sel yang mempengaruhi terhadap status setiap sel.

δ : Fungsi transisi yang akan digunakan untuk memperbarui status sel yang sesuai dengan tahapan perubahan status sel pada langkah 1 dan diterapkan secara bersamaan (paralel) pada semua sel .

C_0 : Keadaan awal sistem. Keadaan awal sistem berupa posisi awal kendaraan di jalan dalam keadaan diam/berhenti beserta jumlah kendaraan yang terdapat pada jalan tersebut. Keadaan awal sistem dibangkitkan secara random dengan mempertimbangkan kepadatan arus lalu lintas, persentase jumlah kendaraan dan panjang jalan dari arus lalu lintas yang dimodelkan.

3. Pembuatan Algoritma dan Program

Algoritma dibuat berdasarkan model *Cellular Automata* yang dibuat sebelumnya. Kemudian, pembuatan program dilakukan menggunakan *software Matlab 7.8.0* dari algoritma yang telah diperoleh. Dalam program yang akan dibuat terdiri dari beberapa *subroutine*. *Subroutine-subroutine* tersebut memiliki fungsi sebagai berikut:

- a. *Subroutine* yang berfungsi untuk melakukan pengacakan posisi kendaraan pada awal iterasi.
- b. *Subroutine* yang berfungsi untuk memindahkan posisi kendaraan ke lajur di sebelahnya untuk mendahului kendaraan di depannya, yang terlebih dahulu dipastikan apakah kendaraan tersebut dapat mendahului kendaraan di depannya.

- c. *Subroutine* yang berfungsi untuk menaikkan kecepatan kendaraan jika telah terpenuhi kondisi yang disyaratkan dalam model.
- d. *Subroutine* yang berfungsi untuk menurunkan kecepatan kendaraan jika telah terpenuhi kondisi yang disyaratkan dalam model.
- e. *Subroutine* yang berfungsi untuk memindahkan kendaraan ke depan sesuai dengan hasil dari *subroutine- subroutine* sebelumnya untuk *time-step* selanjutnya.

Output dari program berupa barisan teks yang merepresentasikan pergerakan arus lalu lintas yang dimodelkan dalam setiap *time-step* yang berlangsung.

4. Simulasi

Simulasi dilakukan dengan memasukkan *input* data ke dalam program yang telah dibuat. *Input* yang akan dimasukkan ke dalam program meliputi panjang sel yang merepresentasikan panjang jalan, persentase kepadatan jalan, persentase jumlah kendaran roda dua, persentase jumlah kendaraan roda empat, probabilitas pengemudi menurunkan kecepatannya, probabilitas pengemudi berpindah ke lajur di sebelahnya, banyak *time-step* yang diinginkan, dan kecepatan maksimal yang dapat dilakukan oleh kendaraan.

BAB 4. HASIL DAN PEMBAHASAN

4.1 Hasil

4.1.1 Perumusan Tahapan Perubahan Status Sel

Dari penjelasan tentang karakteristik arus lalu lintas diketahui bahwa jalan lurus satu arah pada arus lalu lintas yang akan dimodelkan dapat dipandang sebagai jalan yang memiliki empat lajur semu, karena jalan tersebut dapat dilalui maksimal oleh empat kendaraan roda dua secara bersamaan. Sehingga jalan pada arus lalu lintas yang akan dimodelkan dapat direpresentasikan sebagai sel *Cellular Automata* yang memiliki geometri segi empat dua dimensi dalam empat baris sel. Setiap sel *Cellular Automata* dapat terisi sebuah kendaraan roda dua ataupun kosong, sedangkan untuk kendaraan roda empat menempati dua buah sel sekaligus tepat berdampingan pada baris yang berbeda. Pada jalan sebenarnya setiap sel mempunyai panjang 5 meter. Misalkan sel *Cellular Automata* mempunyai panjang 100 sel, maka pada kenyataannya sel tersebut merepresentasikan jalan yang mempunyai panjang 500 meter atau sama dengan $\frac{1}{2}$ kilometer.

Status setiap sel berupa bilangan bulat tak negatif dari himpunan

$$Q = \{0, 1, 2, 3, 4, \dots, v_{max}, 100\} \quad (4.1)$$

yang merupakan representasi dari kecepatan kendaraan, dengan v_{max} adalah batas atas kecepatan maksimal kendaraan dalam model. Status sel juga berarti jarak suatu sel akan dipindah ke depan untuk *time-step* selanjutnya. Status sel dilambangkan dengan huruf abjad, huruf kapital untuk melambangkan kecepatan kendaraan roda empat dan huruf kecil untuk melambangkan kecepatan kendaraan roda dua. Sehingga lambang "a" berarti sel tersebut terisi kendaraan roda dua dengan status 0, lambang "b" berarti sel tersebut terisi kendaraan roda dua dengan status 1, dan seterusnya hingga lambang "z" berarti sel tersebut terisi kendaraan roda dua dengan status 25.

Sedangkan lambang "A" berarti sel tersebut terisi kendaraan roda empat dengan status 0, lambang "B" berarti sel tersebut terisi kendaraan roda empat dengan status 1, dan seterusnya hingga lambang "Z" berarti sel tersebut terisi kendaraan roda empat dengan status 25. Sedangkan untuk sel yang tidak terisi kendaraan memiliki status 100 yang dilambangkan dengan ".".

Status sel "b" atau "B" pada model bila dikonversi menjadi kecepatan kendaraan sebenarnya adalah sama dengan 5 m/s. Hal ini terjadi karena setiap sel mempunyai panjang 5 m pada jalan sebenarnya dan satuan waktu yang digunakan adalah sekon. Jika dikonversikan lagi dalam satuan km/jam, maka

$$5 \text{ m/s} \equiv 5 \cdot \frac{3600}{1000} \text{ km/jam} = 18 \text{ km/jam}$$

Untuk lebih jelasnya lihat Tabel 4.1.

Tabel 4.1 Korespondensi status sel model arus lalu lintas dengan kecepatan kendaraan sebenarnya.

| Kecepatan | | | Kecepatan | | |
|------------|--------|------------|------------|--------|------------|
| Status sel | Simbol | Sebenarnya | Status sel | Simbol | Sebenarnya |
| 0 | a/A | 0 km/jam | 13 | n/N | 234 km/jam |
| 1 | b/B | 18 km/jam | 14 | o/O | 252 km/jam |
| 2 | c/C | 36 km/jam | 15 | p/P | 270 km/jam |
| 3 | d/D | 54 km/jam | 16 | q/Q | 288 km/jam |
| 4 | e/E | 72 km/jam | 17 | r/R | 306 km/jam |
| 5 | f/F | 90 km/jam | 18 | s/S | 324 km/jam |
| 6 | g/G | 108 km/jam | 19 | t/T | 342 km/jam |
| 7 | h/H | 126 km/jam | 20 | u/U | 360 km/jam |
| 8 | i/I | 144 km/jam | 21 | v/V | 378 km/jam |
| 9 | j/J | 162 km/jam | 22 | w/W | 396 km/jam |
| 10 | k/K | 180 km/jam | 23 | x/X | 414 km/jam |
| 11 | l/L | 198 km/jam | 24 | y/Y | 432 km/jam |
| 12 | m/M | 216 km/jam | 25 | z/Z | 450 km/jam |

1. Huruf kapital melambangkan kendaraan roda empat.
2. Huruf kecil melambangkan kendaraan roda dua.

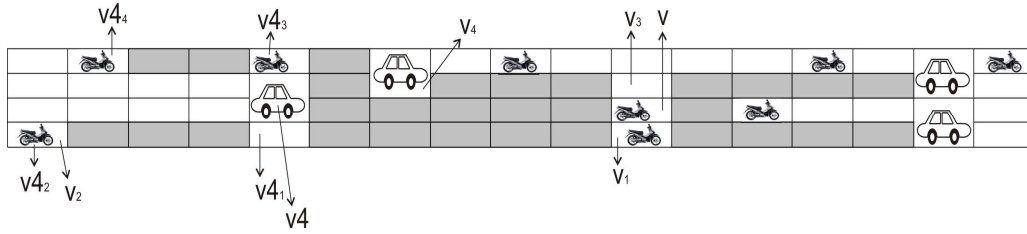
Pada subsubbab 4.1.1 telah dijelaskan bahwa setidaknya terdapat dua karakteristik arus lalu lintas yang akan dimodelkan yaitu perpindahan lajur kendaraan untuk mendahului kendaraan lainnya dan akselerasi/deselerasi kecepatan kendaraan. Kedua karakteristik tersebut dipengaruhi oleh faktor keadaan jalan dan faktor pengemudi. Misalkan v adalah status sel yang terisi kendaraan roda dua dan selanjutnya sel tersebut disebut dengan sel v . Faktor keadaan jalan untuk sel v dimodelkan dengan adanya gap , v_1 , gap_1 , v_2 , gap_2 , v_3 , gap_3 , v_4 , dan gap_4 . Dimana gap adalah jumlah sel kosong secara berurutan tepat di depan sel v , gap mewakili jarak jalan kosong di depan kendaraan roda dua. Sel v_1 adalah sel yang berada tepat di sebelah kanan sel v , dan gap_1 adalah jumlah sel kosong secara berurutan berada tepat di depan sel v_1 . Sel v_2 adalah sel yang berstatus terisi kendaraan yang berada tepat di belakang sel v_1 , dan gap_2 adalah jumlah sel kosong secara berurutan di antara sel v_1 dan sel v_2 . Sedangkan v_3 , gap_3 , v_4 , dan gap_4 berturut-turut adalah sama dengan v_1 , gap_1 , v_2 , dan gap_2 tetapi terletak pada barisan sel yang berada tepat di sebelah kiri sel v . Faktor kedua yang dapat mempengaruhi pergerakan kendaraan roda dua adalah faktor pengemudi kendaraan yang dimodelkan dengan adanya probabilitas pengemudi kendaraan menurunkan kecepatannya (dilambangkan dengan $p.slow$), dan probabilitas pengemudi kendaraan mendahului kendaraan di depannya (dilambangkan dengan $p.move$).

Pergerakan kendaraan roda empat juga dipengaruhi oleh faktor keadaan jalan dan faktor pengemudi kendaraan. Misalkan $v4.kiri$ dan $v4.kanan$ adalah status sel yang terisi kendaraan roda empat. Sel dengan status $v4.kiri$ selalu berada tepat di sebelah kiri sel dengan status $v4.kanan$, dengan $v4.kiri = v4.kanan$ dan selanjutnya keduanya disebut dengan sel $v4$. Faktor keadaan jalan untuk sel $v4$ dimodelkan dengan adanya $gap4$, $v4_1$, $gap4_1$, $v4_2$, $gap4_2$, $v4_3$, $gap4_3$, $v4_4$, dan $gap4_4$. Jumlah sel kosong secara berurutan di depan sel $v4$ adalah $gap4$, dengan

$$gap4 = \min\{gap4.kiri, gap4.kanan\}. \quad (4.2)$$

Dimana $gap4.kiri$ adalah jumlah sel kosong berurutan yang berada tepat di depan sel $v4.kiri$, dan $gap4.kanan$ adalah jumlah sel kosong berurutan yang berada tepat di depan sel $v4.kanan$. Sel $v4_1$ adalah sel yang berada tepat di sebelah kanan sel $v4$, dan $gap4_1$ adalah jumlah sel kosong secara berurutan yang berada tepat di depan sel $v4_1$. Sel $v4_2$ adalah sel yang berstatus terisi kendaraan tepat berada di belakang sel $v4_1$, dan $gap4_2$ adalah jumlah sel kosong secara berurutan di antara sel $v4_1$ dan sel $v4_2$. Sedangkan $v4_3$, $gap4_3$, $v4_4$, dan $gap4_4$ berturut-turut adalah sama dengan $v4_1$, $gap4_1$, $v4_2$, dan $gap4_2$ tetapi terletak pada barisan sel yang berada tepat di sebelah kiri sel $v4$. Faktor pengemudi kendaraan untuk sel $v4$ dimodelkan dengan adanya probabilitas pengemudi menurunkan kecepatannya (dilambangkan dengan $p4.slow$), dan probabilitas pengemudi kendaraan mendahului kendaraan di depannya (dilambangkan dengan $p4.move$).

Sebagai ilustrasi, misalkan terdapat sel v yang memiliki status terisi kendaraan roda dua dan terdapat sel $v4$ yang memiliki status terisi kendaraan roda empat seperti Gambar 4.1. Sel v memiliki sel tetangga v_1, v_2, v_3 , dan sel v_4 . Sel v_1 memiliki status terisi kendaraan roda dua, dan sel v_2 juga memiliki status terisi kendaraan roda dua dan terletak di sebelah kiri sel v_1 . Sel v_3 memiliki status tidak terisi kendaraan, dan sel v_4 merupakan salah satu dari sepasang sel yang memiliki status terisi kendaraan roda empat. Sedangkan untuk sel $v4$ memiliki sel tetangga $v4_1, v4_2, v4_3$, dan sel $v4_4$. Sel $v4_1$ memiliki status tidak terisi kendaraan, dan sel $v4_2$ memiliki status terisi kendaraan roda dua, dimana sel $v4_2$ merupakan sel yang sama dengan sel v_2 . Kemudian, sel $v4_3$ dan sel $v4_4$ sama-sama memiliki status terisi kendaraan roda dua. Sehingga sekarang model *Cellular Automata* dari arus lalu lintas dapat diilustrasikan seperti Gambar 4.1, dengan sel berwarna abu-abu adalah gap dari masing-masing sel $v, v_1, v_2, v_3, v_4, v4, v4_1, v4_2, v4_3, v4_4$.



Gambar 4.1 Ilustrasi model arus lalu lintas dengan *Cellular Automata*

Suatu kendaraan akan bergerak lebih cepat jika jarak dengan kendaraan di depannya cukup jauh dan memungkinkan kendaraan tersebut menambah kecepatannya ($v < gap$ atau $v_4 < gap_4$). Namun jika kendaraan di depannya terlalu dekat ($v \geq gap$ atau $v_4 \geq gap_4$), kendaraan tersebut dapat berpindah lajur jika lajur di sebelahnya kosong, baik di depan maupun di belakang. Jika lajur di sebelahnya ternyata tidak kosong, maka kendaraan tersebut harus menurunkan kecepatannya hingga sama dengan gap .

Sesuai dengan penjelasan di atas dan karakteristik dari arus lalu lintas, maka berikut ini adalah empat langkah untuk memperbarui posisi kendaraan pada model arus lalu lintas dengan *Cellular Automata*:

1. **Pindah lajur:** Setiap kendaraan akan pindah lajur jika terpenuhi syarat-syarat berikut:
 - a. Kendaraan di depannya terlalu dekat,
 - untuk kendaraan roda dua:
$$v \geq gap, \quad (4.3)$$
 - untuk kendaraan roda empat:
$$v_4 \geq gap_4. \quad (4.4)$$
 - b. Lajur di sebelahnya kosong, baik di depan maupun di belakang,
 - untuk kendaraan roda dua:
$$v_1 = 100, v \leq gap_1 - 1 \text{ dan } gap_2 > v_2 \quad (4.5)$$
 - $$v_3 = 100, v \leq gap_3 - 1 \text{ dan } gap_4 > v_4, \quad (4.6)$$
 - untuk kendaraan roda empat:
$$v_4_1 = 100, v_4 \leq gap_4_1 - 1 \text{ dan } gap_4_2 > v_4_2 \quad (4.7)$$

$$v4_3 = 100, v4 \leq gap4_3 - 1 \text{ dan } gap4_4 > v4_4. \quad (4.8)$$

c. Pengemudi kendaraan ingin pindah lajur,

untuk kendaraan roda dua:

$$random() \leq p.move, \quad (4.9)$$

untuk kendaraan roda empat:

$$random() \leq p4.move. \quad (4.10)$$

2. **Percepatan** : Setiap kendaraan akan menaikkan atau menurunkan kecepatannya,

untuk kendaraan roda dua:

$$v := \min \{v + 1, gap, v_{max}\}, \quad (4.11)$$

untuk kendaraan roda empat:

$$v4 := \min \{v4 + 1, gap4, v_{max}\}. \quad (4.12)$$

3. **Perlambatan**: Kecepatan kendaraan yang lebih dari nol diturunkan satu jika memenuhi,

untuk kendaraan roda dua:

$$random() \leq p.slow, \quad (4.13)$$

untuk kendaraan roda empat:

$$random() \leq p4.slow. \quad (4.14)$$

4. **Perpindahan kendaraan**: Masing-masing kendaraan dipindahkan sejauh v sel ke depan.

4.1.2 Pembuatan Model *Cellular Automata*

Model arus lalu lintas dengan *Cellular Automata* dapat dituliskan dalam 5-tupel (L, Q, N, δ, C_0) , dimana:

L = Empat baris sel segi empat dua dimensi.

Q = $\{0, 1, 2, 3, 4, \dots, v_{max}, 100\}$, dimana:

100 = kosong,

$\{0, 1, 2, \dots, v_{max}\}$ = kecepatan kendaraan.

$N = \{\{sel(i, j) | i = \text{kiri, kanan dan } j = q - v_{max}, \dots, q - 2, q - 1, q + 1, q + 2, \dots, q + v_{max}\}, \{sel(p, j) | j = q + 1, q + 2, \dots, q + v_{max}\}, v_1, v_3, v4_1, v4_3\}$, dimana:
 $\{p, q\} = \text{indeks sel } v(p, q)$,
kiri = indeks sel tepat di sebelah kiri sel $v(p, q)$,
kanan = indeks sel tepat di sebelah kanan sel $v(p, q)$.

$\delta =$ Dilakukan dalam empat langkah. Setelah satu langkah selesai dilaksanakan, baru kemudian langkah berikutnya dapat dilaksanakan.

Langkah pertama:

- a. Setiap sel yang mempunyai status terisi kendaraan akan pindah ke baris sebelumnya, jika terpenuhi persamaan (4.3) – (4.10).
- b. Selain itu, sel tidak mengalami perubahan.

Langkah kedua:

- a. Jika status suatu sel tidak kosong ($\neq 100$) dan status sel tersebut lebih kecil dibanding *gap* (atau *gap4*), maka jika status sel tersebut masih belum maksimal ($< v_{max}$), status sel tersebut dinaikkan satu.
- b. Jika status suatu sel tidak kosong ($\neq 100$) dan status sel tersebut lebih besar dibanding *gap* (atau *gap4*), maka status sel tersebut diturunkan sehingga sama dengan *gap* (atau *gap4*).
- c. Selain itu, sel tidak mengalami perubahan.

Langkah ketiga:

- a. Dengan kemungkinan sebesar *p.slow* (atau *p4.slow*) status suatu sel akan berubah,

untuk kendaraan roda dua:

$$v := \max \{0, v - 1\}, \quad (4.15)$$

untuk kendaraan roda empat:

$$v4 := \max \{0, v4 - 1\}. \quad (4.16)$$

- b. Selain itu, sel tidak mengalami perubahan.

Langkah keempat:

- a. Jika status suatu sel lebih besar dari 0, maka status sel dipertukarkan dengan status sel ke- v tetangganya, dengan v adalah status sel tersebut.
- b. Selain itu, sel tidak mengalami perubahan.

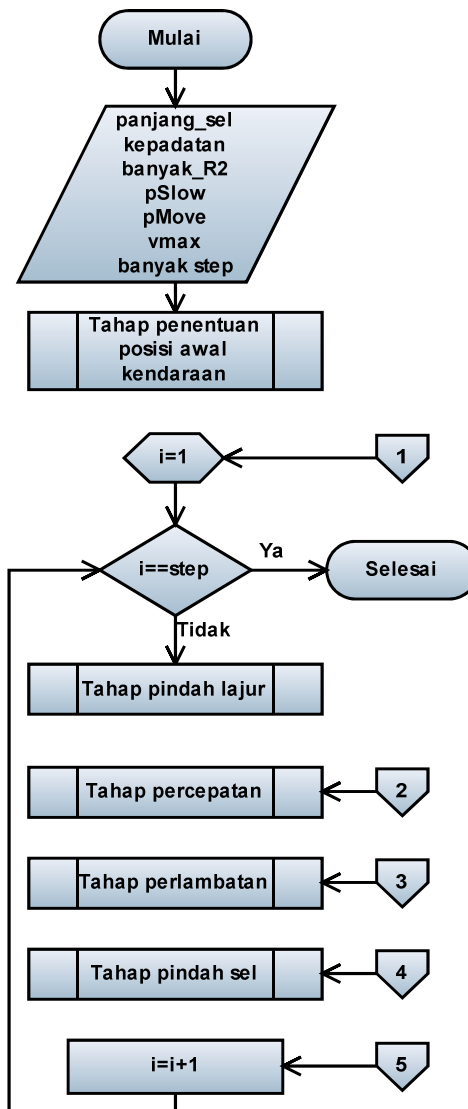
C_0 = Keadaan awal jalan.

Neighbourhood pada model ini sebanyak $n = (5 \times v_{max}) + 2$. Hasil ini merupakan jumlah sel pada himpunan N . Pada subhimpunan $\{sel(i, j) | i = \text{kiri, kanan dan } j = q - v_{max}, \dots, q - 2, q - 1, q + 1, q + 2, \dots, q + v_{max}\}$, terdapat sebanyak $4 \times v_{max}$ sel. Sedangkan pada subhimpunan $\{sel(p, j) | j = q + 1, q + 2, \dots, q + v_{max}\}$ terdapat sebanyak v_{max} sel. Sel v_1 adalah sel tetangga untuk sel yang berstatus terisi kendaraan roda dua, dan ekuivalen dengan sel $4v_1$ yang merupakan sel tetangga untuk sel yang berstatus terisi kendaraan roda empat. Begitu juga untuk sel v_3 yang ekuivalen dengan sel $4v_3$. Sehingga dari $\{v_1, v_3, 4v_1, 4v_3\}$ terdapat 2 sel. Dimana n merupakan jumlah *neighbourhood* maksimal yang dapat dimiliki oleh setiap sel. Sedangkan jari-jari ketetanggaannya adalah $r = v_{max}$

4.1.3 Pembuatan Algoritma dan Program

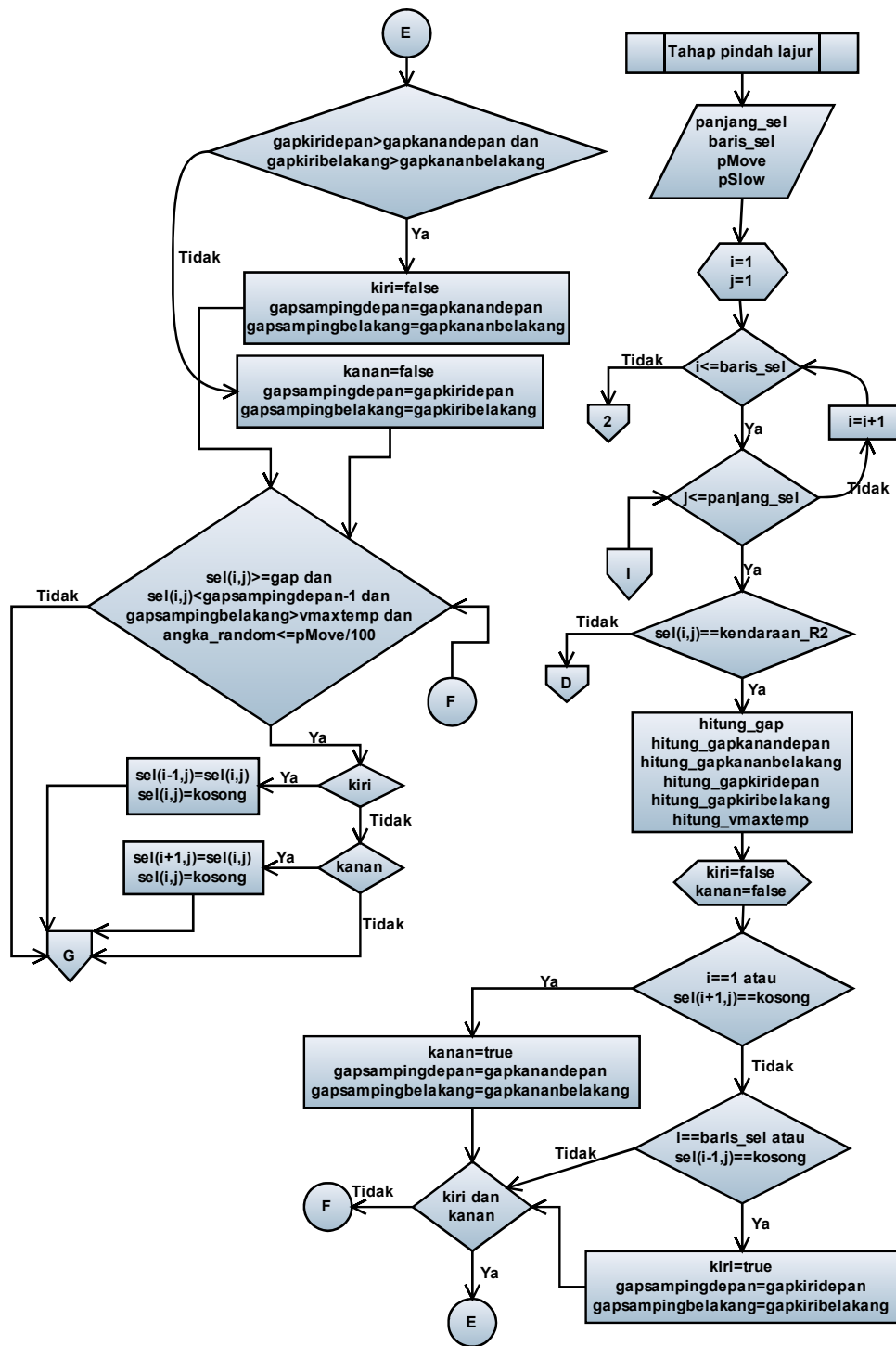
4.1.3.1 Algoritma

Sesuai dengan fungsi transisi yang diperoleh pada bagian sebelumnya, terdapat empat langkah untuk memperbarui status sebuah sel. Sehingga Algoritma model arus lalu lintas dengan *Cellular Automata* secara garis besar terdiri dari empat langkah utama yaitu tahap pindah lajur, tahap percepatan, tahap perlambatan, dan tahap pindah sel. Keempat tahapan tersebut diawali dengan tahap penentuan posisi awal kendaraan. Algoritma model arus lalu lintas dengan *Cellular Automata* disajikan pada Gambar 4.2 dalam bentuk *flowchart*.

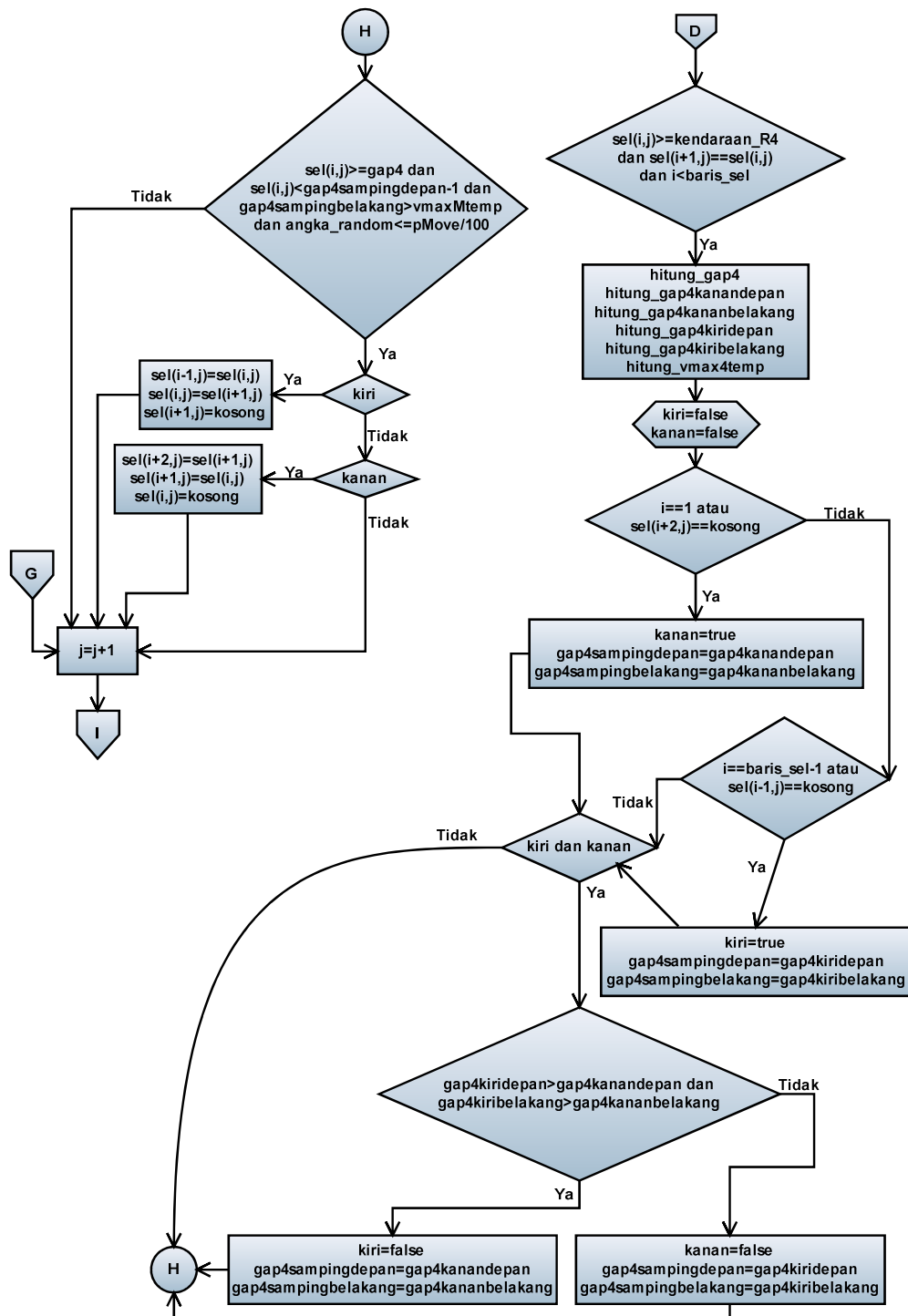


Gambar 4.2 Diagram alir model arus lalu lintas dengan *Cellular Automata*

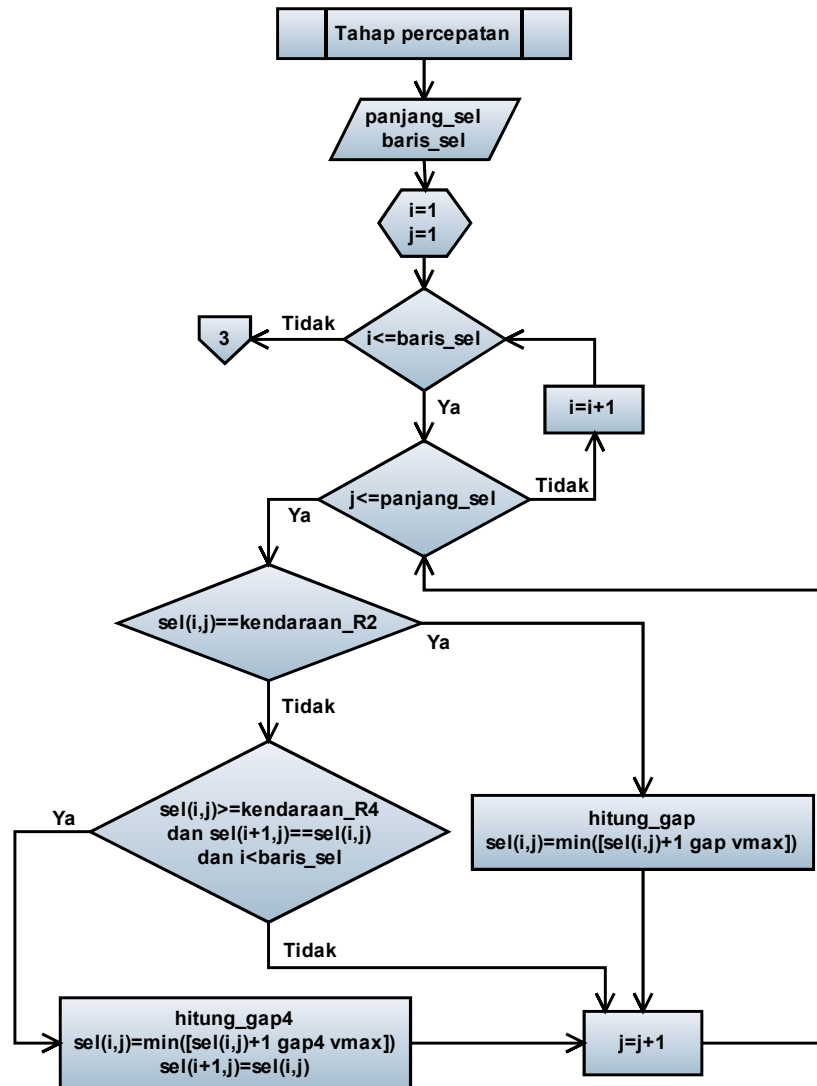
Pada *flowchart* tersebut terlihat bahwa algoritma model arus lalu lintas dengan *Cellular Automata* terdiri dari lima sub algoritma, yaitu sub algoritma tahap penentuan posisi awal kendaraan, sub algoritma tahap pindah lajur, sub algoritma tahap percepatan, sub algoritma tahap perlambatan, dan sub algoritma tahap pindah sel. *Flowchart* tahap penentuan posisi awal kendaraan disajikan pada Gambar 4.3. Pada Gambar 4.4 dan 4.5 merupakan *flowchart* dari tahap pindah lajur. Selanjutnya, Gambar 4.6 merupakan *flowchart* dari tahap percepatan, dan Gambar 4.7 adalah



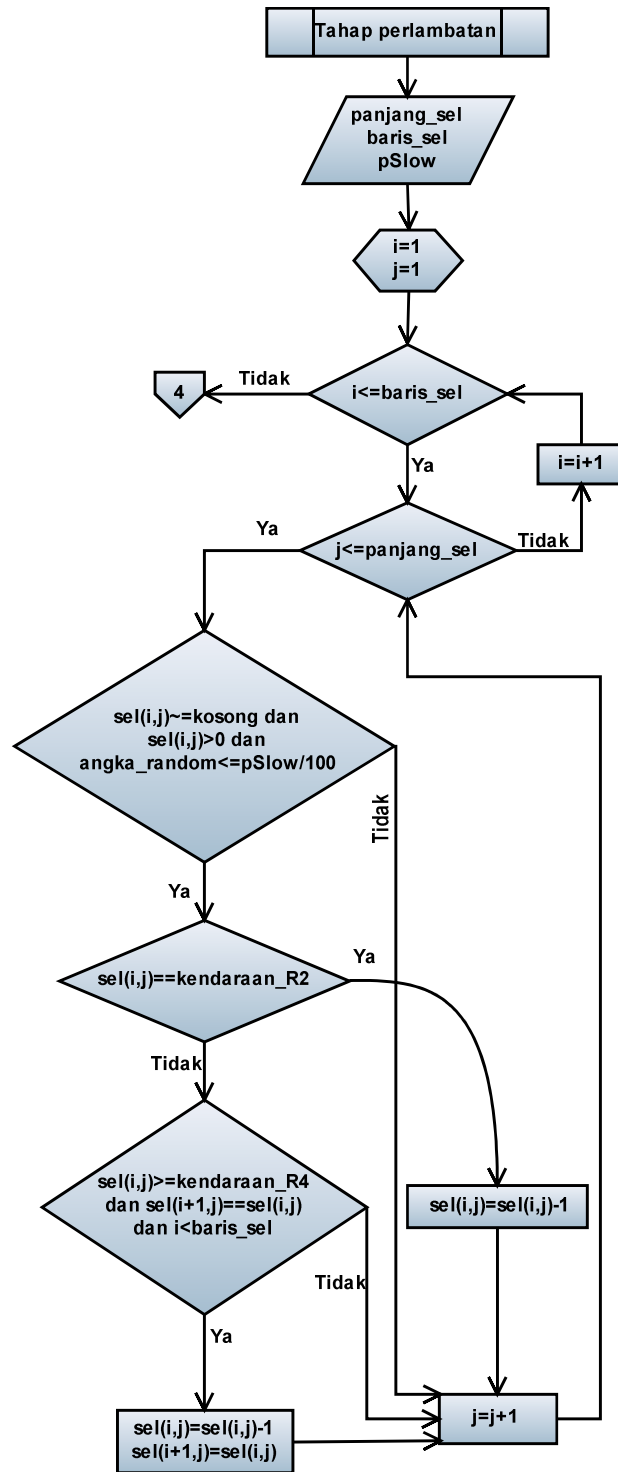
Gambar 4.4 Diagram alir tahap pindah lajur



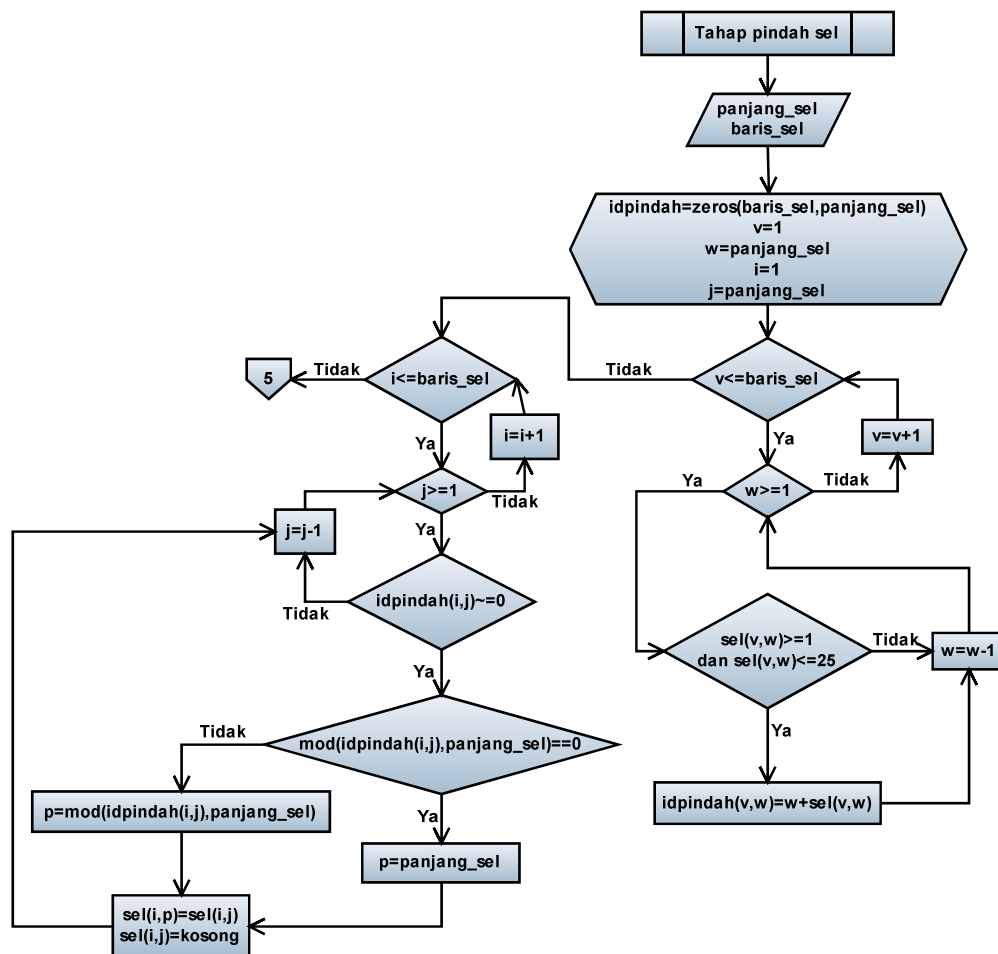
Gambar 4.5 Diagram alir tahap pindah lajur (lanjutan)



Gambar 4.6 Diagram alir tahap percepatan



Gambar 4.7 Diagram alir tahap perlambatan



Gambar 4.8 Diagram alir tahap pindah sel

a. Tahap Penentuan Posisi Awal Kendaraan

Posisi awal kendaraan ditentukan dengan membangkitkannya secara random. Jumlah kendaraan ditentukan dengan mempertimbangkan kepadatan arus lalu lintas. Dari jumlah tersebut dibagi menjadi dua jenis kendaraan yaitu kendaraan roda dua dan roda empat. Jumlah kendaraan roda dua diperoleh dengan mengalikan jumlah kendaraan dengan persentase kendaraan roda dua, begitu juga dengan kendaraan roda empat. Algoritma untuk tahap ini disajikan pada Gambar 4.3. Pada tahap penentuan posisi awal kendaraan setiap sel akan diperiksa, dimulai dari $sel(1,1)$ hingga $sel(baris_sel,panjang_sel)$. Pada setiap sel akan

dibangkitkan angka random antara 0 hingga 1. Angka random tersebut akan dibandingkan dengan persentase kepadatan arus lalu lintas. Jika nilai angka random yang dibangkitkan kurang dari sama dengan kepadatan arus lalu lintasnya, maka sel tersebut akan diisi oleh kendaraan. Kemudian dibangkitkan angka random kembali yang akan dibandingkan dengan persentase kepadatan kendaraan roda dua. Jika angka random yang dibangkitkan kurang dari sama dengan persentase kepadatan kendaraan roda dua, maka sel tersebut akan diisi oleh kendaraan roda dua. Tetapi, jika angka random yang dibangkitkan lebih besar dari persentase kepadatan kendaraan roda dua dan status sel yang berada tepat di sebelah kiri atau kanannya kosong, maka sel tersebut akan diisi kendaraan empat dan status sel kosong yang berada tepat di sebelah kiri atau kanannya akan digantikan dengan status sel tersebut.

b. Tahap Pindah Lajur

Setiap kendaraan yang memiliki kecepatan lebih tinggi daripada kendaraan di depannya dapat berpindah ke lajur sebelahnya jika lajur di sebelahnya aman. Algoritma tahap ini dapat dilihat pada Gambar 4.4 dan 4.5. Pada setiap sel, akan diperiksa status dari sel tersebut. Jika sel yang diperiksa memiliki status terisi kendaraan roda dua, sebut sel tersebut sebagai sel v , maka akan dihitung gap , gap_1 , gap_2 , gap_3 , gap_4 dan $vmax.temp$. Dimana $vmax.temp$ adalah status terbesar dari sel-sel yang berada di belakang sel v . Kemudian status sel tetangga dari sel v akan diperiksa yaitu sel v_1 , v_2 , v_3 , dan v_4 seperti telah dijelaskan pada subsubbab 4.1.1. Jika status dari sel v_1 adalah kosong, maka status sel v dapat dipindah ke sel v_1 dan

$$gap.samping.depan = gap_1 \quad (4.17)$$

$$gap.samping.belakang = gap_2. \quad (4.18)$$

Jika sel v_3 yang memiliki status kosong, maka status sel v dapat dipindah ke posisi sel v_3 dan

$$gap.samping.depan = gap_3 \quad (4.19)$$

$$gap.samping.belakang = gap_4. \quad (4.20)$$

Tetapi, jika status sel v_1 dan sel v_3 sama-sama kosong dan $gap_3 > gap_1$ dan $gap_4 > gap_2$, maka status sel v akan dipindah ke posisi sel v_3 dan menerapkan persamaan (4.19) dan (4.20). Kemudian dibangkitkan bilangan random antara 0 dan 1. Jika bilangan random yang dibangkitkan kurang dari sama dengan probabilitas pindah lajur, $v \geq gap$,

$$v < gap.samping.depan - 1, \text{ dan} \quad (4.21)$$

$$gap.samping.belakang > vmax.temp, \quad (4.22)$$

maka sel v dipindah ke sel di sebelahnya (sebelah kiri atau kanan seperti telah ditentukan sebelumnya). Sedangkan jika sel yang diperiksa memiliki status terisi kendaraan roda empat, maka langkah-langkah yang dilakukan sama dengan langkah-langkah yang sama dengan sel yang terisi kendaraan roda dua.

c. Tahap Percepatan

Pada tahap ini, jika kecepatan setiap kendaraan yang belum mencapai kecepatan maksimal dan jarak dengan kendaraan di depannya cukup jauh, maka akan di naikkan satu. Algoritma pada tahap ini adalah seperti pada Gambar 4.6. Pada tahap percepatan, setiap sel akan diperiksa satu persatu. Jika sel yang diperiksa memiliki status terisi kendaran roda dua, maka dihitung gap dari sel tersebut dan kemudian status sel dirubah sesuai dengan persamaan (4.11). Jika sel yang diperiksa memiliki status terisi kendaraan roda empat, maka dihitung gap_4 dari sel tersebut dan kemudian status sel dirubah sesuai dengan persamaan (4.12).

d. Tahap Perlambatan

Pada arus lalu lintas sebenarnya, setiap kendaraan dapat mempertahankan atau menurunkan kecepatannya meskipun memungkinkan untuk menambah kecepatannya. Hal ini dipengaruhi oleh faktor pengemudi, yaitu terdapat kemungkinan sebesar $p.Slow$ bagi pengemudi untuk menurunkan kecepatan kendaraannya. Algoritma tahap ini adalah seperti pada Gambar 4.7. Pada tahap ini dibangkitkan bilangan random antara 0 dan 1. Kemudian, jika bilangan tersebut

kurang dari sama dengan $p \cdot Slow / 100$, maka status sel berubah sesuai dengan persamaan (4.15) jika status sel terisi kendaraan roda dua atau status sel berubah sesuai dengan persamaan (4.16) jika status sel terisi kendaraan roda empat.

e. Tahap Pindah Sel

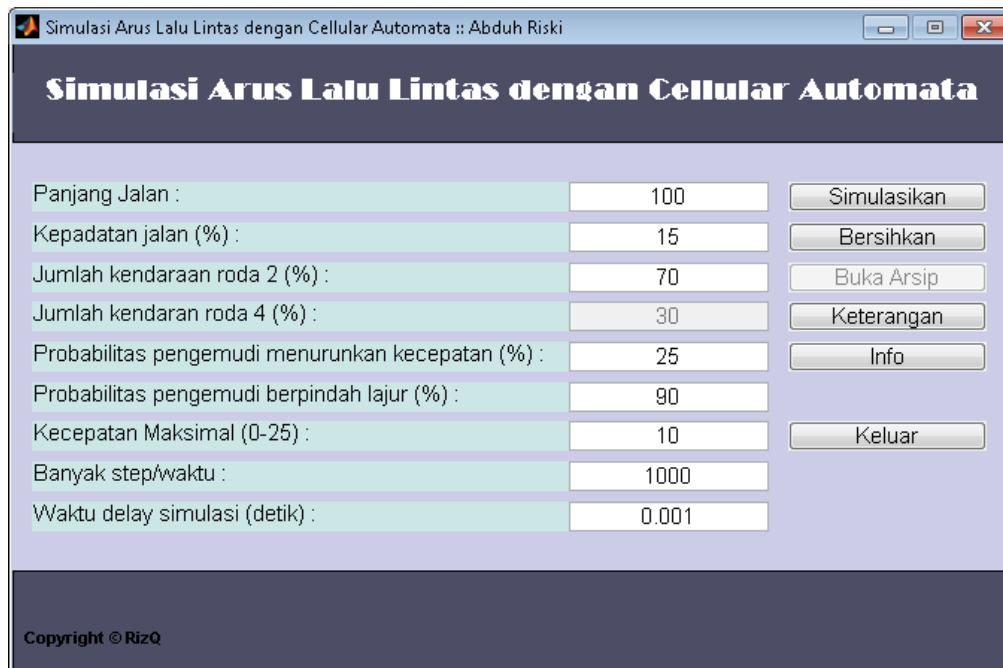
Setelah melalui tahap-tahap sebelumnya, maka sekarang status sel telah berubah menjadi status sel yang baru. Sebelum mengulangi tahap pindah lajur kembali, status sel akan di pindah ke depan sejauh status sel yang didapat sekarang. Dalam kenyataannya, hal ini berarti posisi kendaraan diperbarui dengan menggerakkannya ke depan. Algoritma untuk tahap ini dapat dilihat pada Gambar 4.8. Dimana $zeros(baris_sel, panjang_sel)$ adalah matriks nol dengan ukuran matriks tersebut adalah $baris_sel \times panjang_sel$. Sedangkan $mod(idpindah(i, j), panjang_sel)$ adalah operasi modulus, yaitu operasi yang menghasilkan sisa pembagian dari suatu bilangan terhadap bilangan lainnya. Jadi $mod(idpindah(i, j), panjang_sel)$ adalah bilangan bulat sisa pembagian $idpindah(i, j)$ oleh $panjang_sel$.

4.1.3.2 Pembuatan Program

Program simulasi model arus lalu lintas dengan *Cellular Automata* dibuat dengan *software Matlab 7.8.0*. Matlab adalah bahasa pemrograman level tinggi yang dikhususkan untuk komputasi teknis. Bahasa ini mengintegrasikan kemampuan komputasi, visualisasi, dan pemrograman dalam sebuah lingkungan yang tunggal dan mudah digunakan. *Matlab* memberikan sistem interaktif yang menggunakan konsep *array*/matriks sebagai standar variabel elemennya tanpa membutuhkan pendeklarasian *array* seperti bahasa lainnya.

Program simulasi model arus lalu lintas dengan *Cellular Automata* tersebut diberi nama program RodaDua. Program dibuat mengikuti algoritma sebelumnya, sehingga menghasilkan sebuah fungsi *Matlab* untuk setiap tahapnya. Semua fungsi yang dibuat kemudian dipanggil melalui fungsi utama, yaitu fungsi GUI (*Graphic*

User Interface) dari program RodaDua. Tampilan program RodaDua adalah seperti pada Gambar 4.9.



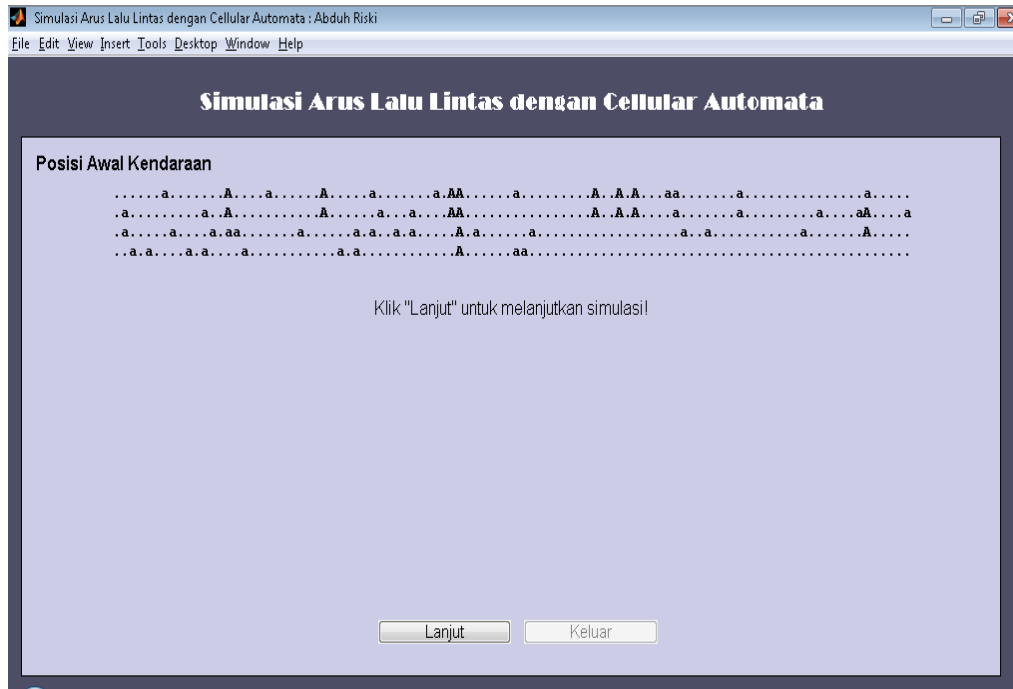
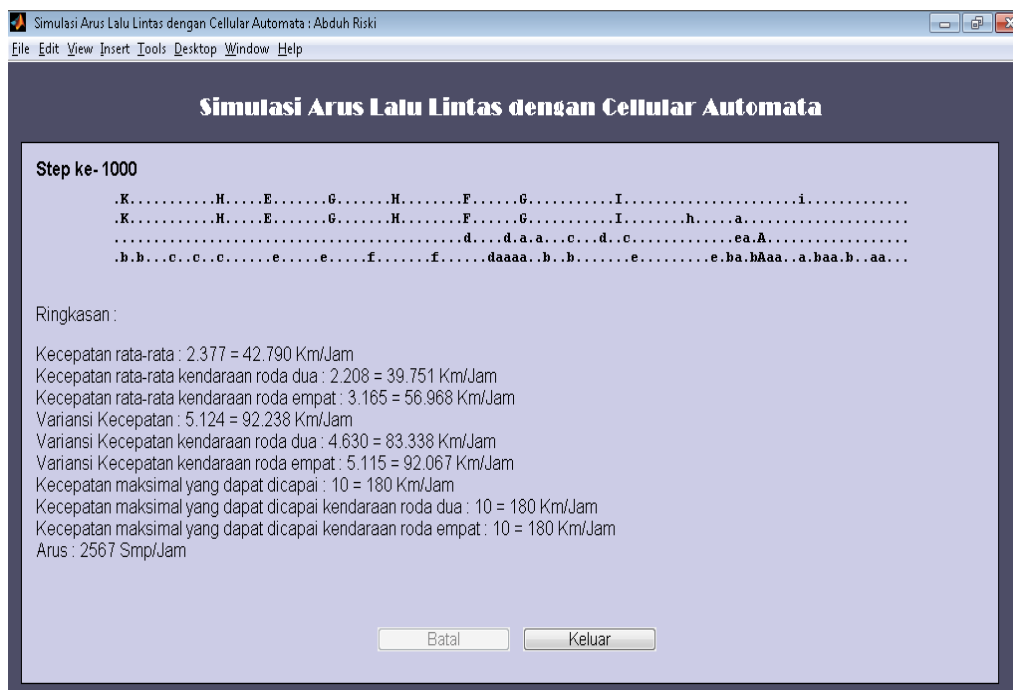
Gambar 4.9 GUI program RodaDua

Terdapat sembilan *input* pada program RodaDua seperti pada Gambar 4.9, dengan *input* jumlah kendaraan roda 4 akan terisi otomatis. Panjang jalan adalah panjang sel *Cellular Automata* yang akan digunakan dalam simulasi. Kepadatan jalan adalah kepadatan pada arus lalu lintas yang akan disimulasikan, dan dimasukkan dalam satuan persen (%). Jumlah kendaraan roda 2 dan jumlah kendaraan roda 4 adalah persentase jumlah masing-masing jenis kendaraan dalam arus lalu lintas dalam satuan persen. Persentase jumlah kendaraan roda 4 sama dengan 100% dikurangi persentase jumlah kendaraan roda 2. Jadi, untuk *input* jumlah kendaraan roda 4 otomatis akan terisi setelah *input* untuk jumlah kendaraan roda 2 diberikan. Probabilitas pengemudi menurunkan kecepatan adalah *input* yang nantinya digunakan pada tahap perlambatan, *input* ini dimasukkan dalam satuan persen. Sedangkan probabilitas pengemudi berpindah lajur adalah *input* yang akan digunakan pada tahap

pindah lajur, *input* ini juga dimasukkan dalam satuan persen. Input kecepatan maksimal adalah kecepatan maksimal yang diperbolehkan dalam simulasi yang dilakukan, *input* ini berupa bilangan bulat non-negatif kurang dari sama dengan 25. Banyak *step*/waktu adalah banyak *time-step* yang akan dilakukan dalam simulasi. *Input* yang terakhir adalah waktu *delay* simulasi dalam satuan detik yang berfungsi untuk mengatur jeda waktu antar setiap *time-step* dalam simulasi.

Pada GUI program RodaDua terdapat enam tombol yang memiliki fungsi-fungsi tersendiri. Tombol simulasi berfungsi untuk menjalankan simulasi jika semua *input* telah dimasukkan. Tombol bersihkan berfungsi untuk mengosongkan *textbox input* yang telah terisi sebelumnya. Tombol buka arsip berfungsi untuk membuka file berekstensi *.doc* yang berisi semua proses simulasi yang telah dilakukan pada simulasi terakhir kali. Tombol ini akan aktif jika telah melakukan simulasi. Tombol keterangan berfungsi untuk membuka GUI yang berisi keterangan dari simbol-simbol yang digunakan dalam simulasi. Tombol info berfungsi untuk membuka GUI yang berisi informasi tentang program RodaDua. Sedangkan tombol tutup berfungsi untuk menutup program RodaDua.

Misalkan telah diberikan semua *input* yang diperlukan seperti pada Gambar 4.9, maka setelah menekan tombol simulasi kemudian akan muncul GUI *output* program RodaDua seperti pada gambar 4.10. GUI tersebut berisi tentang posisi awal kendaraan pada simulasi yang divisualisasikan dengan huruf dan tanda titik. Serta berisi dua buah tombol, tombol lanjut untuk melanjutkan proses simulasi dan tombol keluar yang berfungsi untuk menghentikan proses simulasi. Tetapi tombol keluar dalam keadaan tidak aktif, dan akan aktif jika tombol lanjut telah ditekan. Setelah semua proses simulasi selesai, maka tampilan terakhir dari GUI *output* program RodaDua adalah seperti pada Gambar 4.11.

Gambar 4.10 GUI *output* program RodaDuaGambar 4.11 GUI *output* akhir program RodaDua

4.2 Pembahasan

Sebagai ilustrasi dari perubahan sel pada simulasi arus lalu lintas dengan *Cellular Automata*, maka akan dilakukan simulasi dengan *input* sebagai berikut:

panjang jalan = 75 sel

kepadatan jalan = 15 %

jumlah kendaraan roda dua = 70 %

jumlah kendaraan roda empat = 30 %

probabilitas pengemudi menurunkan kecepatan = 15 %

probabilitas pengemudi berpindah lajur = 90 %

kecepatan maksimal = 20 sel

banyak step/waktu = 9 *time-step*

waktu *delay* simulasi = 0.01 detik.

Setelah semua *input* tersebut dimasukkan ke dalam program RodaDua kemudian simulasi dijalankan, maka perubahan sel untuk setiap *time-step* diperlihatkan pada Gambar 4.12. Gambar 4.12 adalah gambar kondisi jalan yang disimulasikan untuk setiap *time-step*. Satu baris melambangkan kondisi jalan pada satu lajur jalan, dan setiap empat baris melambangkan kondisi jalan pada satu ruas jalan. Tanda titik berarti bagian jalan tersebut tidak ditempati oleh kendaraan. Tanda huruf berarti bagian jalan tersebut sedang ditempati oleh kendaraan. Huruf kapital melambangkan kendaraan roda empat dan huruf kecil melambangkan kendaraan roda dua. Sedangkan tingkatan dari huruf melambangkan kecepatan dari kendaraan tersebut (lihat Tabel 4.1). Arah perpindahan kendaraan adalah ke arah kanan. Pada Gambar 4.12, semua status sel pada saat *step* ke-0 adalah 0 yang dilambangkan dengan huruf “a” atau “A”. Hal ini terjadi karena pada saat tersebut setiap kendaraan dalam keadaan berhenti dan siap untuk bergerak untuk *step* selanjutnya. Posisi-posisi setiap kendaraan pada *step* ke-0 di tentukan pada tahap penentuan posisi awal kendaraan. Selanjutnya, setelah *step* ke-0 status setiap sel akan mengalami dinamika baik itu bertambah, berkurang, atau tetap.

```

Posisi Awal Kendaraan/step ke- 0
....A...a.....a..a...Aa.....A.a..A.a...a...A.....
...aA.....A...a.a...AA...a.A..aA..a.....A..a.....a...
.....a..A.....a..a..Aa.....a.....a.....a.....a.....a
..aa..a..a.....a.....a.....aa.....

step ke- 1
....A...b.....b..b..A.b.....B.b..B.b...b...B.....
...aA.....A...b.b..AA...b.B..b.B..b.....B.a.....b...
.a.....b..A.....b..b.Aa.....b.....b.....b.....b.....a
..a...b..b.....b.....b.....a.b.....

step ke- 2
.....B.....c.....c..cA...c.....B..cA...c.....c..B.....
...a.B.....B.....a..c.AA...b.B.a.A...c.....Ba.....c..
b.b.....b..B.....c.bA.b.....b.....b.....c.....c.....c.....
..b...c..c.....c.....c...b.b.....

step ke- 3
.....C.....d.....caA...c...A..a.B...d...cA.....
.d..b..C.....C...b...bA.B...a.A..b.B...d...A.b.....
a..c.....b..C.....ba.B..c.....b.....d...d...d.....
....c...c..d.....c.....d..b..c.....

step ke- 4
.....d.....D.....e...aaA.....d.B..b..C.....c.a.B.....
.....c...D.....D..c.aA...C...a..B..b..C.....d.B..c.....
..b...c.....c...D...a.b..C..d.....c.....e...d.....e...
.....d...d...e.....d.....c..c...d.....

step ke- 5
.....e.....E.....c.aa.B.....a..C..c..D..a..b..C.....
.e.....d...E.....C.ba.B...Da...C..c..D.....C..d.....
..b.....d.....d...C..b..c..D...e.....d.....e.....e.....
.....e.....e.....f.....e.....c..d...e.....

step ke- 6
.....f.....Fa.a.b..C.....b...C..d..C.b..c..D.....
.....f.....e.....F.Ba.b..C...A.b...C..d..C.....D...e.....
..b.....e...c..B...c.c.A.....f...e.....f...d...f...
.....d.....f.....f...e...c.....e.....f.....

step ke- 7
.....fA.b.b..c...D...c.....D..c.B..c...c...D.....
..f.....g...e.A.A.b..c...D.B..c...D..c.B.....D...f.....
....c.....e...cA.....b.a..B.....f...f.....f...g...e.....
.....e.....e.....f...e...e...d.....f...g.....

step ke- 8
.....a.B.b..c...d.B.....d..B..b..C..d..d...E.....
.....g.....e.b.BA...c...d.B..C..d...B..b..C.....E.....g...
.....d.....caA.....b.b..C.....f...g.....e...f...
.....f.....e...e...d...e.....g.....h.....

step ke- 9
.....a.A...c...da...C.....d..C..c..D..d...e...F.....
..g.....h.b.bA.B...d.b..C..D...e..C..c..D.....F.....
...f.....e...a..B...b..c...D.....g...h...f...
.....b.....g...e..d...e...f.....h.....i...

```

Gambar 4.12 Perubahan sel model arus lalu lintas dengan *Cellular Automata*

Gambar 4.13 merupakan sebagian hasil simulasi sebelumnya pada *step* ke-3 hingga *step* ke-5. Pada simulasi, sel yang berstatus 0 dan dilambangkan dengan “a” atau “A” dapat berarti kendaraan tersebut berhenti atau pengemudi kendaraan tersebut hanya menurunkan kecepatan kendaraanya tetapi tidak berhenti.

- Sebuah kendaraan dikatakan berhenti jika status dari sel pada simulasi yang mewakili sebuah kendaraan adalah selalu 0 pada lebih dari satu *step* secara berturut-turut. Seperti sel (b,1) pada *step* ke-3, sel (b,1) pada *step* ke-4, dan sel (b,1) pada *step* ke-5 yang ketiganya berstatus 0 (Lihat Gambar 4.13).
- Sebuah kendaraan dikatakan hanya berkurang kecepatannya dan tidak sampai berhenti jika status dari sel pada simulasi yang mewakili sebuah kendaraan adalah 0 hanya pada satu *step*. Seperti sel (a,18) pada *step* ke-3, sel (a,18) pada *step* ke-4, dan sel (a,19) pada *step* ke-5. Dimana hanya sel (a,18) pada *step* ke-4 yang berstatus 0 (Lihat Gambar 4.13).

| step ke- 3 | | | | | | | | | | | | | | | | | | | | |
|------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| a | . | . | A | . | . | a | . | B | . | . | . | . | . | d | . | . | . | c | A | . |
| b | a | . | A | . | . | b | . | B | . | . | . | . | . | . | d | . | . | . | A | . |
| c | . | . | . | . | . | . | b | . | . | . | . | . | . | . | . | . | . | . | . | . |
| d | d | . | . | b | . | . | c | . | . | . | . | . | . | . | . | . | . | . | . | . |

| step ke- 4 | | | | | | | | | | | | | | | | | | | | |
|------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| a | . | d | . | B | . | . | b | . | . | C | . | . | . | . | . | c | . | a | . | B |
| b | a | . | . | B | . | . | b | . | . | C | . | . | . | . | . | . | . | d | . | B |
| c | d | . | . | . | . | . | . | . | c | . | . | . | . | . | . | . | . | . | . | . |
| d | . | . | c | . | . | c | . | . | . | d | . | . | . | . | . | . | . | . | . | . |

| step ke- 5 | | | | | | | | | | | | | | | | | | | | |
|------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| a | . | a | . | . | . | C | . | . | c | . | . | . | D | . | . | a | . | . | b | . |
| b | a | . | . | . | . | C | . | . | c | . | . | . | D | . | . | . | . | . | . | . |
| c | . | . | . | . | e | . | . | . | . | . | . | d | . | . | . | . | . | . | . | . |
| d | . | . | . | . | c | . | . | . | d | . | . | . | . | e | . | . | . | . | . | . |

Gambar 4.13 Contoh kasus kendaraan berkurang kecepatannya dan berhenti

Jadi sel berstatus 0 pada simulasi masih merepresentasikan hal yang wajar pada kenyataannya. Meskipun terkadang untuk kasus kendaraan berhenti tidak dapat dijelaskan alasan dari kendaraan tersebut berhenti. Tetapi pada umumnya kendaraan berhenti pada saat terjadi kemacetan, dan pada simulasi ditunjukkan pada sel yang

selalu berstatus 0 pada lebih dari satu *step* berturut-turut karena tepat di depannya terdapat sel lain yang bersatus terisi kendaraan.

Pada Gambar 4.14 dapat dilihat terjadinya kendaraan berpindah lajur untuk mendahului kendaraan di depannya. Terlihat bahwa kendaraan di sel (b,5) pada *step* ke-4 berpindah lajur ke lajur di sebelah kanannya, yaitu berpindah ke sel (c,9) pada *step* ke-5. Kasus ini terjadi pada tahap pindah lajur. Terjadi karena status sel (b,5) pada *step* ke-4 lebih besar dari *gap* sel tersebut yaitu sebesar 1 sel, sel di sebelah kanannya berstatus kosong baik depan dan di belakangnya, dan tentunya juga karena memenuhi unsur probabilistik yaitu bilangan random yang dibangkitkan kurang dari probabilitas pengemudi berpindah lajur.

| step ke- 4 | | | | | | | | | | | | | | | |
|------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| a | . | . | c | . | a | . | B | . | . | . | . | . | . | . | . |
| b | . | . | . | . | d | . | B | . | . | c | . | . | . | . | . |
| c | . | . | . | . | . | . | . | . | . | . | . | . | . | e | . |
| d | . | . | . | . | . | . | . | . | . | . | . | . | . | . | . |

| step ke- 5 | | | | | | | | | | | | | | | |
|------------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| a | . | . | a | . | . | b | . | . | C | . | . | . | . | . | . |
| b | . | . | . | . | . | . | . | . | C | . | . | . | d | . | . |
| c | . | . | . | . | . | . | . | . | e | . | . | . | . | . | . |
| d | e | . | . | . | . | . | . | . | . | . | . | . | . | . | . |

Gambar 4.14 Contoh kasus tahap pindah lajur

Kemudian, pada Gambar 4.15 merupakan contoh kasus pada tahap perlambatan. Terlihat pada Gambar 4.15 (a) bahwa status sel (c,2) pada *step* ke-4 adalah 1. Tetapi setelah tahap perpindahan sel, sel tersebut berpindah ke sel (c,3) pada *step* ke-5 dan statusnya tetap 1 yang dilambangkan dengan “b”. Padahal *gap* di depan sel tersebut lebih besar dari status sel tersebut yaitu $v < gap$, dan status sel tersebut juga kurang dari kecepatan maksimal yang telah ditentukan yaitu sebesar 20 sel yang dilambangkan dengan abjad “u”. Oleh karena itu, tentunya status sel tersebut pada tahap percepatan akan dinaikkan satu. Tetapi sel tersebut tetap berstatus 1. Hal ini terjadi karena pada tahap perlambatan bilangan random yang dibangkitkan kurang dari probabilitas pengemudi menurunkan kecepatannya. Hal serupa juga terjadi pada

sel (b,4) *step* ke-4 dan sel (b,4) *step* ke-5 pada gambar 4.15 (b). Sel (b,4) pada *step* ke-4 berstatus 3 dilambangkan dengan “d” dan pada saat tersebut *gap* di depannya terlalu kecil dan sel yang berstatus terisi kendaraan tepat di depannya memiliki status yang lebih kecil. Sehingga syarat-syarat dalam tahap percepatan terpenuhi dan status sel tersebut akan berubah menjadi sama dengan *gap* yaitu 1 yang dilambangkan dengan “b”. Tetapi pada tahap perlambatan, status dari sel tersebut kembali dikurangi. Hal ini terjadi karena bilangan random yang dibangkitkan kurang dari probabilitas pengemudi menurunkan kecepatannya. Sehingga status sel tersebut yaitu sel (b,4) pada *step* ke-5 menjadi 0 dan dilambangkan dengan “a”, tetapi pada *step* ke-6 (lihat Gambar 4.12) status sel tersebut kembali bertambah. Jadi pada kenyataannya hal ini berarti pengemudi kendaraan berkeinginan mengurangi kecepatan kendaraannya tetapi tidak sampai berhenti total, meskipun tidak terjadi kondisi yang mengharuskan pengemudi untuk mengurangi kecepatannya.

| step ke- 4 | | | | | | | | | |
|------------|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| a | . | . | . | . | d | . | . | . | . |
| b | . | . | . | . | . | . | c | . | . |
| c | . | b | . | . | . | . | c | . | . |
| d | . | . | . | . | . | . | . | . | d |

| step ke- 5 | | | | | | | | | |
|------------|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| a | . | . | . | . | . | . | . | . | e |
| b | . | e | . | . | . | . | . | . | . |
| c | . | . | b | . | . | . | . | . | . |
| d | . | . | . | . | . | . | . | . | . |

| step ke- 4 | | | | | | | | | |
|------------|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| a | . | . | . | . | d | . | B | . | . |
| b | . | . | . | a | . | . | B | . | . |
| c | . | . | . | d | . | . | . | . | . |
| d | . | . | . | . | . | c | . | . | C |

| step ke- 5 | | | | | | | | | |
|------------|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| a | . | . | . | . | a | . | . | . | C |
| b | . | . | D | a | . | . | . | . | C |
| c | . | . | D | . | . | . | . | e | . |
| d | e | . | . | . | . | . | . | c | . |

(a)

(b)

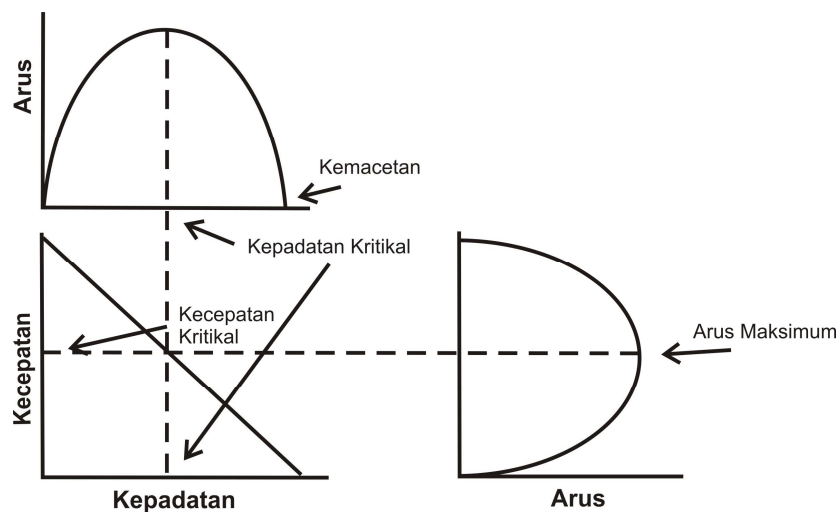
(a) Tahap perlambatan setelah status sel bertambah;

(b) Tahap perlambatan setelah status sel berkurang

Gambar 4.15 Contoh kasus tahap perlambatan

Simulasi kedua dilakukan untuk melihat relasi dari ketiga parameter arus lalu lintas. Dari ketiga parameter arus lalu lintas, terbentuk relasi antara ketiganya yang dapat digunakan untuk merepresentasikan arus lalu lintas. Ketiga parameter tersebut berelasi sesuai dengan persamaan (2.4). Dalam literatur dijelaskan bahwa relasi antara arus dan kepadatan, kepadatan dan kecepatan, kecepatan dan arus dapat

direpresentasikan dalam bentuk grafik. Grafik tersebut disebut sebagai “Diagram Fundamental” dari arus lalu lintas seperti pada Gambar 4.16 (Mathew dan Rao, 2006). Diagram fundamental adalah alat sederhana yang dapat digunakan untuk memahami karakteristik arus lalu lintas. Grafik tersebut memiliki bentuk karakteristik yang sama untuk setiap bagian jalan pada arus lalu lintas. Diagram fundamental pertama kali diperkenalkan oleh Greenshields pada tahun 1935 (Bellemans *et al.*, 2002).

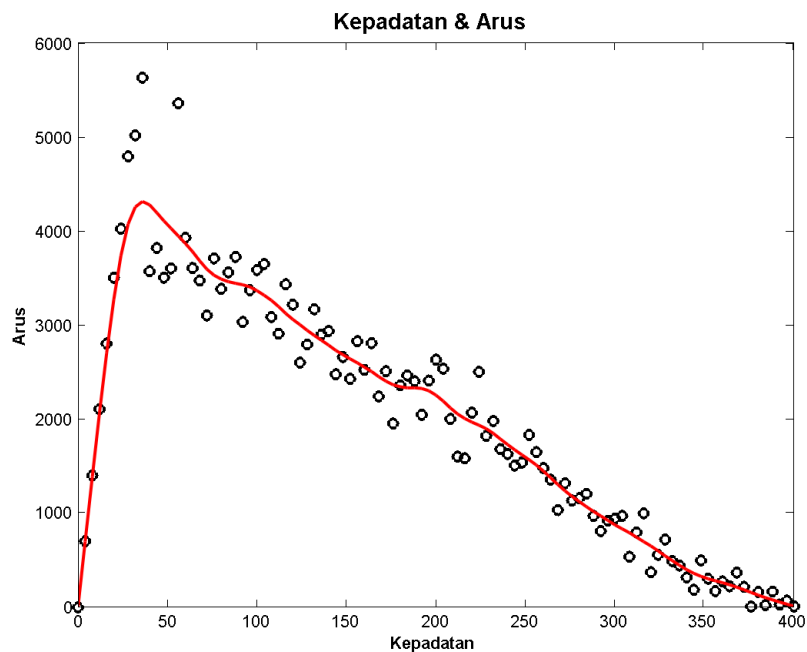


Gambar 4.16 Diagram fundamental arus lalu lintas

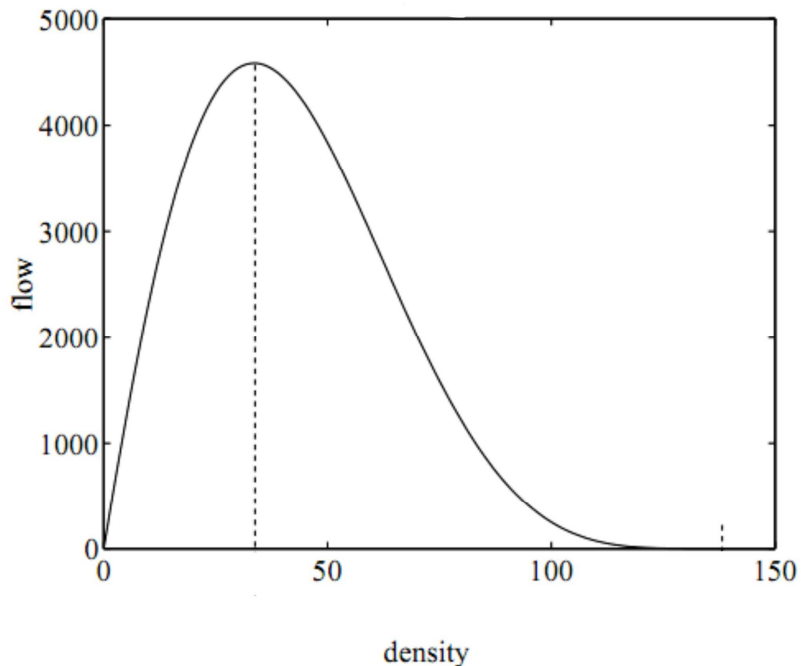
Simulasi kedua dilakukan dengan memvariasikan kepadatan, dan kepadatan yang digunakan pada simulasi ini adalah bilangan asli kurang dari sama dengan 100. Sedangkan *input* yang lainnya ditetapkan sebagai berikut:

- panjang jalan = 100 sel
- jumlah kendaraan roda dua = 60 %
- jumlah kendaraan roda empat = 40 %
- probabilitas pengemudi menurunkan kecepatan = 20 %
- probabilitas pengemudi berpindah lajur = 90 %
- kecepatan maksimal = 10 sel
- banyak *step*/waktu = 1000 *time-step*.

Gambar 4.17 menunjukkan grafik relasi arus – kepadatan dari model arus lalu lintas dengan *Cellular Automata*. Secara umum sesuai dengan diagram fundamental pada Gambar 4.16 , grafik relasi arus – kepadatan akan menyerupai parabola terbuka ke bawah. Sedangkan dari pemodelan arus lalu lintas oleh Bellemans *et al.* (2002), relasi arus – kepadatan akan menghasilkan grafik seperti pada Gambar 4.18. Dari grafik relasi arus – kepadatan Gambar 4.17 terlihat bahwa untuk kepadatan nol (tidak ada kendaraan di jalan), tentu saja arus juga nol. Dengan meningkatnya kepadatan (jumlah kendaraan bertambah di jalan), arus juga akan meningkat. Jika kepadatan meningkat lebih lanjut, peningkatan arus akan melambat. Pada kepadatan tertentu, akan tercapai arus maksimum. Kepadatan tersebut disebut kepadatan kritis ρ_c . Pada simulasi model arus lalu lintas dengan *Cellular Automata*, terjadi arus maksimum sebesar 5625 smp/jam pada kepadatan 0,09 atau 36 smp/km. Arus akan menurun dengan meningkatnya kepadatan jika kepadatannya lebih besar dari kepadatan kritis. Akhirnya arus menjadi nol kembali pada kepadatan maksimal. Kepadatan maksimal pada simulasi arus lalu lintas yang telah dilakukan adalah 400 smp/jam.



Gambar 4.17 Grafik relasi arus dan kepadatan



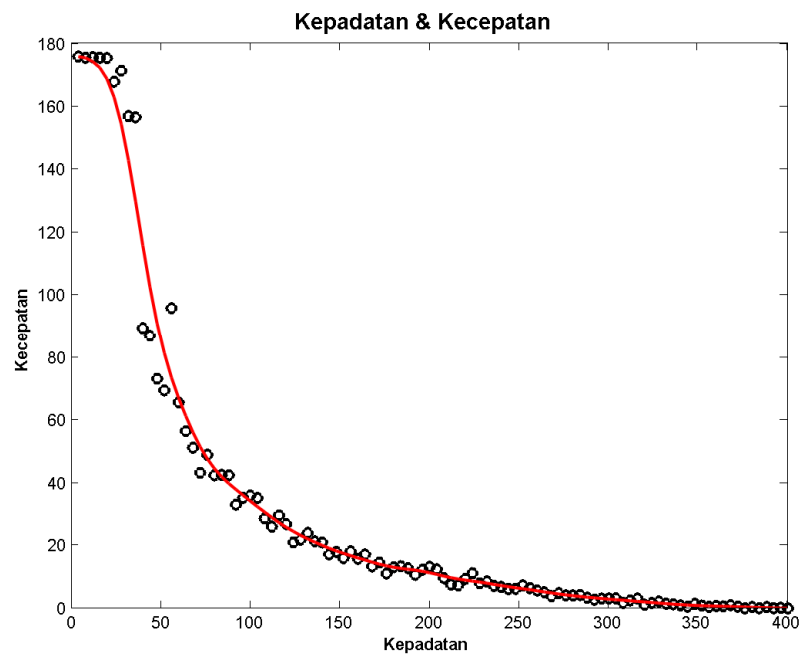
Gambar 4.18 Grafik relasi arus dan kepadatan Bellemans (Bellemans *et al.*, 2002)

Dari grafik relasi arus – kepadatan dapat dijelaskan karakteristik arus lalu lintas sebagai berikut: ketika kepadatan nol, arus juga akan bernilai nol. Ketika kepadatan mulai meningkat, maka hal tersebut juga akan meningkatkan arus yang terjadi. Untuk kepadatan yang rendah, hubungan antara arus dan kepadatan hampir linier karena interaksi yang terjadi antar kendaraan adalah kecil. Hal ini terjadi karena setiap kendaraan dapat bergerak bebas pada saat kepadatan rendah. Pada saat kepadatan meningkat lebih lanjut, interaksi antar kendaraan meningkat pesat. Misalnya kendaraan dengan kecepatan tinggi harus menurunkan kecepatannya atau berpindah lajur untuk mendahului kendaraan di depannya pada saat kendaraan di depannya lebih pelan dan cukup dekat.

Arus lalu lintas mencapai arus maksimum pada kepadatan kritis ρ_c , setelah itu arus akan semakin berkurang pada saat kepadatan meningkat. Dalam situasi tersebut, pengemudi cenderung mengurangi kecepatannya yang mengakibatkan penurunan arus, meskipun kepadatan semakin tinggi. Pada saat kepadatan mencapai

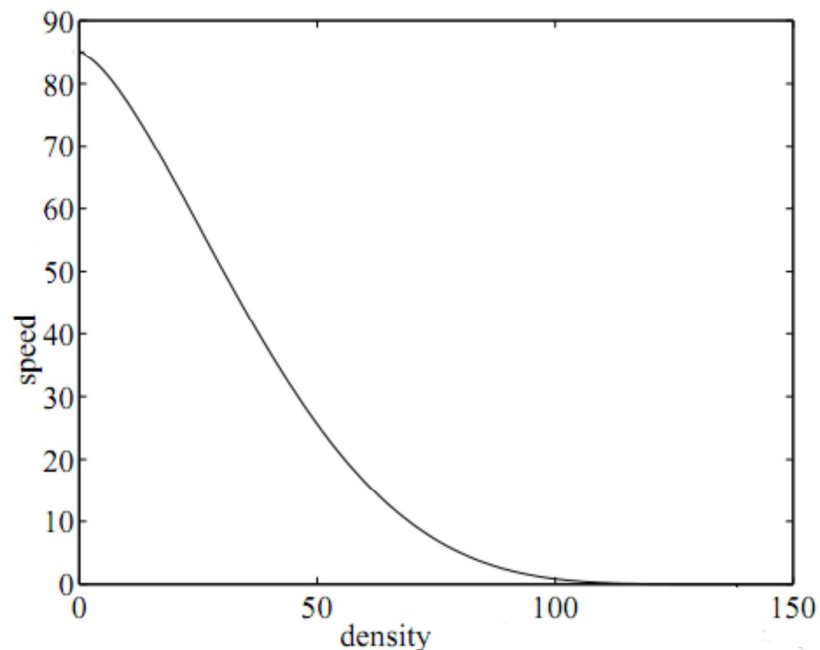
kepadatan maksimal akan terjadi kemacetan arus lalu lintas dimana kendaraan terpaksa berhenti karena kendaraan di depannya berhenti.

Gambar 4.19 menunjukkan grafik relasi kecepatan – kepadatan dari model arus lalu lintas dengan *Cellular Automata*. Dari gambar terlihat *free flow* masih dapat dicapai hingga kepadatan 0,09 dengan kecepatan yang terjadi sebesar 156,258 km/jam. *Free flow* adalah keadaan arus lalu lintas yang tidak terlalu padat, sehingga tiap kendaraan masih dapat melaju dengan kecepatan yang diinginkan tanpa terpengaruh oleh keadaan luar. Setelah kepadatan 0,09, keadaan *free flow* tidak dapat dipenuhi lagi dan kecepatan mengalami penurunan drastis menjadi 89,217 km/jam pada saat kepadatan 0,1. Hal ini disebabkan karena kemacetan yang mulai ditimbulkan oleh pengemudi yang menurunkan kecepatannya pada tahap perlambatan. Kecepatan akan semakin menurun pada saat kepadatan terus meningkat karena kemacetan yang terjadi terus bertambah dan sulit terurai hingga akhirnya terjadi kemacetan total pada saat kepadatan maksimal yang mengakibatkan kecepatan nol.



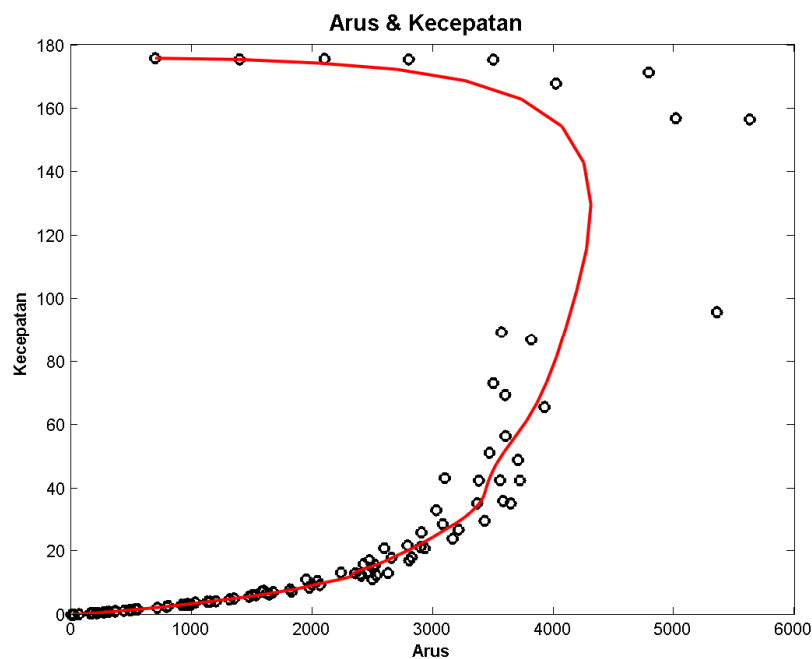
Gambar 4.19 Grafik relasi kecepatan dan kepadatan

Diagram fundamental relasi kecepatan – kepadatan menggambarkan relasi linier dengan gradien negatif, sedangkan grafik yang dihasilkan pada simulasi arus lalu lintas dengan *Cellular Automata* menggambarkan relasi eksponensial antara kecepatan dan kepadatan. Hal tersebut sama dengan pemodelan arus lalu lintas yang dilakukan oleh Underwood yang menyimpulkan bahwa relasi antara kecepatan dan kepadatan adalah eksponensial (Mathew dan Rao, 2006). Pemodelan arus lalu lintas yang dilakukan oleh Bellemans *et al.* (2002) juga menghasilkan grafik hubungan kecepatan – kepadatan seperti pada Gambar 4.20 yang juga menyerupai kurva eksponensial. Grafik yang serupa juga dihasilkan dari pemodelan arus lalu lintas oleh Maerivoet dan Moor (2005). Sedangkan dalam Manual Kapasitas Jalan Indonesia (MKJI), dari survey yang dilakukan di Indonesia disimpulkan bahwa relasi kecepatan dan kepadatan arus lalu lintas di Indonesia bukanlah linier tetapi eksponensial seperti pada Gambar 4.20 (Departemen Pekerjaan Umum Ditjen Bina Marga, 1997).



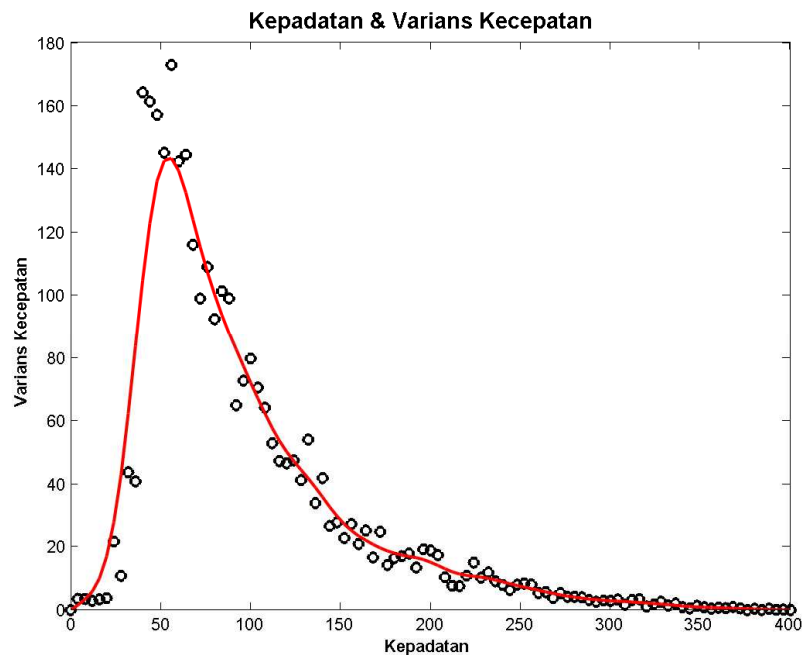
Gambar 4.20 Grafik relasi kecepatan dan kepadatan Bellemans (Bellemans *et al.*, 2002)

Gambar 4.21 adalah grafik relasi arus – kecepatan dari model arus lalu lintas dengan *Cellular Automata*. Grafik yang dihasilkan menyerupai diagram fundamental untuk relasi arus – kecepatan dan grafik relasi arus – kecepatan dalam Manual Kapasitas Jalan Indonesia (Departemen Pekerjaan Umum Ditjen Bina Marga, 1997). Dari Gambar 4.21 dapat diketahui bahwa pada saat arus rendah dapat terjadi kecepatan yang sangat tinggi dan kecepatan yang mendekati nol. Hal ini dapat terjadi pada saat jumlah kendaraan di jalan sedikit dan pada saat jumlah kendaraan di jalan sangat banyak. Ketika jumlah kendaraan di jalan sedikit, maka arus yang terjadi akan rendah dan setiap kendaraan dapat bebas menaikkan kecepatan kendaraan hingga mencapai kecepatan maksimal tanpa takut terganggu keberadaan kendaraan lainnya. Dalam kasus tersebut akan tercapai kecepatan yang tinggi. Ketika jalan banyak dipadati kendaraan, maka tentulah arus yang terjadi akan tinggi tetapi dengan begitu akan terjadi kemacetan pada arus lalu lintas yang terjadi. Pada saat tersebut setiap kendaraan dipaksa untuk menurunkan kecepatannya sehingga kecepatan yang terjadi sangat rendah.



Gambar 4.21 Grafik relasi arus dan kecepatan

Selain ketiga relasi sebelumnya yang mengacu pada diagram fundamental, relasi lain yang dapat dianalisis adalah relasi varians kecepatan – kepadatan. Pada Gambar 4.22 varians kecepatan terus meningkat dan mencapai maksimal pada saat kepadatan 0,1 sebesar 163,9728 km/jam, tidak berbeda jauh dengan batas kepadatan dimana kendaraan masih dapat mencapai keadaan *free flow*. Sebelum kepadatan 0,1, varians kecepatan terus meningkat disebabkan mulai munculnya kemacetan-kemacetan kecil yang timbul dan hilang pada lintasan. Setelah kepadatan 0,1, varians kecepatan mulai turun. Hal ini dikarenakan semakin panjangnya kemacetan yang terjadi dan kemacetan tersebut sulit untuk hilang, sehingga menyebabkan makin banyak mobil yang berhenti atau terjebak kemacetan dan menurunkan varians kecepatan secara global.



Gambar 4.22 Grafik relasi varians kecepatan dan kepadatan

BAB 5. PENUTUP

5.1 Kesimpulan

Dari hasil dan pembahasan dapat ditarik kesimpulan bahwa *Cellular Automata* dapat digunakan untuk memodelkan arus lalu lintas yang berisi kendaraan roda empat dan kendaraan roda dua dengan baik. Relasi yang terjadi antar parameter arus lalu lintas yang diperoleh dari hasil simulasi adalah sebagai berikut:

- a. Relasi arus – kepadatan adalah parabolik dimana grafik relasi keduanya hampir menyerupai segitiga karena relasi sebelum dan setelah arus maksimal terjadi adalah hampir linier. Semakin tinggi kepadatan, maka arus akan semakin tinggi sampai nilai kepadatan tertentu dimana terjadi arus maksimum, setelah itu semakin padat arus lalu lintas yang terjadi, maka arus akan semakin kecil.
- b. Relasi kecepatan – kepadatan adalah eksponensial, kecepatan akan terus menurun seiring dengan meningkatnya kepadatan. *Free flow* terjadi dari kepadatan nol hingga kepadatan tertentu. Setelah itu nilai kecepatan akan menurun drastis, tetapi pada akhirnya relasi kecepatan dan kepadatan hampir linier karena kemacetan yang terjadi semakin lama terurai.
- c. Relasi kecepatan – arus adalah parabolik, hal tersebut menunjukkan bahwa semakin besar kecepatan akan menyebabkan arus terus bergerak naik dan kemudian arus akan terus bergerak turun setelah kecepatan tertentu yang menjadi puncak parabola.
- d. Relasi varians kecepatan – kepadatan mirip dengan diagram fundamental relasi arus – kepadatan yaitu berupa parabolik tetapi memiliki ekor yang lebih panjang di bagian kanannya. Hal ini terjadi karena semakin padat arus lalu lintas, maka varians dari kecepatan semakin mengecil. Dimana sebelumnya varians kecepatan terus membesar seiring dengan penambahan kepadatan dan berhenti pada saat kepadatan tertentu yang kemudian varians kecepatan terus menurun.

5.2 Saran

Masih terdapat banyak permasalahan bagi pembaca yang ingin mengembangkan pemodelan arus lalu lintas menggunakan *Cellular Automata* ini. Permasalahan tersebut antara lain adalah pemodelan arus lalu lintas dengan menggunakan algoritma yang lebih efisien; pembuatan program simulasi menggunakan software lain sehingga pemvisualisasiannya lebih baik dan menarik; pemodelan jenis arus lalu lintas yang memiliki karakteristik berbeda. Beberapa jenis arus lalu lintas lainnya yang dimaksud adalah sebagai berikut:

- a. arus lalu lintas terganggu (terdapat lampu lalu lintas, marka lalu lintas, atau rambu lalu lintas);
- b. arus lalu lintas dengan simpangan masuk dan/atau simpangan keluar;
- c. arus lalu lintas dua arah.

DAFTAR PUSTAKA

- Bellemans, T., Schutter, B. D., & Moor, B. D. 2002. Models for Traffic Control. *Journal A*. Vol. 43(3–4): 13–22.
- Benjaafar, S., Dooley, K., & Setyawan, W. 1997. *Cellular Automata for Traffic Flow Modeling*. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.32.3670&rep=rep1&type=pdf>. [15 Nopember 2010].
- Daoudia, A. K., & Moussa, N. 2003. Numerical Simulations of a Three-Lane Traffic Model Using Cellular Automata. *Chines. J. Phys.* Vol. 41(6): 671-681.
- Departemen Pekerjaan Umum Ditjen Bina Marga. 1997. *Manual Kapasitas Jalan Indonesia (MKJI)*. Indonesia: Departemen Pekerjaan Umum.
- Ilachinski, A. 2001. *Cellular Automata: A Discrete Universe*. Singapore: World Scientific Publishing.
- Maerivoet, S. & Moor, B. D. 2005. Cellular Automata Models of Road Traffic. *Physics Reports*. Vol. 419 (1): 1–64.
- Mathew, T. V., & Rao, K. V. K. 2006. *Introduction to Transportation Engineering*. <http://www.cdeep.iitb.ac.in/nptel/Civil%20Engineering/Transportation%20Engg%201/TOC.htm>. [25 September 2011].
- Nagel, K. & Schreckenberg, M. 1992. A Cellular Automaton Model for Freeway Traffic. *J. Phys. I France*. Vol. 2(12): 2221-2229.
- Roess, R. P., Prassas, E. S., & McShane, W. R. 2004. *Traffic Engineering*. New Jersey: Pearson Prentice Hall.
- Sudrajat, Sumarto, T., & Asropi. 2008. *Model Jalan Lalu Lintas Jalan Tol Dalam Persamaan Diferensial Parsial*. http://pustaka.unpad.ac.id/wp-content/uploads/2010/08/model_lalulintas_dengan_pdp.pdf. [27 Oktober 2010].
- Susiliawati, R. 2009. *Tinjauan matematis pada model makroskopik dan mikroskopik arus lalu lintas*. <http://repository.ipb.ac.id/bitstream/handle/123456789/17021/G09rsu.pdf?sequence=2>. [20 Pebruari 2011].
- Syalim, A. 2003. *Cellular Automata: Pemodelan dan Implementasi Paralel untuk Simulasi Arus Lalu Lintas Kendaraan di Jalan Raya*. <http://sun.csce.kyushu-u.ac.jp/~amr/amril-ta.pdf>. [4 September 2010].

LAMPIRAN A. LISTING PROGRAM GUI RodaDua

```

function RodaDuaGUI ()

%% RodaDua
%Program Simulasi Arus Lalu Lintas Roda Dua dengan cellular Automata
%Abduh Riski
%071810101005
%Jurusan Matematika FMIPA UNEJ
%Nopember 2011, Jember
%razkhy@yahoo.com
%%

close all;
clear all;
clc;
win1=figure(...
    'units','points',...
    'position',[50 50 500 310],...
    'color',[.8 .8 .9],...
    'menubar','none',...
    'resize','off',...
    'Visible','off',...
    'numbertitle','off',...
    'name',...
    'Simulasi Arus Lalu Lintas dengan Cellular Automata :: Abduh
Riski');
movegui(win1,'center');
set(win1,'Visible','on');
sizewin1=get(win1,'Position');

frame1=uicontrol('parent',win1,...
    'units','points',...
    'position',[0 265 sizewin1(3)+10 50],...
    'backgroundcolor',[.3 .3 .4],...
    'style','Frame');

frame2=uicontrol('parent',win1,...
    'units','points',...
    'position',[0 0 sizewin1(3)+10 50],...
    'backgroundcolor',[.3 .3 .4],...
    'style','Frame');

labell1=uicontrol('parent',win1,...
    'units','points',...
    'position',[0 270 500 30],...
    'backgroundcolor',[.3 .3 .4],...
    'style','Text',...
    'horizontalalignment','center',...

```

```

        'string','Simulasi Arus Lalu Lintas dengan Cellular
Automata',...
        'fontname','broadway',...
        'fontsize',15,...
        'fontweight','bold',...
        'foregroundcolor',[1 1 1]);

label2=icontrol('parent',win1,...
    'units','points',...
    'position',[10 230 270 15],...
    'backgroundcolor',[.8 .9 .9],...
    'style','Text',...
    'horizontalalignment','left',...
    'string','Panjang Jalan : ',...
    'fontname','arial',...
    'fontsize',11);
label3=icontrol('parent',win1,...
    'units','points',...
    'position',[10 210 270 15],...
    'backgroundcolor',[.8 .9 .9],...
    'style','Text',...
    'horizontalalignment','left',...
    'string','Kepadatan jalan (%) : ',...
    'fontname','arial',...
    'fontsize',11);
label4=icontrol('parent',win1,...
    'units','points',...
    'position',[10 190 270 15],...
    'backgroundcolor',[.8 .9 .9],...
    'style','Text',...
    'horizontalalignment','left',...
    'string','Jumlah kendaraan roda 2 (%) : ',...
    'fontname','arial',...
    'fontsize',11);
label5=icontrol('parent',win1,...
    'units','points',...
    'position',[10 170 270 15],...
    'backgroundcolor',[.8 .9 .9],...
    'style','Text',...
    'horizontalalignment','left',...
    'string','Jumlah kendaran roda 4 (%) : ',...
    'fontname','arial',...
    'fontsize',11);
label6=icontrol('parent',win1,...
    'units','points',...
    'position',[10 150 270 15],...
    'backgroundcolor',[.8 .9 .9],...
    'style','Text',...
    'horizontalalignment','left',...
    'string','Probabilitas pengemudi menurunkan kecepatan (%) :
',...
    'fontname','arial',...

```

```

        'fontsize',11);
label7=uicontrol('parent',win1,...
    'units','points',...
    'position',[10 130 270 15],...
    'backgroundcolor',[.8 .9 .9],...
    'style','Text',...
    'horizontalalignment','left',...
    'string','Probabilitas pengemudi berpindah lajur (%) : ',...
    'fontname','arial',...
    'fontsize',11);
label8=uicontrol('parent',win1,...
    'units','points',...
    'position',[10 110 270 15],...
    'backgroundcolor',[.8 .9 .9],...
    'style','Text',...
    'horizontalalignment','left',...
    'string','Kecepatan Maksimal (0-25) : ',...
    'fontname','arial',...
    'fontsize',11);
label9=uicontrol('parent',win1,...
    'units','points',...
    'position',[10 90 270 15],...
    'backgroundcolor',[.8 .9 .9],...
    'style','Text',...
    'horizontalalignment','left',...
    'string','Banyak step/waktu : ',...
    'fontname','arial',...
    'fontsize',11);
label10=uicontrol('parent',win1,...
    'units','points',...
    'position',[10 70 270 15],...
    'backgroundcolor',[.8 .9 .9],...
    'style','Text',...
    'horizontalalignment','left',...
    'string','Waktu delay simulasi (detik) : ',...
    'fontname','arial',...
    'fontsize',11);
label10=uicontrol('parent',win1,...
    'units','points',...
    'position',[5 5 270 15],...
    'backgroundcolor',[.3 .3 .4],...
    'style','Text',...
    'horizontalalignment','left',...
    'string',sprintf('%s%s', 'Copyright ',char(169), ' RizQ'),...
    'fontname','Arial',...
    'fontweight','bold',...
    'fontsize',8);

edit2=uicontrol('parent',win1,...
    'units','points',...
    'position',[280 230 100 15],...
    'backgroundcolor',[1 1 1],...

```

```

        'TooltipString',...
        'Masukkan panjang jalan/ukuran kolom sel Cellular Automata',...
        'style','edit',...
        'string','',...
        'fontname','arial',...
        'fontsize',11);
edit3=uicontrol('parent',win1,...
    'units','points',...
    'position',[280 210 100 15],...
    'backgroundcolor',[1 1 1],...
    'TooltipString','Masukkan kepadatan jalan (dalam %)',...
    'style','edit',...
    'callback',@Fedit3,...
    'string','',...
    'fontname','arial',...
    'fontsize',11);
edit4=uicontrol('parent',win1,...
    'units','points',...
    'position',[280 190 100 15],...
    'backgroundcolor',[1 1 1],...
    'TooltipString','Masukkan jumlah kendaraan roda dua (dalam
%)',...
    'style','edit',...
    'callback',@Fedit4,...
    'string','',...
    'fontname','arial',...
    'fontsize',11);
edit5=uicontrol('parent',win1,...
    'units','points',...
    'position',[280 170 100 15],...
    'backgroundcolor',[1 1 1],...
    'TooltipString',...
    sprintf('%s\n%s','Jumlah kendaraan roda empat (dalam %)',...
    '100%-jumlah kendaraan roda dua'),...
    'style','edit',...
    'Enable','off',...
    'string','',...
    'fontname','arial',...
    'fontsize',11);
edit6=uicontrol('parent',win1,...
    'units','points',...
    'position',[280 150 100 15],...
    'backgroundcolor',[1 1 1],...
    'TooltipString',...
    'Masukkan probabilitas (dalam %) pengemudi menurunkan kecepatan
kendaraannya',...
    'style','edit',...
    'callback',@Fedit6,...
    'string','',...
    'fontname','arial',...
    'fontsize',11);
edit7=uicontrol('parent',win1,...

```

```

        'units','points',...
        'position',[280 130 100 15],...
        'backgroundcolor',[1 1 1],...
        'TooltipString',...
        'Masukkan probabilitas (dalam %) pengemudi berpindah lajur
kendaraannya',...
        'style','edit',...
        'callback',@Fedit7,...
        'string','',...
        'fontname','arial',...
        'fontsize',11);
edit8=icontrol('parent',win1,...
    'units','points',...
    'position',[280 110 100 15],...
    'backgroundcolor',[1 1 1],...
    'TooltipString',sprintf('%s\n%s\n%s',...
    'Masukkan batas kecepatan maksimal kendaraan dalam simulasi',...
    'Kecepatan maksimal terkecil yang diperbolehkan = 0',...
    'Kecepatan maksimal terbesar yang diperbolehkan = 25'),...
    'style','edit',...
    'callback',@Fedit8,...
    'string','',...
    'fontname','arial',...
    'fontsize',11);
edit9=icontrol('parent',win1,...
    'units','points',...
    'position',[280 90 100 15],...
    'backgroundcolor',[1 1 1],...
    'TooltipString',...
    'Masukkan banyak step/waktu iterasi yang diinginkan',...
    'style','edit',...
    'string','',...
    'fontname','arial',...
    'fontsize',11);
edit10=icontrol('parent',win1,...
    'units','points',...
    'position',[280 70 100 15],...
    'backgroundcolor',[1 1 1],...
    'TooltipString',...
    'Masukkan waktu untuk delay simulasi dalam detik',...
    'style','edit',...
    'string','',...
    'fontname','arial',...
    'fontsize',11);

tomsimulasi=icontrol('parent',win1,...
    'units','points',...
    'position',[390 230 100 15],...
    'style','pushbutton',...
    'TooltipString','Simulasikan Program RodaDua',...
    'callback',@Fsimulasi,...
    'string','Simulasikan',...

```

```

        'fontname','arial',...
        'fontsize',11);
tombersih=uicontrol('parent',win1,...
    'units','points',...
    'position',[390 210 100 15],...
    'style','pushbutton',...
    'TooltipString','Kosongkan input',...
    'callback',@Fbersih,...
    'string','Bersihkan',...
    'fontname','arial',...
    'fontsize',11);
tomarsip=uicontrol('parent',win1,...
    'units','points',...
    'position',[390 190 100 15],...
    'style','pushbutton',...
    'Enable','off',...
    'TooltipString',...
    sprintf('%s\n%s','Buka Arsip Program RodaDua',...
    'Dalam Format MS Word'),...
    'callback',@Farsip,...
    'string','Buka Arsip',...
    'fontname','arial',...
    'fontsize',11);
tomketerangan=uicontrol('parent',win1,...
    'units','points',...
    'position',[390 170 100 15],...
    'style','pushbutton',...
    'TooltipString','Keterangan Simbol Program RodaDua',...
    'callback','FketeranganGUI();',...
    'string','Keterangan',...
    'fontname','arial',...
    'fontsize',11);
tominfo=uicontrol('parent',win1,...
    'units','points',...
    'position',[390 150 100 15],...
    'style','pushbutton',...
    'TooltipString','Info Program RodaDua',...
    'callback','FinfoGUI();',...
    'string','Info',...
    'fontname','arial',...
    'fontsize',11);
tomkeluar=uicontrol('parent',win1,...
    'units','points',...
    'position',[390 110 100 15],...
    'style','pushbutton',...
    'TooltipString','Tutup Program RodaDua',...
    'callback','close all;',...
    'string','Keluar',...
    'fontname','arial',...
    'fontsize',11);

```

```
function Fedit3(hObject,eventdata)
```



```

pdt = str2num(get(edit3, 'String'));
trueinput = cekinput(3, pdt);
if ~trueinput
    msgbox(sprintf('%s\n%s', 'Input salah silahkan coba
kembali', ...
        'Interval kepadatan jalan=[0,100]'), 'Input
Salah', 'error');
    set(edit3, 'string', '');
end
end
function Fedit4(hObject, eventdata)
pdt2=str2num(get(edit4, 'String'));
set(edit5, 'string', num2str(100-pdt2));

trueinput = cekinput(4, pdt2);
if ~trueinput
    msgbox(sprintf('%s\n%s', 'Input salah silahkan coba
kembali', ...
        'Interval jumlah kendaraan roda dua=[0,100]'), ...
        'Input Salah', 'error');
    set(edit4, 'string', '');
end
end
function Fedit6(hObject, eventdata)
pSlow = str2num(get(edit6, 'String'));
trueinput = cekinput(6, pSlow);
if ~trueinput
    msgbox(sprintf('%s\n%s', 'Input salah silahkan coba
kembali', ...
        'Interval probabilitas pengemudi menurunkan
kecepatan=[0,100]'), ...
        'Input Salah', 'error');
    set(edit6, 'string', '');
end
end
function Fedit7(hObject, eventdata)
pMove = str2num(get(edit7, 'String'));
trueinput = cekinput(7, pMove);
if ~trueinput
    msgbox(sprintf('%s\n%s', 'Input salah silahkan coba
kembali', ...
        'Interval probabilitas pengemudi pindah
lajur=[0,100]'), ...
        'Input Salah', 'error');
    set(edit7, 'string', '');
end
end
function Fedit8(hObject, eventdata)
vmax = str2num(get(edit8, 'String'));
trueinput = cekinput(8, vmax);
if ~trueinput

```

```

        msgbox(sprintf('%s\n%s', 'Input salah silahkan coba
kembali', ...
        'Interval kecepatan maksimal kendaraan=[0,25]'), ...
        'Input Salah', 'error');
        set(edit8, 'string', '');
    end
end

function Fsimulasi(hObject,eventdata)
    clear all;
    cel = str2num(get(edit2, 'String'));
    row=4;
    pdt = str2num(get(edit3, 'String'));
    pdt2 = str2num(get(edit4, 'String'));
    pdt4 = str2num(get(edit5, 'string'));
    pSlow = str2num(get(edit6, 'String'));
    pMove = str2num(get(edit7, 'String'));
    vmax = str2num(get(edit8, 'String'));
    step = str2num(get(edit9, 'String'));
    delay = str2double(get(edit10, 'String'));
    RodaDua(cel, row, pdt, pdt2, pdt4, pSlow, pMove, vmax, step, delay);
    set(tomarsip, 'Enable', 'on');
end

function Fbersih(hObject,eventdata)
    clear all;
    set(edit2, 'string', '');
    set(edit3, 'string', '');
    set(edit4, 'string', '');
    set(edit5, 'string', '');
    set(edit6, 'string', '');
    set(edit7, 'string', '');
    set(edit8, 'string', '');
    set(edit9, 'string', '');
    set(edit10, 'string', '');
end

function Farsip(hObject,eventdata)
    try
        open rodaduahasil.doc
    catch ME
        msgbox(sprintf('%s\n%s\n%s', ...
            'File arsip rodaduahasil.doc tidak dapat di buka', ...
            'Silahkan buka secara manual', ...
            'Atau Silahkan Ulangi Simulasi!'), 'Arsip gagal dibuka',
'error');
    end
end
end
end

```

LAMPIRAN B. LISTING PROGRAM TAHAP PENENTUAN POSISI AWAL KENDARAAN

```

function id24=acakPos(panjang_sel,baris_sel, pdt, pdt2)
%% fungsi untuk membangkitkan posisi awal kendaraan atau untuk
% membangkitkan status awal sel

id24=zeros(baris_sel,panjang_sel);
isi= pdt*(baris_sel*panjang_sel/100);
isi2=round(pdt2*isi/100);
isi4=round((isi-isi2)/2);
idisi2=0;idisi4=0;
penuh=0;
while penuh==0
    for i=1:baris_sel
        for j=1:panjang_sel
            rnd=100*rand;
            if rnd<=pdt
                rnd2=100*rand;
                if rnd2<=pdt2 && id24(i,j)==0 && idisi2<isi2
                    id24(i,j)=2;
                    idisi2=idisi2+1;
                elseif rnd2>pdt2 && id24(i,j)==0 && idisi4<isi4
                    switch i
                        case 1
                            if id24(2,j)==0
                                id24(1,j)=400+round(46*rand)+...
                                    round(10*rand)+round(24*rand)+...
                                    round(20*rand);
                                id24(2,j)=id24(1,j);
                                idisi4=idisi4+1;
                            end
                        case baris_sel
                            if id24(baris_sel-1,j)==0
                                id24(baris_sel,j)=...
                                    400+round(46*rand)+...
                                    round(10*rand)+round(24*rand)+...
                                    round(20*rand);
                                id24(baris_sel-1,j)=...
                                    id24(baris_sel,j);
                                idisi4=idisi4+1;
                            end
                        otherwise
                            if id24(i-1,j)==0
                                id24(i,j)=400+round(46*rand)+...
                                    round(10*rand)+round(24*rand)+...
                                    round(20*rand);
                                id24(i-1,j)=id24(i,j);
                                idisi4=idisi4+1;
                            end
                    end
                end
            end
        end
    end
    penuh=1;
end

```

```
else
    id24(i,j)=400+round(46*rand)+...
    round(10*rand)+round(24*rand)+...
    round(20*rand);
    id24(i+1,j)=id24(i,j);
    idisi4=idisi4+1;
end
end
end
end
end
end
if idisi2>=isi2 && idisi4>=isi4
    penuh=1;
end
end
end
```

LAMPIRAN C. LISTING PROGRAM TAHAP PINDAH LAJUR

```

function jalan=Pindah(jal, cel, n, pMove, vmax)

%% fungsi pindah lajur

penjal=Fpenjal(jal);
jal=Fchar2num(jal);

for i=1:n
    for j=1:cel
        try
            if penjal(i,j)==2
                gap=FgapDepan(i,j,jal);

                gapkanandepan=0;
                gapkananbelakang=0;
                if i<n
                    gapkanandepan=FgapDepan(i+1,j,jal);
                    gapkananbelakang=FgapBelakang(i+1,j,jal);
                end

                gapkiridepan=0;
                gapkiribelakang=0;
                if i>1
                    gapkiridepan=FgapDepan(i-1,j,jal);
                    gapkiribelakang=FgapBelakang(i-1,j,jal);
                end

                vmaxtemp=0;
                for v=1:n
                    for w=j-1:-1:1
                        if jal(v,w)~=100
                            vmaxtemp=max([vmaxtemp jal(v,w)]);
                        end
                    end
                end

                kiri=false;
                kanan=false;
                try
                    if jal(i+1,j)==100
                        kanan=true;
                        gapsampingdepan=gapkanandepan;
                        gapsampingbelakang=gapkananbelakang;
                    elseif jal(i-1,j)==100
                        kiri=true;
                        gapsampingdepan=gapkiridepan;
                    end
                end
            end
        end
    end
end

```

```

        gapsampingbelakang=gapkiribelakang;
    end
catch ME
    %indeks terlalu besar
end

if kiri && kanan
    if gapkiridepan>gapkanandepan && ...
        gapkiribelakang>gapkananbelakang
        kanan=false;
        gapsampingdepan=gapkiridepan;
        gapsampingbelakang=gapkiribelakang;
    else
        kiri=false;
        gapsampingdepan=gapkanandepan;
        gapsampingbelakang=gapkananbelakang;
    end
end

if kiri || kanan
    probMove=100*rand;
    if jal(i,j)>=gap && jal(i,j)<...
        gapsampingdepan-1 && ...
        gapsampingbelakang>vmaxtemp && ...
        probMove<=pMove
        if kiri
            jal(i-1,j)=jal(i,j);
            jal(i,j)=100;
            penjal(i-1,j)=penjal(i,j);
            penjal(i,j)=0;
        elseif kanan
            jal(i+1,j)=jal(i,j);
            jal(i,j)=100;
            penjal(i+1,j)=penjal(i,j);
            penjal(i,j)=0;
        end
    end
end

else
    [pindahkiri pindahkanan]=...
        PindahR2R4(jal,penjal,i,j);
    if pindahkiri && (100*rand)<=pMove
        jal(i-1,j)=jal(i,j);
        jal(i,j)=100;
        penjal(i-1,j)=penjal(i,j);
        penjal(i,j)=0;
    elseif pindahkanan && (100*rand)<=pMove
        jal(i+1,j)=jal(i,j);
        jal(i,j)=100;
        penjal(i+1,j)=penjal(i,j);
        penjal(i,j)=0;
    end
end
end
end

```

```

elseif penjala(i,j)>=400 && penjala(i+1,j)==...
    penjala(i,j) && i<n
    gapM = FgapDepanM(i,j,jala);

    gapMkanandepan=0;
    gapMkananbelakang=0;
    if i<n-1
        gapMkanandepan = FgapDepanM(i+1,j,jala);
        gapMkananbelakang = FgapBelakangM(i+1,j,jala);
    end

    gapMkiridepan=0;
    gapMkiribelakang=0;
    if i>1
        gapMkiridepan = FgapDepanM(i-1,j,jala);
        gapMkiribelakang = FgapBelakangM(i-1,j,jala);
    end

    vmaxMtemp=0;
    for v=1:n
        for w=j-1:-1:1
            if jala(v,w)~=100
                vmaxMtemp=max([vmaxMtemp jala(v,w)]);
            end
        end
    end

    kiri=false;
    kanan=false;
    try
        if jala(i+2,j)==100
            kanan=true;
            gapMsampingdepan=gapMkanandepan;
            gapMsampingbelakang=gapMkananbelakang;
        elseif jala(i-1,j)==100
            kiri=true;
            gapMsampingdepan=gapMkiridepan;
            gapMsampingbelakang=gapMkiribelakang;
        end
    catch ME
        %indeks terlalu besar
    end

    if kiri && kanan
        if gapMkiridepan>gapMkanandepan && ...
            gapMkiribelakang>gapMkananbelakang
            kanan=false;
            gapMsampingdepan=gapMkiridepan;
            gapMsampingbelakang=gapMkiribelakang;
        else

```

```

        kiri=false;
        gapMsampingdepan=gapMkanandepan;
        gapMsampingbelakang=gapMkananbelakang;
    end
end

if kiri || kanan
    probMove=100*rand;
    if jal(i,j)>=gapM && jal(i,j)<...
        gapMsampingdepan-1 && ...
        gapMsampingbelakang>vmaxMtemp && ...
        probMove<=pMove
        if kiri
            jal(i-1,j)=jal(i,j);
            jal(i,j)=jal(i+1,j);
            jal(i+1,j)=100;
            penjal(i-1,j)=penjal(i,j);
            penjal(i,j)=penjal(i+1,j);
            penjal(i+1,j)=0;
        elseif kanan
            jal(i+2,j)=jal(i+1,j);
            jal(i+1,j)=jal(i,j);
            jal(i,j)=100;
            penjal(i+2,j)=penjal(i+1,j);
            penjal(i+1,j)=penjal(i,j);
            penjal(i,j)=0;
        end
    end
end
end
catch ME
    %indeks terlalu besar
end
end
end

jal=Fnum2char(jal,penjal);
jalan=jal;
end

```


LAMPIRAN D. LISTING PROGRAM TAHAP PERCEPATAN

```

function jalan=Percepatan(jal,panjang_sel,baris_sel,vmax)
%% Fungsi tahap percepatan

penjal=Fpenjal(jal);
jal=Fchar2num(jal);
for i=1:baris_sel
    for j=1:panjang_sel
        try
            if penjal(i,j)==2
                gap=FgapDepan(i,j,jal);
                jal(i,j)=min([jal(i,j)+1 gap vmax]);
            elseif penjal(i,j)>=400 && penjal(i+1,j)==...
                penjal(i,j) && i<baris_sel
                gapM = FgapDepanM(i,j,jal);
                jal(i,j)=min([jal(i,j)+1 gapM vmax]);
                jal(i+1,j)=jal(i,j);
            end
        catch ME
            %indeks terlalu besar
        end
    end
end
jal=Fnum2char(jal,penjal);
jalan=jal;
end

```

LAMPIRAN E. LISTING PROGRAM TAHAP PERLAMBATAN

```

function jalan=Perlambatan(jal,panjang_sel,baris_sel,pSlow)
%% fungsi tahap perlambatan

penjal=Fpenjal(jal);
jal=Fchar2num(jal);
for i=1:baris_sel
    for j=1:panjang_sel
        angka_random=100*rand;
        try
            if jal(i,j)~=100 && jal(i,j)>0 && angka_random<=pSlow
                if penjal(i,j)==2
                    jal(i,j)=jal(i,j)-1;
                elseif penjal(i,j)>=400 && penjal(i+1,j)==...
                    penjal(i,j) && i<baris_sel
                        jal(i,j)=jal(i,j)-1;
                        jal(i+1,j)=jal(i,j);
                    end
                end
            catch ME
                %indeks terlalu besar
            end
        end
    end
end
jal=Fnum2char(jal,penjal);
jalan=jal;
end

```

LAMPIRAN F. LISTING PROGRAM TAHAP PINDAH SEL

```

function jalan=moveCar(jal,panjang_sel,baris_sel)
%% fungsi tahap pindah sel

penjal=Fpenjal(jal);
jal=Fchar2num(jal);

idpindah=zeros(baris_sel,panjang_sel);
for i=1:baris_sel
    for j=panjang_sel:-1:1
        if jal(i,j)>=1 && jal(i,j)<=25
            idpindah(i,j)=j+jal(i,j);
        end
    end
end

for i=1:baris_sel
    for j=panjang_sel:-1:1
        if idpindah(i,j)~=0
            if mod(idpindah(i,j),panjang_sel)==0
                p=panjang_sel;
            else
                p=mod(idpindah(i,j),panjang_sel);
            end
            penjal(i,p)=penjal(i,j);
            penjal(i,j)=0;
            jal(i,p)=jal(i,j);
            jal(i,j)=100;
        end
    end
end
jalan=Fnum2char(jal,penjal);
end

```

LAMPIRAN G. TABEL HASIL SIMULASI

Input:

Panjang sel : 100 sel

Banyak kendaraan roda dua : 60 %

Banyak kendaraan roda empat : 40 %

Probabilitas pengemudi memperlambat kendaraan : 20 %

Probabilitas pengemudi pindah : 90 %

Kecepatan maksimal : 10 sel

Banyak step (waktu) : 1000 iterasi

| Kepadatan (smp/km) | Arus (smp/jam) | Kecepatan (km/jam) | Varians Kecepatan |
|-------------------------------|---------------------------|-------------------------------|------------------------------|
| 4 | 702 | 175,518 | 3,55842 |
| 8 | 1401 | 175,1742 | 3,54348 |
| 12 | 2104 | 175,3272 | 2,95722 |
| 16 | 2802 | 175,1364 | 3,4263 |
| 20 | 3501 | 175,0662 | 3,81618 |
| 24 | 4021 | 167,5584 | 21,7116 |
| 28 | 4789 | 171,0324 | 10,84914 |
| 32 | 5012 | 156,6162 | 43,5636 |
| 36 | 5625 | 156,258 | 40,7034 |
| 40 | 3569 | 89,217 | 163,9728 |
| 44 | 3816 | 86,7186 | 161,1072 |
| 48 | 3502 | 72,9594 | 156,8898 |
| 52 | 3600 | 69,2262 | 144,9288 |
| 56 | 5353 | 95,58 | 172,6722 |
| 60 | 3925 | 65,4192 | 142,1514 |
| 64 | 3603 | 56,3022 | 144,2988 |
| 68 | 3470 | 51,0228 | 115,8282 |
| 72 | 3100 | 43,0578 | 98,7948 |
| 76 | 3706 | 48,7584 | 108,7866 |
| 80 | 3382 | 42,2784 | 92,268 |
| 84 | 3558 | 42,3558 | 101,1672 |
| 88 | 3722 | 42,3 | 98,8272 |
| 92 | 3030 | 32,9382 | 64,7838 |
| 96 | 3368 | 35,0856 | 72,5436 |
| 100 | 3583 | 35,829 | 79,5996 |
| 104 | 3646 | 35,0604 | 70,38 |
| 108 | 3083 | 28,5498 | 63,9936 |

| | | | |
|-----|------|----------|----------|
| 112 | 2906 | 25,947 | 52,7886 |
| 116 | 3430 | 29,5704 | 47,1402 |
| 120 | 3214 | 26,7858 | 46,4292 |
| 124 | 2599 | 20,961 | 47,3562 |
| 128 | 2792 | 21,8106 | 41,1138 |
| 132 | 3165 | 23,9742 | 53,9406 |
| 136 | 2898 | 21,312 | 33,8364 |
| 140 | 2935 | 20,9646 | 41,769 |
| 144 | 2475 | 17,18568 | 26,6076 |
| 148 | 2659 | 17,96598 | 27,7416 |
| 152 | 2427 | 15,96636 | 22,8132 |
| 156 | 2826 | 18,117 | 27,2268 |
| 160 | 2522 | 15,76368 | 20,9358 |
| 164 | 2806 | 17,11242 | 25,1784 |
| 168 | 2240 | 13,33062 | 16,66566 |
| 172 | 2507 | 14,5755 | 24,7932 |
| 176 | 1952 | 11,09088 | 14,32242 |
| 180 | 2358 | 13,09878 | 16,51014 |
| 184 | 2463 | 13,38498 | 17,04834 |
| 188 | 2400 | 12,76506 | 17,91522 |
| 192 | 2045 | 10,65078 | 13,4847 |
| 196 | 2408 | 12,28554 | 19,2402 |
| 200 | 2629 | 13,1445 | 18,8784 |
| 204 | 2534 | 12,42162 | 17,42454 |
| 208 | 2000 | 9,61452 | 10,40886 |
| 212 | 1600 | 7,54614 | 7,72884 |
| 216 | 1583 | 7,32654 | 7,59546 |
| 220 | 2067 | 9,39456 | 10,92798 |
| 224 | 2501 | 11,16396 | 15,05826 |
| 228 | 1820 | 7,98408 | 10,27944 |
| 232 | 1979 | 8,53074 | 11,916 |
| 236 | 1680 | 7,1172 | 9,15948 |
| 240 | 1628 | 6,78348 | 7,9821 |
| 244 | 1506 | 6,17364 | 6,41916 |
| 248 | 1538 | 6,20028 | 8,0361 |
| 252 | 1831 | 7,2657 | 8,4303 |
| 256 | 1648 | 6,43806 | 8,11872 |
| 260 | 1479 | 5,68836 | 5,42034 |
| 264 | 1356 | 5,1345 | 5,73678 |

| | | | |
|-----|------|-----------|-----------|
| 268 | 1034 | 3,8574 | 3,84624 |
| 272 | 1317 | 4,84056 | 5,4252 |
| 276 | 1134 | 4,10994 | 4,21182 |
| 280 | 1158 | 4,13388 | 4,24368 |
| 284 | 1206 | 4,24746 | 4,08222 |
| 288 | 970 | 3,36942 | 3,19896 |
| 292 | 810 | 2,77542 | 2,64996 |
| 296 | 920 | 3,10824 | 3,06576 |
| 300 | 943 | 3,1437 | 2,95758 |
| 304 | 971 | 3,19554 | 3,33756 |
| 308 | 537 | 1,744254 | 1,706868 |
| 312 | 796 | 2,54988 | 3,2535 |
| 316 | 998 | 3,15918 | 3,49488 |
| 320 | 373 | 1,16514 | 1,128024 |
| 324 | 558 | 1,723158 | 1,83708 |
| 328 | 720 | 2,19402 | 2,68812 |
| 332 | 491 | 1,479618 | 1,447362 |
| 336 | 444 | 1,322388 | 2,115 |
| 340 | 317 | 0,933426 | 0,918882 |
| 344 | 186 | 0,540504 | 0,531126 |
| 348 | 498 | 1,430082 | 1,39635 |
| 352 | 304 | 0,862686 | 0,885204 |
| 356 | 172 | 0,4824 | 0,484938 |
| 360 | 275 | 0,76446 | 0,798516 |
| 364 | 224 | 0,614322 | 0,629316 |
| 368 | 370 | 1,00602 | 0,99963 |
| 372 | 218 | 0,58473 | 0,590544 |
| 376 | 8 | 0,0214632 | 0,0252 |
| 380 | 164 | 0,430848 | 0,443538 |
| 384 | 24 | 0,062469 | 0,0807264 |
| 388 | 168 | 0,4329 | 0,446148 |
| 392 | 29 | 0,0732096 | 0,1092024 |
| 396 | 71 | 0,18054 | 0,180612 |
| 400 | 10 | 0,0250056 | 0,0305262 |