
AGENT-BASED MODELLING AND SIMULATION DENGAN NETLOGO 6.0

**EKO NURSUBIYANTORO
ASTRID WAHYU ADVENTRI WIBOWO
ISMANTI**

Agent-Based Modelling and Simulation dengan Netlogo 6.0

Penulis :
Eko Nursubiyantoro
Astrid Wahyu Adventri Wibowo
Ismianti

LEMBAGA PENELITIAN DAN PENGABDIAN KEPADA
MASYARAKAT
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”
YOGYAKARTA
2020

AGENT-BASED MODELLING AND SIMULATION DENGAN NETLOGO 6.0

Penulis : Eko Nursubiyantoro
Astrid Wahyu Adventri Wibowo
Ismianti

Copyright © 2020, pada penulis

Hak Cipta dilindungi Undang-undang
Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ini dalam bentuk apapun, baik secara elektronik maupun mekanis, termasuk memfotocopy, merekam atau dengan sistem penyimpanan lainnya, tanpa izin tertulis dari Penulis.

Nursubiyantoro, E., Wibowo, A.W.A., Ismianti, I.

Agent-Based Modelling and Simulation Dengan Netlogo 6.0 / Ed. I
Nursubiyantoro, E., Wibowo, A.W.A., Ismianti, I. –
Yogyakarta. 2020

viii + 74 ; 20 cm

ISBN 978-623-6797-16-7

Diterbitkan oleh:
Lembaga Penelitian dan Pengabdian kepada Masyarakat
UPN Veteran Yogyakarta
Jl. Pajajaran (Lingkar Utara), Condongcatur , Yogyakarta, 55283
Telp. (0274) 486188,486733, Fax. (0274) 486400
e-mail: lppm@upnyk.ac.id.

DAFTAR ISI

DAFTAR ISI.....	iii
DAFTAR GAMBAR	iv
DAFTAR TABEL.....	v
KATA PENGANTAR.....	vi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	3
1.2 Sistem Kompleks dan Kemunculannya	5
1.3 Aplikasi ABMS	8
1.4 Kontribusi ABMS terhadap Pandemi Covid-19.....	13
BAB II TEORI	17
2.1. Pengertian ABMS.....	19
2.2.1. Agen Otonom (<i>Autonomous Agents</i>)	20
2.2.2. <i>Interacting Agents</i>	23
2.2.3. <i>Agent Environment</i>	24
2.2. Sejarah ABMS.....	25
2.3. Karakteristik ABMS	30
2.3.1. <i>Specification and Formalisation</i>	31
2.3.2. <i>Modelling, Verification, and Experimentation</i>	33
2.3.3. <i>Calibration and Validation</i>	34
2.4. <i>Framework</i> ABMS	35
2.5. Aplikasi ABMS	39
BAB III IMPLEMENTASI ABMS DENGAN SOFTWARE	
NETLOGO 6.0	43
3.1. <i>Software</i> dan Bahasa Pemrograman.....	45
3.2. NetLogo 6.0.....	47
3.3. Sample Model: Party	48
3.4. Tutorial #1 <i>Models</i>	55
BAB IV PENUTUP	65
DAFTAR PUSTAKA	69
GLOSARIUM.....	72
TENTANG PENULIS.....	73

DAFTAR GAMBAR

Gambar 1. Pemodelan Penyebaran Penyakit AIDS menggunakan ABMS	6
Gambar 2. Komponen Model Penyakit (Hunter et al., 2017)	11
Gambar 3. Struktur Agen (Macal & North, 2010)	23
Gambar 4. Topologi Interaksi Agen	24
Gambar 5. John Von Neumann (Wikipedia, 2020a)	26
Gambar 6. Implementasi Pertama Mesin Von Neumann (Wikipedia, 2020d)	26
Gambar 7. Stanislaw Ulam (Wikipedia, 2020b)	27
Gambar 8. Thomas Schelling (Wikipedia, 2020c)	29
Gambar 9. Langkah dan Tahapan Utama untuk Membangun ABMS (Salgado & Gilbert, 2013)	31
Gambar 10. Model Library pada NetLogo	49
Gambar 11. Interface Model Party	50
Gambar 12. Tolerance	51
Gambar 13. Pilihan dalam Model Library	56
Gambar 14. Interface Wolf Sheep Predation	58
Gambar 15. Switches and Sliders dalam Wolf Sheep Predation	59

DAFTAR TABEL

Tabel 1. Peta Ilmu Penelitian.....	14
Tabel 2. Elemen ODD Protocol.....	37
Tabel 3. Perbandingan Swarm, RePast, Mason, dan NetLogo (Gilbert, 2007).....	45

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan yang Maha Esa atas limpahan rahmat- Nya sehingga penulis dapat menyelesaikan buku *Agent-Based Modelling And Simulation* dengan Netlogo 6.0.

Buku ini berisi informasi tentang teknologi dan informasi terkait *Agent-Based Modelling And Simulation* (ABMS) dengan menggunakan NetLogo 6.0. ABMS adalah salah satu teknik komputasi, yang tumbuh seiring dengan pematangan teknologi komputer. Munculnya komputasi yang kuat telah membawa perubahan dramatis di banyak bidang kehidupan, termasuk perubahan signifikan dalam praktik dan konten sains. Ketika akses ke komputasi yang kuat meningkat (dan biayanya menurun), para ilmuwan mampu melakukan penghitungan dan simulasi yang sebelumnya tidak mungkin dilakukan.

Harapan penulis, buku ini dapat memberikan informasi dalam bidang simulasi. Tidak lupa juga penulis mengucapkan terimakasih kepada banyak pihak yang telah terlibat dan membantu dalam penyusunan dan penyelesaian buku ini. Kepada KEMENDIKBUD dan LPPM UPN “Veteran” Yogyakarta juga kami mengucapkan terimakasih karena telah memberikan dana Hibah Penelitian Dasar, sehingga kegiatan penelitian serta penyusunan buku ini dapat terlaksana.

Demikian buku ajar ini disusun dan penulis menyadari masih banyak kekurangan yang ada pada buku ini, saran dan masukan yang membangun dari pembaca adalah upaya penulis dalam perbaikan dan penyempurnaan buku ini kedepannya.

Yogyakarta, September 2020
Salam Hormat,

Penulis



BAB I
PENDAHULUAN

1.1 Latar Belakang

Buku ini berisi pengantar metodologi pemodelan berbasis agen dan bagaimana metode tersebut itu dapat membantu manusia untuk lebih memahami dunia alam dan sosial serta rekayasa solusi untuk masalah sosial. Sebelum membahas tentang mengapa pemodelan berbasis agen penting, akan dijelaskan secara singkat apa itu pemodelan berbasis agen. Pemodelan dan simulasi berbasis agen atau Agent-Based Modelling and Simulation (ABMS) adalah sebuah pendekatan baru yang secara relatif memodelkan sistem yang kompleks dan terdiri dari interaksi, 'agen' yang otonom. Agen memiliki perilaku, sering dijelaskan dengan aturan sederhana, dan interaksi dengan agen lain, yang pada gilirannya mempengaruhi perilakunya. Agen biasanya juga memiliki komponen grafis yang dapat dilihat pada layar komputer. Agen dapat mewakili elemen apa pun dari suatu sistem, misalnya sebuah gas agen molekuler yang memiliki sifat seperti "massa" bernilai 30 atom satuan massa, "kecepatan" dengan nilai 10 meter per detik, dan "heading" dengan nilai sudut menghadapnya. Contoh lainnya adalah agen domba, yang memiliki properti seperti "*speed*" with nilai 3 mph, berat dengan nilai 30 lbs., dan bulu domba dengan nilai "penuh" (diskrit-tekstual bukan nilai numerik). Selain itu agen juga memiliki aturan tingkah laku. Agen molekuler gas mungkin memiliki aturan untuk bertabrakan dengan molekul lain; seekor agen domba mungkin memiliki aturan untuk memakan rumput jika ada rumput yang tersedia di dekatnya.

ABMS dapat dianalogikan sebagai jam universal. Saat jam terus berdetak, semua agen menjalankan aturan mereka. Jika kondisi peraturan terpenuhi, (misal, Mereka berada di tepi kotak, atau rumput di dekatnya), mereka memberlakukan perilaku (yaitu, memantul atau memakan rumput). Tujuan dari pemodelan berbasis agen adalah untuk membuat agen dan aturan yang akan menghasilkan target perilaku. Terkadang aturannya tidak terlalu dikenal, atau hanya ingin mempelajari perilaku sistem. Dalam hal ini, ABMS dapat digunakan untuk lebih memahami fenomena melalui eksperimen dengan aturan dan properti.

ABMS adalah teknologi representasi transformatif yang memungkinkan untuk lebih memahami topik yang sudah dikenal, dan

pada usia yang lebih muda; memahami dan menganalisis topik yang sampai sekarang belum dijelajahi; dan memungkinkan demokratisasi akses ke alat komputasi untuk memahami kompleksitas dan perubahan. Dengan demikian, mengembangkan literasi ABMS adalah keterampilan profesional dan keterampilan kehidupan yang kuat untuk siswa, dan kita harus mengupayakan literasi ABMS universal untuk semua kalangan, dari siswa hingga profesional. Dalam dua dekade terakhir, ilmuwan semakin banyak menggunakan metode pemodelan berbasis agen untuk melakukan mereka penelitian.

ABMS adalah salah satu teknik komputasi, yang tumbuh seiring dengan pematangan teknologi komputer. Munculnya komputasi yang kuat telah membawa perubahan dramatis di banyak bidang kehidupan, termasuk perubahan signifikan dalam praktik dan konten sains. Ketika akses ke komputasi yang kuat meningkat (dan biayanya menurun), para ilmuwan mampu melakukan penghitungan dan simulasi yang sebelumnya tidak mungkin dilakukan. Peningkatan daya komputasi dan konektivitas ini juga memungkinkan pengumpulan dan analisis kumpulan data yang sangat besar. Kumpulan data ini sering kali menyertakan data pada tingkat skala mikro, memungkinkan untuk mengekstraksi lebih banyak wawasan tentang bagaimana individu dalam masyarakat berperilaku, hewan dalam ekosistem bertahan hidup, atau elemen dari sistem rekayasa mempengaruhi satu sama lain. Kombinasi data besar, komputasi murah, dan konektivitas tinggi memungkinkan model berbasis agen dilakukan dengan jutaan agen individu yang properti dan perilakunya telah terbentuk dan divalidasi. Selain itu, representasi komputasi yang bersifat dinamis dan dapat dieksekusi, memungkinkan untuk interaktivitas yang lebih besar antara pengguna dan representasi. Dan yang lebih penting, representasi berbasis agen memiliki keunggulan tertentu karena mudah digunakan oleh orang lain memahami.

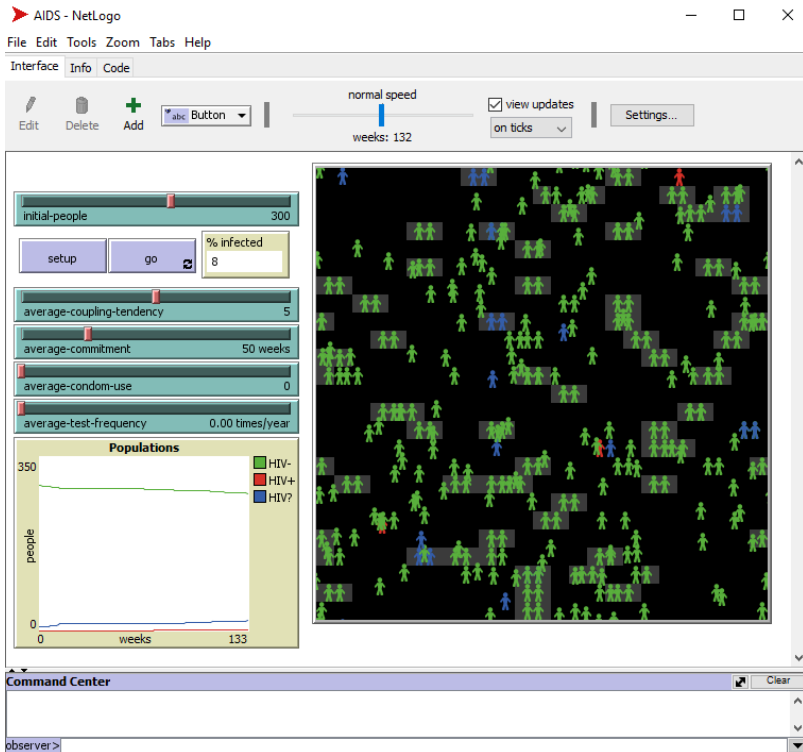
Representasi berbasis agen lebih mudah dipahami daripada representasi matematis dari fenomena yang sama. Hal ini terjadi karena model berbasis agen dibuat dari objek individu dan aturan sederhana untuk pergerakan perilakunya, sebagai lawan dari model persamaan yang dibangun dari simbol matematika. Namun, sebagian besar perubahan dramatis dalam representasi dan praktik ilmu pengetahuan

belum menghasilkan perubahan yang signifikan dalam dunia pendidikan. Ada banyak alasan lambatnya perubahan dalam "transfer teknologi" ke sistem pendidikan. Kendalanya adalah kurangnya pemahaman yang luas tentang manfaat transfer teknologi semacam itu.

Bahasa pemrograman yang digunakan dalam buku ini adalah NetLogo, dikembangkan oleh Uri Wilensky. Bahasa pemrograman ini dirancang supaya pemula dapat dengan cepat menggunakannya untuk melakukan hal-hal yang bermakna dan berguna namun juga dapat digunakan oleh ilmuwan dan peneliti untuk merancang model ilmiah mutakhir. Bahasa ini juga meminjam banyak sintaksnya dari bahasa Logo, yang dirancang agar dapat diakses oleh anak-anak. Seperti Logo, NetLogo menyebut agen sebagai *turtle*. Namun, sementara di Logo, file pengguna mengarahkan *turtle* untuk menggambar bentuk geometris, di NetLogo, ini digeneralisasikan untuk ribuan *turtle*. Alih-alih menggambar dengan pena, *turtle* biasanya menggambar dengan tubuh mereka, bergerak sesuai aturan, dan konfigurasi tubuh mereka menghadirkan visualisasi dari fenomena yang dimodelkan. NetLogo pertama kali dikembangkan pada akhir 1990-an, dan sekarang telah digunakan oleh ratusan ribu pengguna di seluruh dunia. Ribuan makalah ilmiah telah memanfaatkan NetLogo untuk membangun dan mengeksplorasi model dalam berbagai disiplin ilmu. Dengan adanya buku ini diharapkan dapat memberikan pengetahuan tentang ABMS dan Bahasa NetLogo secara komprehensif. Contoh pemodelan dengan ABMS menggunakan software Netlogo dapat dilihat pada Gambar 1.

1.2 Sistem Kompleks dan Kemunculannya

Apa yang dianggap sulit memiliki dimensi kognitif, tetapi kesulitan juga sangat dipengaruhi oleh kebutuhan manusia saat ini. Ketika perdagangan berkembang pada abad pertengahan, muncullah sebuah kebutuhan akan aritmatika dengan angka yang besar, sehingga kesulitan melakukannya dengan angka romawi menjadi lebih menonjol. Saat ini, dunia telah menjadi semakin kompleks, sebagian karena, pada periode-periode awal sejarah, tidak perlu terlalu diperhatikan interaksi yang kompleks; manusia dapat bertahan dengan memahami sistem sederhana dan efek lokal. Namun, seiring kemajuan teknologi dan sains, manusia menjadi lebih terpengaruh oleh interaksi



Gambar 1. Pemodelan Penyebaran Penyakit AIDS menggunakan ABMS

yang kompleks. Contohnya, perubahan pada hutan hujan di Brasil dapat berdampak dramatis pada iklim negara yang letaknya jauh dari Brasil; keputusan keuangan yang tidak bijaksana di satu negara dapat berdampak secara signifikan bagi perekonomian di seluruh dunia; satu kasus penyakit baru di Wuhan, Cina dapat menyebar ke seluruh dunia dalam waktu singkat; atau video berdurasi empat menit yang diunggah oleh seorang bintang pop Korea dapat membuatnya menjadi sensasi dunia dalam hitungan hari. Dengan demikian, kesulitan memahami sistem yang kompleks menjadi lebih menonjol.

Namun, meskipun tingkat kerumitan dalam hidup tetap konstan selama berabad-abad, pencarian pengetahuan yang terus-menerus pada akhirnya akan menuntun manusia untuk mempelajari sistem yang kompleks. Model dinamika populasi sederhana, misalnya, membuat

asumsi implisit bahwa semua anggota suatu spesies adalah sama, tetapi kemudian, menjadi penting untuk mengeksplorasi berbagai kompleksitas jaring makanan dan bagaimana setiap individu berinteraksi dengan setiap individu lainnya. Dengan demikian, perlu dipahami sistem yang lebih kompleks juga dimana merupakan hasil alami dari pertumbuhan pengetahuan manusia.

Saat manusia memperoleh pengetahuan, pasti akan menciptakan alat yang lebih canggih dan alat ini memungkinkan manusia untuk menemukan masalah baru dan membuat solusi penyelesaiannya. Seperti yang dijelaskan sebelumnya, munculnya komputasi yang kuat memungkinkan manusia untuk membuat model, mensimulasikan, dan menyelidiki lebih dalam sistem yang kompleks.

Untuk alasan yang telah disebutkan, bidang sistem yang kompleks telah muncul dan berkembang. Teori sistem kompleks mengembangkan prinsip dan alat untuk memahami kompleksitas dunia dan mendefinisikan sistem yang kompleks sebagai sistem yang terdiri dari beberapa elemen individu yang berinteraksi satu sama lain namun properti atau perilakunya tidak dapat diprediksi dari elemen itu sendiri. Melalui interaksi dari beberapa pendistribusian elemen terjadilah sebuah "fenomena yang muncul". Fenomena kemunculan adalah karakteristik dari sistem yang kompleks. Kemunculan didefinisikan sebagai munculnya struktur, pola, dan properti baru dan koheren melalui interaksi dari beberapa elemen terdistribusi. Struktur yang muncul tidak dapat disimpulkan hanya dari properti elemen, tetapi juga muncul dari interaksi elemen. Struktur yang muncul adalah properti sistem namun sering kali merupakan umpan balik kepada elemen individu yang menyusunnya.

Dalam sistem yang kompleks, keteraturan dapat muncul tanpa desain atau desainer apa pun. Ide tentang keteraturan tanpa desain telah menjadi kontroversi sepanjang sejarah sains. Di zaman modern, ketidakmungkinan keteraturan tanpa desain telah menjadi kunci utama gerakan desain cerdas melawan evolusi naturalistik, sebagai pendukung berpendapat itu. Namun, teori sistem kompleks menemukan sistem yang lebih rumit yang mungkin pada awalnya tampak tidak dapat direduksi tetapi ditemukan terorganisir sendiri atau berevolusi daripada dirancang dengan cerdas oleh seorang desainer.

1.3 Aplikasi ABMS

Memodelkan perilaku manusia sama sekali tidak jelas karena manusia tidak random, manusia memiliki pengetahuan dan kemampuan yang beragam, manusia dikendalikan oleh pengambilan keputusan yang rasional, serta perilaku manusia yang emosional (Kennedy, 2011). Penanggulangan bencana juga membutuhkan pengambilan keputusan yang rasional dan cepat. Fase respon atau fase tanggap darurat merupakan fase penting dalam penanggulangan bencana. Fase ini membutuhkan suatu sistem koordinasi relawan dan sistem yang terintegritasi dengan baik dalam pengelolaannya. Setiap kesalahan dalam sistem pada fase ini akan berimbas pada peningkatan signifikan dari jumlah korban maupun kerugian materil yang ditimbulkan. Oleh karena itu, Khair (no date) membuat sebuah penelitian yang bertujuan untuk mengevaluasi sistem koordinasi sentralisasi dan desentralisasi penanggulangan bencana erupsi Merapi. Penelitian ini mengintegrasikan pendekatan ABMS dan *Geographic Information System* (GIS) dalam memodelkan simulasi evaluasi sistem koordinasi relawan dengan beberapa skenario. ABMS bertujuan untuk menggambarkan perilaku *agents* (pengungsi dan relawan) saat terjadi bencana dengan karakteristik masing-masing. Sedangkan data spasial GIS berguna untuk menggambarkan kondisi jalanan secara nyata untuk kawasan Kabupaten Sleman. Hasil simulasi menunjukkan skenario alternative desentralisasi yang mengkombinasikan 4 Posko koordinasi menunjukkan penanganan dan penyaluran bantuan menuju barak pengungsian lebih baik dibandingkan dengan scenario awal (sentralisasi).

Penelitian yang dilakukan oleh Ramadhan and Nugraha (2014) terkait industri makanan yang memiliki kompleksitas, berupa banyaknya komponen yang terlibat, interdependensi antar komponen yang rumit termasuk pengaruh umpan balik komponen, adanya unsur ketidakpastian, dan proses yang dinamis, telah dilakukan. Kompleksitas ini mengindikasikan diperlukannya suatu model yang komprehensif untuk digunakan dalam menganalisis kebijakan. Salah satu metode yang dapat digunakan dalam menganalisis permasalahan sistem yang kompleks adalah pemodelan dan simulasi berbasis agen. Metode ini dapat memperlihatkan karakteristik dan perilaku dari setiap pelaku

sistem yang berdampak pada sistem secara keseluruhan. Setelah dilakukan verifikasi dan beberapa pengujian (validasi), dapat disimpulkan bahwa model yang dibuat valid dan dapat menggambarkan perilaku yang sesuai dengan kondisi nyata. Model ini bermanfaat dan dapat digunakan oleh pihak-pihak yang memiliki kepentingan dalam pengembangan suatu kawasan industri kuliner seperti pemerintah dan pelaku usaha kuliner dalam menganalisis suatu kebijakan yang berkaitan dengan aktivitas sistem industri kuliner.

Susiloningtyas et al. (2015) telah mengembangkan pemodelan berbasis perilaku di bidang perikanan. Perikanan cumi-cumi di Pulau Salura dibentuk oleh adanya agen nelayan andon, penduduk lokal, serta sumberdaya cumi-cumi. Kehadiran nelayan andon yang melakukan migrasi musiman dari Tanjung Luar di Pulau Lombok sampai ke Pulau Salura dengan aktivitas penangkapan cumi-cumi menjadikan sistem perikanan cumi-cumi ini menjadi berbeda dengan daerah lain. Intensitas migrasi dari nelayan andon yang tinggi berpengaruh terhadap perilaku penduduk lokal, serta kelimpahan sumberdaya cumi-cumi yang ada. Tujuan dari penelitian ini adalah mengkaji model pengelolaan aktivitas migrasi musiman dengan pengaturan alat tangkap yang dibawa. Metode yang digunakan adalah dengan ABMS melalui perangkat lunak Netlogo 6.2. Hasil penelitian menunjukkan bahwa pengaturan jumlah unit kapal yang terlibat dalam penangkapan cumi-cumi dalam waktu satu tahun harus diatur.

Pemodelan berbasis perilaku dibidang ekonomi juga telah dikembangkan oleh Worldailmi and Ismianti (2020). Faktor kebijakan ekonomi negara, yang ketersediaan berbagai pembayaran non tunai, dan penjualan dan pembelian online dapat mendorong kecenderungan untuk menggunakan transaksi non tunai (e-payment). Salah satu cara untuk mengetahui tren ini adalah dengan menggunakan model. Model dapat membantu memahami dan menjelaskan fenomena nyata lebih mudah dan efisien daripada mengamati langsung. Dalam penelitian tersebut, faktor-faktor yang mempengaruhi tren untuk menggunakan e-payment diperoleh dari berbagai referensi. Dari hasil simulasi, jumlah pengguna e-payment mengikuti pola Scurve. Polanya naik dan turun terus, lalu naik tajam, dan kemudian naik dan turun secara stabil lagi.

Dari analisis sensitivitas, keamanan yang dirasakan, harapan kinerja, dan harapan usaha memiliki peran penting dalam tren ini.

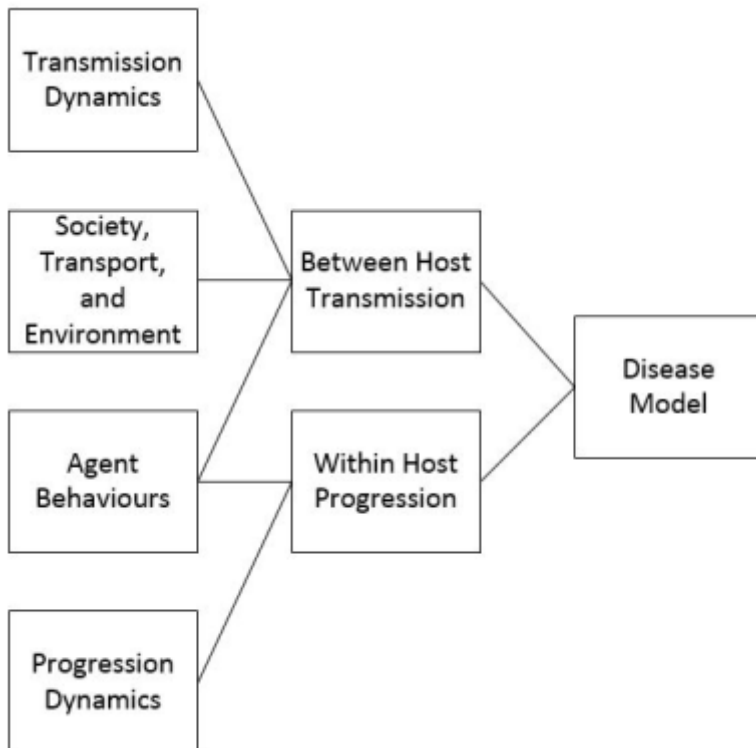
Model merupakan suatu representasi dari sebuah sistem nyata atau sistem yang sedang berlangsung dalam kehidupan. Sedangkan pemodelan adalah proses pembentukan sebuah model dari suatu sistem nyata. Pemodelan dapat digunakan untuk menentukan tujuan dan fungsi utama dari sebuah sistem, untuk memahami karakteristik dari pemodelan sistem yang digunakan, supaya dapat menentukan model sistem yang akan digunakan dalam membangun sebuah sistem, dan supaya dapat menganalisa kebutuhan user dalam membuat sebuah model sistem. Banyak sekali kasus dalam kehidupan sehari-hari yang dapat dipecahkan dengan pemodelan, salah satunya adalah pemodelan tentang penyebaran penyakit.

Nuraini, Khairudin, and Apri (2020) telah melakukan simulasi pemodelan penyebaran Covid-19 di Indonesia berdasarkan data endemik awal. Model ini didasarkan pada Kurva Richard yang mewakili persamaan logistik yang dimodifikasi. Berdasarkan model yang telah dibangun, didapat hasil bahwa endemik Covid-19 akan berakhir pada April 2020 dengan jumlah total kasus lebih dari 8000.

Ivorra et al. (2020) telah mengembangkan model matematika untuk penyebaran Covid-19. Model yang dikembangkan adalah model baru dari θ -SEIHRD (bukan SIR, SEIR atau model tujuan umum lainnya), yang memperhitungkan ciri khusus penyakit ini, seperti adanya infeksi yang tidak terdeteksi dan sanitasi yang berbeda dan kondisi infeksi orang yang dirawat di rumah sakit. Model ini juga mampu memperkirakan kebutuhan tempat tidur di rumah sakit. Penelitian ini mempelajari kasus khusus Tiongkok (termasuk Cina Daratan, Makau, Hong Kong dan Taiwan, seperti yang dilakukan oleh WHO dalam laporannya tentang Covid-19), negara yang menyebarkan penyakit, dan menggunakan laporannya untuk mengidentifikasi parameter model, yang dapat digunakan untuk memperkirakan penyebaran Covid-19 negara lainnya.

Agent-Based Modelling and Simulation (ABMS) merupakan simulasi berbasis komputer untuk memodelkan semua perilaku entitas (agen) yang terlibat dalam dunia nyata dengan harapan interaksi antar entitas dapat menghasilkan atau menggambarkan sifat utama yang

dapat digunakan lagi sebagai alat bantu untuk eksplanatori atau prediksi dalam mengambil keputusan di dunia nyata (Macal & North, 2010). ABMS juga dapat dikatakan sebagai alat yang dapat digunakan untuk lebih memahami dinamika suatu wabah penyakit menular, karena dapat dipengaruhi oleh banyak faktor termasuk vaksinasi atau tingkat kekebalan, kepadatan populasi, dan struktur usia populasi (Hunter et al., 2018). Terdapat empat komponen utama dari ABMS epidemiologi yaitu penyakit, masyarakat, transportasi, dan lingkungan (Hunter et al., 2017). Ketika membuat model berbasis agen untuk epidemiologi penyakit menular, pertimbangan tentang bagaimana penyakit menular ditularkan antara agen dan bagaimana penyakit berkembang pada agen yang terinfeksi harus dilakukan. Komponen model penyakit telah digambarkan pada Gambar 2.



Gambar 2. Komponen Model Penyakit (Hunter et al., 2017)

Sebuah penelitian telah dilakukan untuk menyimulasikan wabah campak yang terjadi di Schull, Irlandia pada tahun 2012 (Hunter et al., 2018). Wabah tersebut disimulasikan di 33 kota yang berbeda dan melihat korelasi antara hasil model dan karakteristik kota (populasi, luas, tingkat vaksinasi, dan struktur umur) untuk menentukan apakah hasil model dipengaruhi oleh interaksi karakteristik kota dan keputusan tentang agen dalam model. Hasil menunjukkan bahwa wabah tidak berkorelasi kuat dengan karakteristik utama.

Mobilitas manusia adalah elemen kunci dalam pemahaman penyebaran epidemi. Hal ini berarti bahwa ketepatan dalam memodelkan mobilitas manusia sangat penting untuk mempelajari penularan penyakit menular dalam skala besar dan meningkatkan pengendalian epidemi. Penelitian yang dilakukan oleh Hackl dan Dubernet (2019) digunakan untuk mempelajari wabah influenza musiman di wilayah metropolitan Zurich, Swiss. Pengamatan model berbasis agen dibandingkan dengan hasil dari classical SIR models. Model ini direpresentasikan dengan prototipe yang dapat digunakan untuk menganalisis beberapa skenario pada kasus penyebaran penyakit di skala perkotaan, dengan mempertimbangkan pengaturan variasi parameter model yang berbeda. Hasil simulasi ini dapat membantu meningkatkan pemahaman tentang dinamika penyebaran penyakit dan untuk mengambil langkah-langkah yang lebih baik menuju pencegahan dan pengendalian epidemi.

Laskowski et al. (2011) mengembangkan kerangka kerja pemodelan berbasis agen untuk mensimulasikan penyebaran infeksi virus influenza pada tata letak berdasarkan perwakilan darurat rumah sakit di Winnipeg, Kanada. Penelitian ini melengkapi teknik pemodelan matematika untuk penyebaran penyakit, serta aplikasi pemodelan yang berfokus pada penyebaran infeksi nosokomial yang kebal antibiotik di rumah sakit. Pendekatan pemodelan berbasis agen mewakili pemodelan sistem, di mana departemen darurat dimodelkan sebagai kumpulan agen (pasien dan petugas layanan kesehatan) dan karakteristik, perilaku, dan interaksi mereka masing-masing. Hasil menunjukkan bahwa dalam konteks contoh yang diberikan, kebijakan pengendalian infeksi berorientasi pasien (pengobatan alternatif aliran, menutupi

pasien yang bergejala) cenderung memiliki efek yang lebih besar daripada kebijakan yang menargetkan petugas layanan kesehatan.

Pemodelan simulasi penyebaran penyakit tuberkulosis berbasis sistem agen telah dilakukan oleh Saputra, Irawan and Kusuma (2018). Berbagai faktor mempengaruhi penyebaran penyakit ini diantaranya adalah kontak dengan orang yang terinfeksi, jarak antar orang sehat dan orang yang terinfeksi, faktor lingkungan, suhu ruangan, serta daya tahan tubuh manusia itu sendiri. Pemodelan dan simulasi penyebaran penyakit dengan menggunakan Agent Based Model (ABM) adalah salah satu cara agar dapat mengetahui bagaimana penyakit tuberkulosis menyebar melalui orang yang terinfeksi penyakit kepada orang lain. Simulasikan perkembangan penyakit Tuberkulosis ini dikembangkan melalui beberapa faktor yaitu lingkungan, manusia sehat dan manusia yang terinfeksi.

1.4 Kontribusi ABMS terhadap Pandemi Covid-19

Memodelkan perilaku manusia terhadap penyebaran Covid-19 menggunakan ABMS adalah tantangan nyata karena empat komponen utama, yaitu penyakit, masyarakat, transportasi, dan lingkungan, harus dipertimbangkan. Penelitian terkait pemodelan ABMS efek perilaku individu terhadap penyebaran Covid-19 belum banyak dilakukan. Pemodelan berbasis ABMS merupakan *tool* yang fleksibel dan dapat dibuat untuk mencerminkan setiap lingkungan tertentu, ABMS juga merupakan alat pendukung keputusan bagi para praktisi dan pembuat kebijakan untuk menilai dampak relatif dari penyebaran Covid-19. Penelitian tersebut jelas memiliki kontribusi yang nyata dibidang pendidikan dan medis supaya dapat mengetahui bagaimana dampak dari perilaku individu terhadap penyebaran Covid-19 dan kebijakan pemerintah dalam mengatur perilaku individu terhadap penyebaran Covid-19.

Tabel 1. Peta Ilmu Penelitian

	Kurva Richard		θ -SEIHRD	Classical <i>SIR Models</i>	ABMS	GIS
Penanggulangan Bencana					Memodelkan simulasi evaluasi sistem koordinasi relawan dengan beberapa skenario (Khair, no date)	
Makanan					Model simulasi yang menggambarkan tingkat pertumbuhan industri kuliner di suatu Kawasan (Ramadhan & Nugraha, 2014)	
Perikanan					Model pengelolaan aktivitas migrasi musiman dengan pengaturan alat tangkap yang dibawa (Susiloningtyas et al., 2015)	
Ekonomi					Pemodelan tren penggunaan <i>e-payment</i> (Worldailmi & Ismianti, 2020).	
Kesehatan	Covid-19	Simulasi pemodelan penyebaran Covid-19 di Indonesia (Nuraini et al., 2020)	Model matematika untuk penyebaran Covid-19		Kontribusi ABMS	

	Kurva Richard	θ-SEIHRD	<i>Classical SIR Models</i>	ABMS	GIS
			(Ivorra et al., 2020)		
	Campak			Pemodelan wabah campak yang terjadi di Schull, Irlandia pada tahun 2012(Hunter et al., 2018)	
	Influenza			Mempelajari wabah influenza musiman di wilayah metropolitan Zurich, Swiss (Hackl & Dubernet, 2019).	
				Mensimulasikan penyebaran infeksi virus influenza pada tata letak berdasarkan perwakilan darurat rumah sakit di Winnipeg, Kanada (Laskowski et al., 2011).	
	Tuberkulosis			Pemodelan simulasi penyebaran penyakit tuberkulosis berbasis sistem agen (Saputra, Irawan and Kusuma, 2018).	



BAB II
TEORI

2.1. Pengertian ABMS

ABMS memiliki koneksi ke banyak bidang lain termasuk ilmu kompleksitas, ilmu sistem, dinamika sistem, ilmu pengetahuan komputer, ilmu manajemen, beberapa cabang ilmu sosial, serta pemodelan dan simulasi konvensional. ABMS mengacu pada bidang-bidang ini untuk landasan teoretisnya, pandangan dan filosofi dunia untuk landasan konseptualnya, dan teknik untuk pemodelan yang dapat diterapkan. ABMS sangat berelasi dengan bidang sistem multi-agen (MAS) dan robotika dari bidang kecerdasan buatan (AI), serta Artificial Life (ALife). Tapi ABMS tidak hanya terikat pada pemahaman dan perancangan agen "artifisial". Dengan ini, muncul kebutuhan untuk merepresentasikan perilaku, interaksi sosial, kolaborasi, perilaku kelompok, dan kemungkinan kemunculannya.

ABMS dapat ditelusuri ke investigasi ke dalam sistem yang kompleks (Weisbuch, 1991), sistem adaptif yang kompleks (Holland, 1995; Kauffman, 1993), dan kehidupan buatan (Langton, 1989), dikenal sebagai ALife (Macal, 2009) untuk review dari pengaruh investigasi kehidupan buatan pada perkembangan pemodelan berbasis agen. Kompleks sistem terdiri dari interaksi, komponen otonom; sistem adaptif yang kompleks memiliki kemampuan tambahan bagi agen untuk beradaptasi di tingkat individu atau populasi. Investigasi kolektif ini ke dalam sistem yang kompleks dicari untuk mengidentifikasi prinsip-prinsip universal dari sistem tersebut, seperti dasar untuk organisasi diri, fenomena yang muncul, dan asal mula adaptasi di alam.

Berdasarkan definisi simulasi konvensional, istilah ABMS dalam buku ini merujuk pada simulasi berbasis agen, di mana proses dinamis dan bergantung pada waktu yang dimodelkan, dan jenis pemodelan berbasis agen yang lebih umum dimana mencakup model yang dirancang untuk dilakukan optimasi. Model berbasis agen yang khas memiliki tiga elemen:

1. Seperangkat agen (*agents*), atribut (*attributes*) dan perilaku mereka (*behaviours*).
2. Seperangkat hubungan (*relationship*) agen dan metode interaksi: Sebuah topologi yang mendasari keterhubungan dan mendefinisikan bagaimana dan dengan siapa agen berinteraksi.

3. Lingkungan (*environment*) agen: Agen berinteraksi dengan mereka lingkungan selain agen lainnya.

2.2.1. Agen Otonom (*Autonomous Agents*)

Karakteristik tunggal yang paling penting dari sebuah agen adalah kemampuannya untuk bertindak secara otonom, yaitu bertindak berdasarkan kemampuannya sendiri tanpa arahan eksternal dalam menanggapi situasi sebuah pertemuan (Macal & North, 2010). Agen dibekali dengan perilaku yang memungkinkan mereka untuk membuat keputusan independen. Biasanya, agen bersifat aktif, memulai tindakan mereka untuk mencapai tujuan internalnya, daripada hanya pasif, menanggapi agen dan lingkungan lainnya secara reaktif.

Tidak ada kesepakatan universal dalam literatur tentang definisi yang tepat tentang agen. Jennings (2000) memberikan definisi agen berdasarkan ilmu computer yaitu menekankan pada karakteristik esensial dari perilaku otonom. Beberapa penulis menganggap apapun jenis komponen independen (perangkat lunak, model, individu, dll) untuk menjadi agen (Bonabeau, 2001). Dalam pandangan ini, sebuah perilaku komponen dapat berkisar dari yang sederhana dan reaktif terhadap aturan '*if-then*' untuk perilaku kompleks yang dimodelkan oleh teknik kecerdasan buatan adaptif. Penulis lain bersikeras bahwa sebuah agen haruslah adaptif, mampu untuk mempelajari dan mengubah perilakunya sebagai tanggapan terhadap pengalaman, untuk disebut agen. Casti (1997) mengemukakan bahwa agen harus berisi kedua aturan tingkat dasar untuk perilaku dan aturan tingkat yang lebih tinggi yang menerapkan 'aturan untuk mengubah aturan'. Aturan tingkat dasar memberikan respons yang lebih pasif terhadap lingkungan, sedangkan 'aturan untuk mengubah aturan' memberikan kemampuan yang lebih aktif dan adaptif.

Karakteristik agen dari sudut pandang pemodelan praktis, berdasarkan bagaimana dan mengapa model agen sebenarnya dibuat dan dijelaskan dalam aplikasi adalah (Macal & North, 2010):

- Agen adalah mandiri, modular, dan individu yang unik. Persyaratan modularitas menyiratkan bahwa agen memiliki batas. Bisa dengan mudah menentukan apakah sesuatu adalah bagian dari agen, bukan bagian dari agen, atau merupakan atribut bersama. Agen memiliki atribut yang membedakan dengan agen lain dan diakui oleh agen lain.
- Agen itu otonom dan mandiri. Sebuah agen dapat berfungsi secara mandiri di lingkungannya dan berinteraksi dengan agen lain. Sebuah agen memiliki perilaku yang berhubungan dengan informasi yang dirasakan oleh agen untuk keputusan dan tindakannya. Informasi agen datang melalui interaksi dengan agen lain dan dengan lingkungannya. Perilaku agen dapat ditentukan mulai dari aturan sederhana hingga model abstrak, seperti jaringan saraf atau program genetik.
- Seorang agen memiliki keadaan (*state*) yang bervariasi dari waktu ke waktu. Sama seperti sistem negara bagian yang terdiri dari kumpulan variable negaranya, agen juga memiliki status yang mewakili variabel penting dan terkait dengan situasi saat ini. Status agen terdiri dari satu set atau subset dari atributnya. Perilaku agen dikondisikan pada statusnya.
- Agen memiliki interaksi sosial yang dinamis dengan agen lain yang mempengaruhi perilakunya. Agen memiliki protocol untuk interaksi dengan agen lain, seperti untuk komunikasi, pergerakan dan perebutan ruang, kapabilitas menanggapi lingkungan, dan lain-lain. Agen memiliki kemampuan untuk

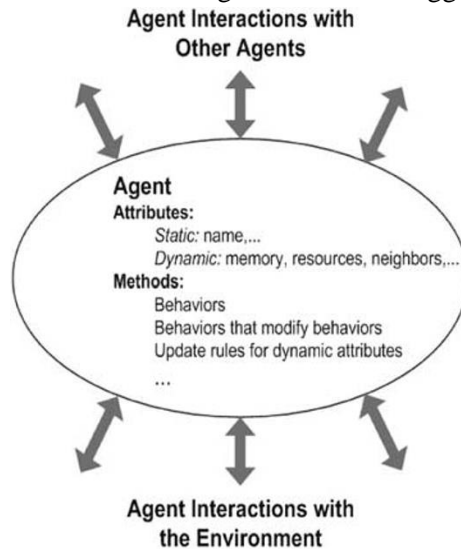
mengenali dan membedakan ciri-ciri agen lainnya.

Agen mungkin juga memiliki karakteristik berguna lainnya (Macal & North, 2010):

- Sebuah agen dapat bersifat adaptif, misalnya dengan aturan atau mekanisme yang lebih abstrak yang mengubah perilakunya. Sebuah agen mungkin memiliki kemampuan untuk mempelajari dan menyesuaikannya perilaku berdasarkan pengalaman yang terakumulasi.
- Sebuah agen dapat diarahkan pada suatu tujuan, memiliki tujuan untuk dicapai (tidak harus tujuan untuk memaksimalkan). Hal ini memungkinkan agen untuk membandingkan hasil dari perilakunya relatif terhadap tujuannya dan menyesuaikannya tanggapan dan perilaku dalam interaksi masa depan.
- Agen mungkin bersifat heterogen. Karakteristik agen dan perilaku dapat bervariasi dalam tingkat dan kecanggihannya, seberapa banyak informasi yang dipertimbangkan dalam membuat keputusan agen, model internal agen dari dunia luar, pandangan agen tentang kemungkinan reaksi dari agen lain sebagai tanggapan atas tindakannya, dan tingkat ingatan peristiwa di masa lalu yang disimpan dan digunakan untuk membuat keputusan. Agen juga mungkin dibekali dengan jumlah sumber daya yang berbeda atau mengakumulasi berbagai tingkat sumber daya sebagai hasil interaksi agen.

Struktur agen yang khas diilustrasikan pada Gambar 3. Dalam sebuah model berbasis agen, segala sesuatu yang berhubungan dengan agen adalah atribut agen atau metode agen yang beroperasi di agen. Atribut agen dapat menjadi statis, tidak dapat diubah selama simulasi, atau dinamis, dapat diubah selama

simulasi berlangsung. Misalnya, atribut statis adalah nama agen; atribut dinamis adalah memori interaksi agen di masa lalu. Metode agen mencakup perilaku, seperti aturan atau representasi yang lebih abstrak seperti jaringan neural, yang menghubungkan situasi agen dengan tindakan atau kumpulannya tindakan potensial. Contohnya adalah metode yang digunakan seorang agen digunakan untuk mengidentifikasi tetangganya.



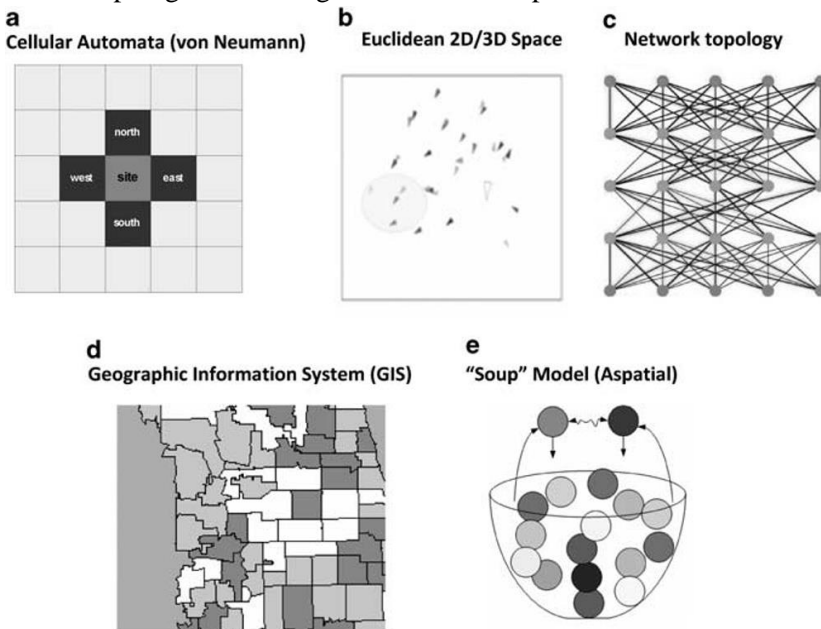
Gambar 3. Struktur Agen (Macal & North, 2010)

2.2.2. *Interacting Agents*

ABMS menyangkut pemodelan hubungan dan interaksi antara dirinya dengan agen lain. Dua masalah utama dalam memodelkan interaksi agen adalah menentukan siapa, terhubung kepada siapa, dan mekanisme interaksi yang dinamis. Kedua aspek tersebut harus dibenahi dalam pembangunan ABMS. Agen dapat berinteraksi dengan agen lain, tetapi tidak semua agen berinteraksi secara langsung dengan semua agen lainnya sepanjang waktu, seperti sistem di dunia nyata. Agen yang berinteraksi dengan bagian lain dari sebuah agen, disebut tetangga agen. Informasi lokal diperoleh dari interaksi dengan tetangga agen (bukan setiap agen atau semua agen) dan dari

lingkungan lokalnya (bukan dari bagian mana pun dari seluruh lingkungan). Umumnya, file kumpulan tetangga agen berubah dengan cepat sebagai simulasi hasil.

Bagaimana agen terhubung satu sama lain secara umum disebut topologi atau keterhubungan model berbasis agen. Topologi tipikal mencakup grid spasial atau jaringan *nodes* (*agents*) dan *links* (*relationships*). Topologi menggambarkan siapa yang mentransfer informasi dan kepada siapa informasi tersebut ditransfer. Di beberapa model, agen dapat berinteraksi dengan beberapa topologi. Misalnya, model pandemi berbasis agen baru-baru ini memiliki agen yang berinteraksi melalui *spatial grid* untuk memodelkan kontak fisik saat agen melakukan aktivitas sehari-hari dan kemungkinan menularkan infeksi. Topologi interaksi agen diilustrasikan pada Gambar 4.



Gambar 4. Topologi Interaksi Agen

2.2.3. Agent Environment

Agen dapat berinteraksi dengan lingkungan mereka dan agen lain. Lingkungan hanya dapat digunakan untuk menyediakan informasi tentang lokasi relatif spasial agen

terhadap agen lain atau mungkin menyediakan kumpulan geografis yang kaya informasi, seperti dalam GIS. Lokasi agen, disebut sebagai atribut dinamis, terkadang diperlukan untuk melacak agen saat bergerak melintasi lanskap, memperebutkan ruang, memperoleh sumber daya, dan menghadapi situasi lain. Model lingkungan yang kompleks dapat digunakan untuk memodelkan lingkungan dari agen. Misalnya model disperse hidrologi atau atmosfer dapat memberikan data spesifik lokasi titik pada tingkat air tanah atau polutan atmosfer, yang dapat diakses oleh agen. Dengan demikian, lingkungan mungkin dapat membatasi tindakan agen. Misalnya lingkungan dalam model transportasi berbasis agen akan mencakup infrastruktur dan kapasitas node dan link dari jaringan jalan raya. Kapasitas ini akan menimbulkan efek kemacetan (kecepatan perjalanan berkurang) dan membatasi jumlah agen yang bergerak melalui jaringan transportasi kapan saja.

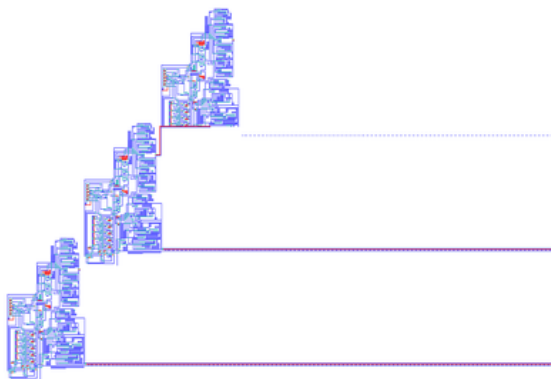
2.2. Sejarah ABMS

ABMS sebagian besar dimulai sebagai seperangkat ide, teknik, dan alat untuk mengimplementasikan model komputasi dari sistem adaptif yang kompleks. Banyak dari model berbasis agen awal dikembangkan menggunakan perangkat lunak pemodelan Swarm yang dirancang oleh Langton dan orang lain untuk memodelkan ALife (Minar et al, 1996). Awalnya, agen perilaku dimodelkan menggunakan aturan yang sangat sederhana yang masih menyebabkan perilaku muncul yang sangat kompleks. Dalam 10 tahun terakhir atau lebih, alat perangkat lunak pemodelan berbasis agen dan lingkungan pengembangan telah tersedia berkembang pesat baik dalam jumlah maupun kemampuan.



Gambar 5. John Von Neumann (Wikipedia, 2020a)

Ide ABMS dikembangkan sebagai konsep yang relatif sederhana di akhir 1940-an. Karena memerlukan prosedur komputasi yang intensif, maka ABMS tidak meluas sampai tahun 1990-an. Sejarah ABMS dapat ditelusuri dari mesin Von Neumann yaitu sebuah mesin teoritis yang mampu mereproduksi. Mesin yang diusulkan Von Neumann akan mengikuti instruksi yang sangat rinci untuk membuat salinannya sendiri.



Gambar 6. Implementasi Pertama Mesin Von Neumann (Wikipedia, 2020d)

Konsep mesin Von Neumann kemudian dibangun oleh teman Von Neumann, Stanislaw Ulam, juga seorang ahli matematika; Ulam menyarankan agar mesin dibuat di atas kertas, sebagai kumpulan sel di atas kisi. Ide tersebut membuat Von Neumann penasaran, yang membuatnya menciptakan perangkat pertama yang kemudian disebut *cellular automata*. Kemajuan lain diperkenalkan oleh matematikawan John Conway. Dia membangun Game of Life yang terkenal. Tidak seperti mesin von Neumann, Game of Life Conway dioperasikan dengan aturan sederhana di dunia virtual dalam bentuk papan catur 2 dimensi.

Bahasa pemrograman Simula, yang dikembangkan pada pertengahan 1960 dan diterapkan secara luas pada awal 1970, adalah kerangka kerja pertama untuk mengotomatiskan simulasi berbasis agen secara tahap demi tahap. Simula adalah nama dari dua bahasa pemrograman simulasi, yaitu Simula I dan Simula 67, yang dikembangkan pada tahun 1960 di Pusat Komputasi Norwegia di Oslo, oleh Ole-Johan Dahl dan Kristen Nygaard. Simula dianggap sebagai bahasa pemrograman berorientasi



Gambar 7. Stanislaw Ulam (Wikipedia, 2020b)

objek pertama. Seperti namanya, versi Simula pertama pada tahun 1962 dirancang untuk melakukan simulasi. Simula 67 meskipun dirancang

untuk menjadi bahasa pemrograman yang bertujuan umum dan juga dapat menyediakan kerangka kerja untuk banyak fitur bahasa berorientasi objek saat ini.

Salah satu model ABMS paling awal dalam konsep adalah model segregasi Thomas Schelling (Schelling, 1971). Meskipun Schelling awalnya menggunakan koin dan kertas grafik, tidak menggunakan komputer, model tersebut mewujudkan konsep dasar model ABMS sebagai agen otonom yang berinteraksi dalam lingkungan bersama dengan hasil agregat yang muncul dan diamati.

Pada awal 1980-an, Robert Axelrod menyelenggarakan turnamen strategi *Prisoner's Dilemma* dan meminta para peserta untuk berinteraksi dengan menggunakan ABMS sebagai cara penentuan pemenang. Axelrod kemudian mengembangkan banyak model ABMS lainnya di bidang ilmu politik yang meneliti fenomena dari etnosentrisme hingga penyebaran budaya (Axelrod, 1997). Pada akhir 1980-an, karya Craig Reynolds pada model berkelompok berkontribusi pada pengembangan beberapa model ABMS biologis pertama yang berisi karakteristik sosial. Reynolds mencoba memodelkan realitas agen biologis yang hidup, yang dikenal sebagai *artificial life*, istilah yang diciptakan oleh Christopher Langton.

Penemu pertama dari kata "*agent*" dan definisi yang sekarang digunakan saat ini sulit untuk dilacak. Salah satu kandidat yang mungkin adalah makalah milik John Holland dan John H. Miller yang telah dipresentasikan di suatu konferensi sebelumnya. Pada saat yang sama, selama 1980-an, ilmuwan sosial, matematikawan, peneliti operasi, dan beberapa orang dari disiplin lain mengembangkan *Computational and Mathematical Organization Theory* (CMOT). Bidang ini tumbuh sebagai kelompok minat khusus dari *The Institute of Management Sciences* (TIMS) dan *Operations Research Society of America* (ORSA).



Gambar 8. Thomas Schelling (Wikipedia, 2020c)

Tahun 1990-an sangat terkenal untuk perluasan ABMS dalam ilmu sosial, salah satu upaya penting adalah ABMS skala besar, Sugarscape, yang dikembangkan oleh Joshua M. Epstein dan Robert Axtell untuk mensimulasikan dan mengeksplorasi peran fenomena sosial seperti migrasi musiman, polusi, reproduksi seksual, pertempuran, dan penularan penyakit dan bahkan budaya (Epstein & Axtell, 1996). Perkembangan penting tahun 1990-an lainnya termasuk Kathleen Carley ABM dari Universitas Carnegie Mellon, untuk mengeksplorasi ko-evolusi jaringan sosial dan budaya. Selama jangka waktu 1990-an Nigel Gilbert menerbitkan buku teks pertama tentang Simulasi Sosial: Simulasi untuk ilmuwan sosial (1999) dan membuat jurnal dari perspektif ilmu sosial: *Journal of Artificial Societies and Social Simulation* (JASSS). Selain JASSS, ABMS dari disiplin apa pun berada dalam cakupan *SpringerOpen journal Complex Adaptive Systems Modeling* (CASM).

Sepanjang pertengahan 1990-an, rangkaian ilmu sosial ABM mulai berfokus pada isu-isu seperti merancang tim yang efektif, memahami komunikasi yang diperlukan untuk efektivitas organisasi, dan perilaku jejaring sosial. CMOT, kemudian berganti nama menjadi

Computational Analysis of Social and Organizational Systems (CASOS), menggabungkan semakin banyak pemodelan ABMS.

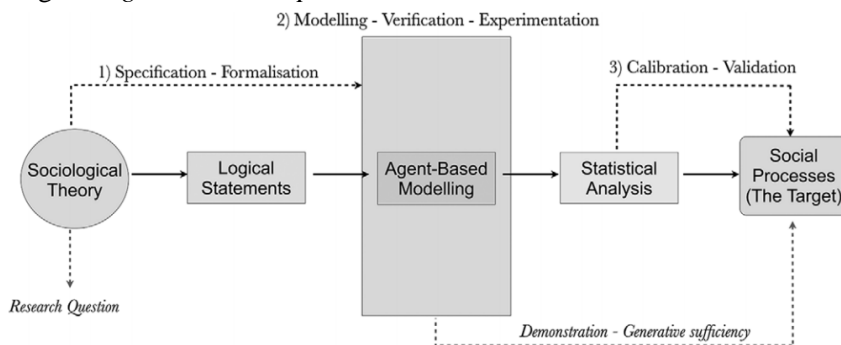
Pada akhir 1990-an, penggabungan TIMS dan ORSA yang membentuk INFORMS, dan perubahan frekuensi pertemuan INFORMS dari dua pertemuan setiap tahun menjadi satu, memacu kelompok CMOT untuk membentuk kelompok yang terpisah, *North American Association for Computational Social and Organizational Sciences* (NAACSOS). Kathleen Carley adalah kontributor utama, terkhusus untuk model jaringan sosial, ia memperoleh pendanaan *National Science Foundation* untuk konferensi tahunan dan menjabat sebagai Presiden pertama NAACSOS. Dia digantikan oleh David Sallach dari Universitas Chicago dan Laboratorium Nasional Argonne, dan kemudian oleh Michael Prietula dari Universitas Emory. Pada waktu yang hampir bersamaan NAACSOS dimulai, *European Social Simulation Association* (ESSA) dan *Pacific Asian Association for Agent-Based Approach in Social Systems Science* (PAAA), mitra dari NAACSOS yang diorganisir. Pada 2013, ketiga organisasi ini bekerja sama secara internasional. Kongres Dunia Pertama, pada bulan Agustus 2006, tentang Simulasi Sosial diadakan di bawah sponsor bersama di Kyoto, Jepang. Kongres Dunia Kedua diadakan di pinggiran utara Virginia di Washington, DC, pada bulan Juli 2008, dengan Universitas George Mason mengambil peran utama dalam pengaturan lokal.

Baru-baru ini, Ron Sun mengembangkan ABMS pada model kognisi manusia, yang dikenal sebagai simulasi sosial kognitif (Sun, 2006). Bill McKelvey, Suzanne Lohmann, Dario Nardi, Dwight Read dan lainnya di UCLA juga memberikan kontribusi yang signifikan dalam perilaku organisasi dan pengambilan keputusan. Sejak 2001, UCLA telah mengatur konferensi di Lake Arrowhead, California, yang telah menjadi titik pertemuan besar lainnya bagi para praktisi di bidang ini.

2.3. Karakteristik ABMS

Pada penelitian ABMS telah dikembangkan standar proses penelitian yang terdiri dari urutan langkah-langkah. Dalam praktiknya, beberapa langkah ini terjadi secara parallel dan keseluruhan proses sering dilakukan secara iteratif saat ide-ide disempurnakan dan

dikembangkan. Gambar 9 menunjukkan langkah-langkah utama yang harus diikuti peneliti dalam membangun ABMS (Salgado & Gilbert, 2013). Tahapan dimulai dimulai dari *sociological theory* dan diakhiri dengan *target* atau *social processes*.



Gambar 9. Langkah dan Tahapan Utama untuk Membangun ABMS (Salgado & Gilbert, 2013)

Di antara tahapan *sociological theory* dengan *target* atau *social processes*, terdapat serangkaian langkah yang harus dicapai untuk mengembangkan ABMS. Untuk mempermudah penyajian, telah diidentifikasi tiga tahap utama: 1) *Specification and formalisation*; 2) *Modelling, verification and experimentation*; dan 3) *Calibration and validation*. Tahap pertama adalah proses untuk menerjemahkan hipotesis teoritis yang menjelaskan proses sosial yang menarik, biasanya diekspresikan dalam bahasa alami, ke bahasa formal, menggunakan logika atau matematika. Tahap kedua meliputi pemodelan itu sendiri, di mana peneliti membangun dan memverifikasi model dengan cara eksperimental. Langkah ketiga meliputi kalibrasi model dengan data empiris dan validasi konsekuensi dengan uji statistik yang sesuai.

2.3.1. *Specification and Formalisation*

Mendefinisikan secara tepat sebuah *research question* atau pertanyaan, yang menjadi model akan dibahas pada tahap awal, merupakan hal yang penting untuk dilakukan. *Research question* atau pertanyaan yang khas dimana peneliti mencoba

untuk menjawabnya dengan menggunakan ABMS adalah pertanyaan yang menjelaskan bagaimana mengamati keteraturan dari interaksi *agent* level mikro yang dapat muncul di tingkat sosial atau makro.

Untuk mendefinisikan secara tepat *research question*, diperlukan juga model yang tertanam dalam teori sosial yang sudah ada. Mereview teori-teori yang ada berkaitan dengan *research question* adalah suatu hal yang penting untuk mengidentifikasi mekanisme penyebab sebuah model yang cenderung signifikan. Jadi, pemilihan teori juga penting untuk menyediakan mekanisme sebab-akibat yang paling masuk akal dan dapat diuji secara empiris. Dalam hal ini berarti bahwa, satu fitur penting dari ABMS adalah bahwa ABMS tidak memaksakan batasan apriori apa pun pada operasi mekanisme yang diasumsikan. ABM tidak didasarkan pada teori aksi atau interaksi yang spesifik.

Model merupakan penyederhanaan dari 'dunia nyata'. Fakta atau alasan mengapa ilmuwan ingin membangun sebuah model adalah karena ilmuwan ingin mengurangi kompleksitas dunia dan mengisolasi elemen-elemen utama yang menyebabkan fenomena tersebut. Untuk alasan ini, peneliti menggunakan ABMS yang bertujuan untuk menentukan mekanisme penyebab yang mendasari beberapa fenomena. Seperti yang telah kita bahas sebelumnya, di ABMS peneliti harus menentukan *agent* yang akan terlibat dalam model dan lingkungan tempat *agent* akan bertindak. Untuk setiap jenis *agent* dalam model, atribut dan aturan perilaku perlu ditentukan - yaitu, seperangkat aturan sederhana yang menentukan bagaimana *agent* berperilaku dan bereaksi terhadap lingkungan lokalnya. Atribut adalah karakteristik atau ciri *agent*, dan merupakan sesuatu yang membantu membedakan *agent* yang satu dengan yang lain dalam model dan tidak berubah, atau sesuatu yang berubah saat simulasi berjalan.

Begitu teori yang sesuai, dimana memberikan penjelasan kausal yang masuk akal tentang proses sosial target, telah diidentifikasi dan aturan perilaku telah ditentukan, peneliti

dilengkapi dengan hipotesis teoritis untuk diuji, dalam hal ini, *in silico*, yaitu menggunakan komputer dalam laboratorium untuk menjalankan eksperimen. Dalam ilmu sosial, hipotesis biasanya diekspresikan dalam bentuk tekstual atau menggunakan bahasa alami. Namun, bahasa alami biasanya ambigu atau multi tafsir dan konsepnya tidak selalu didefinisikan dengan ketat. Untuk itulah, ketika memiliki teori tentang bagaimana individu berperilaku dalam situasi yang sedang dianalisis, akan berguna untuk mengungkapkannya dalam bentuk sebuah prosedur atau bahasa formal atau buatan, menggunakan logika atau matematika. Sebuah keuntungan menggunakan logika adalah memberikan kondisi, misalnya hasil ini berlaku ketika dan jika kondisi berikut benar. Dengan demikian, peneliti dapat menggunakan aturan bersyarat seperti 'Jika C1, C2, dan C3, maka EP', di mana C mewakili beberapa kondisi dan EP adalah properti yang muncul atau makro yang ingin dijelaskan. Formalisasi semacam ini memfasilitasi tujuan akhir dalam ABM yaitu untuk memformalkan hipotesis teoritis dalam bentuk program komputer.

2.3.2. *Modelling, Verification, and Experimentation*

Setelah teori ditentukan dan diformalkan ke dalam logika atau matematika, pemodel dapat menerjemahkan formalisasi ini ke dalam program komputer. Sehingga model formal dapat diprogram dan dijalankan di komputer, dan perilaku simulasi dapat diamati dan diuji. Membangun sebuah model sangat mirip dengan membuat sebuah desain eksperimen. Model komputasi berbasis agen adalah alat baru untuk penelitian empiris. Ini menawarkan lingkungan alami untuk studi koneksionis fenomena dalam ilmu sosial. Pemodelan berbasis agen menyediakan cara yang ampuh untuk menjawab pertanyaan-pertanyaan tertentu yang multi disiplin. Hal tersebut memungkinkan seseorang untuk menundukkan teori inti tertentu. Tujuan akhir para peneliti yang menggunakan ABMS adalah untuk membuat akun pencapaian konfigurasi dengan sistem desentralisasi yang terbuat dari agen heterogen dan otonom.

Namun, saat menulis program komputer, terutama program yang rumit sangat umum membuat kesalahan. Proses pemeriksaan bahwa program melakukan apa yang telah direncanakan untuk dilakukan dikenal sebagai verifikasi. Dalam kasus ABMS, kesulitan verifikasi diperparah oleh fakta bahwa banyak simulasi termasuk generator nomor acak, yang berarti bahwa setiap proses berbeda dan bahwa hanya distribusi hasil yang dapat diantisipasi oleh teori. Oleh karena itu penting untuk 'men-debug' simulasi dengan hati-hati, sebaiknya menggunakan serangkaian pengujian kasus, mungkin situasi ekstrim di mana hasil mudah diprediksi dan menjalankan beberapa eksperimen untuk menjelajahi dan mengukur ruang parameter, kumpulan nilai parameter yang ditemukan dalam model tertentu.

2.3.3. Calibration and Validation

Setelah peneliti menentukan beberapa ABMS yang secara substansial masuk akal, dan model ini menghasilkan pola makro yang muncul dari minat, yang dapat digunakan sebagai data empiris untuk memperkirakan ukuran berbagai parameter yang tidak diketahui model ini dan kesesuaian antara data yang diprediksi dengan yang sebenarnya. Hal tersebut dicapai dengan mengkalibrasi dan memvalidasi model.

Verifikasi menyangkut apakah program bekerja sesuai dengan harapan peneliti, sedangkan validasi menyangkut apakah file simulasi adalah model yang baik dari sistem nyata. Cara umum untuk memvalidasi model adalah dengan membandingkan keluaran simulasi dengan data nyata yang dikumpulkan. Namun, ada beberapa peringatan yang harus diingat saat melakukan perbandingan ini. Misalnya, korespondensi yang tepat antara data aktual dengan data simulasi seharusnya tidak diharapkan. Jadi, peneliti harus memutuskan apa yang menyebabkan perbedaan antara kedua jenis data tersebut serta model tersebut dapat diterima atau tidak. Hal ini biasanya dilakukan dengan menggunakan beberapa analisis statistik untuk menguji signifikansi dari perbedaan data. Sedangkan *goodness-of-fit* selalu bisa ditingkatkan dengan menambahkan lebih banyak

faktor penjas, ada *trade-off* antara *goodness-of-fit* dan kesederhanaan. Terlalu banyak *fine-tuning* dapat mengakibatkan pengurangan daya penjas karena modelnya menjadi sulit untuk ditafsirkan. Paling ekstrim, jika model menjadi serumit dunia nyata, akan sama sulitnya untuk menafsirkan dan tidak menawarkan kekuatan penjas. Oleh karena itu, di sini ada paradoks yang tidak memiliki solusi yang jelas.

Banyak cara untuk dapat memvalidasi dan mengkalibrasi model ABMS, menurut Bianchi et al. (2007) validasi model berbasis agen ada tiga cara, yaitu:

- *Descriptive output validation*, atau mencocokkan output yang dihasilkan secara komputasi terhadap data yang sudah tersedia. Prosedur validasi semacam ini mungkin adalah cara yang paling intuitif dan dapat mewakili langkah fundamental menuju kalibrasi model yang baik.
- *Predictive output validation*, atau mencocokkan data yang dihasilkan secara komputasi dengan data sistem yang belum diperoleh. Jelas, masalah utama yang berkaitan dengan prosedur ini adalah penundaan antara hasil simulasi dan perbandingan terakhir dengan data aktual.
- *Input validation*, atau memastikan bahwa struktur fundamental, perilaku dan kondisi awal kelembagaan yang tergabung dalam model mereproduksi aspek utama dari sistem yang sebenarnya.

2.4. Framework ABMS

Pekerjaan terbaru pada pemodelan dan simulasi sistem adaptif kompleks telah menunjukkan kebutuhan untuk menggabungkan model berbasis agen dan model berbasis jaringan kompleks. Pendeskripsikan kerangka kerja yang terdiri dari empat tingkat pengembangan model sistem adaptif kompleks yaitu sebagai berikut (Aditya et al., 2013; Niazi, 2011):

1. *Complex Network Modeling Level* untuk mengembangkan model menggunakan data interaksi berbagai komponen sistem.
2. *Exploratory Agent-based Modeling Level* untuk mengembangkan model berbasis agen serta menilai kelayakan penelitian lebih lanjut. Pemodelan ini dapat berguna untuk mengembangkan model bukti konsep seperti untuk aplikasi pendanaan tanpa memerlukan kurva pembelajaran yang ekstensif bagi para peneliti.
3. *Descriptive Agent-based Modeling (DREAM)* untuk mengembangkan deskripsi model berbasis agen dengan menggunakan *template* dan model berbasis jaringan yang kompleks. Membangun model DREAM memungkinkan perbandingan model lintas disiplin ilmu.
4. Pemodelan berbasis agen yang divalidasi menggunakan *Virtual Overlay Multiagent System (VOMAS)* untuk pengembangan model yang diverifikasi dan divalidasi secara formal.

Metode lain untuk menggambarkan ABMS, termasuk *code template* dan metode berbasis teks, adalah protokol ODD (*Overview, Design Concept, Detail*). Ahli ekologi dan ilmuwan sosial telah lama dihadapkan pada tantangan bagaimana memodelkan kompleksitas yang melekat dalam banyak sistem ekologi, sosial, atau sosio-ekologi dunia nyata. Satu pendekatan untuk menjelajahi sistem semacam itu dapat menggunakan ABMS, namun pada awalnya pendokumentasian dianggap sangat buruk sehingga model tidak dapat dievaluasi (Lorek & Sonnenschein, 1999). Kritik ini memotivasi protokol ODD untuk membuat format generik dan struktur standar yang digunakan untuk mendokumentasikan semua ABM (Grimm et al., 2006). Tujuan utama protokol ODD adalah untuk membuat penulisan dan pembacaan deskripsi model lebih mudah dan lebih efisien. Selain itu, ODD diharapkan mampu mendeskripsikan model dengan lebih lengkap dan komprehensif, sehingga membuat ABMS lebih mudah untuk direplikasi dan dalam banyak sistem ekologi, sosial, atau sosio-ekologis. Tabel 2 berisi tujuh Protokol ODD yang disusun dengan

menjawab pertanyaan-pertanyaan yang terkait dengan setiap elemen (Grimm et al., 2006).

Tabel 2. Elemen ODD Protocol

<i>ODD Protocol</i>	Elemen	Sub-Elemen	Uraian
<i>Overview</i>	<i>Purpose</i>		Pada elemen ini akan dijelaskan tujuan pembuatan model.
	<i>Entities, state variables, scales</i>		<i>Entities</i> adalah siapa atau apa saja yang terlibat dalam model, yaitu <i>agent, spatial unit, environment, dan collectives</i> . <i>State variable</i> adalah peubah yang membedakan antar <i>entity</i> . <i>Scales</i> adalah perbandingan satuan dalam model dengan sistem nyata.
	<i>Process overview, scheduling</i>		<i>Process overview</i> menggambarkan perilaku dari <i>agent</i> . <i>Scheduling</i> adalah waktu terjadinya perilaku, perubahan atribut, dan bagaimana peubah waktu di dalam model (diskrit atau kontinu).
<i>Design Concept</i>	<i>Design concept</i>	<i>Basic principles</i>	Konsep dasar, teori, hipotesa, ataupun metode yang menjadi fondasi pembuatan model.
		<i>Emergence</i>	Keluaran yang paling penting dari model

<i>ODD Protocol</i>	Elemen	Sub-Elemen	Uraian
			sebagai hasil interaksi <i>agent</i> dengan <i>agent</i> lain dan lingkungannya.
		<i>Adaptation</i>	Kemampuan adaptasi <i>agent</i> , <i>rule</i> , atau aturan apa saja yang mendasari perubahan perilaku agen dan factor yang memicu.
		<i>Objectives</i>	Tujuan atau <i>goal</i> dari masing-masing <i>agent</i> .
		<i>Learning</i>	Kemampuan <i>agent</i> untuk merubah daya adaptasinya.
		<i>Prediction</i>	Kemampuan <i>agent</i> untuk memperkirakan apa yang akan terjadi.
		<i>Sensing</i>	Kemampuan <i>agent</i> untuk mempertimbangkan kondisi internalnya dan kondisi lingkungannya sebagai dasar untuk merubah perilakunya.
		<i>Interaction</i>	Interaksi apa saja yang terjadi dan apa dampak dari interaksi tersebut.
		<i>Stochasticity</i>	Unsur-unsur dalam model yang mengandung unsur ketidakpastian.
		<i>Collectives</i>	Kelompok yang dibentuk oleh individu atau <i>agent</i> yang memberi pengaruh pada perilaku <i>agent</i> atau

<i>ODD Protocol</i>	Elemen	Sub-Elemen	Uraian
			justru dipengaruhi oleh perilaku <i>agent</i> .
		<i>Observation</i>	Data apa saja yang dikumpulkan dari keluaran yang dihasilkan model dan bagaimana cara interpretasinya.
<i>Details</i>	<i>Initialization</i>		Kondisi awal pada saat model akan dijalankan.
	<i>Input Data</i>		Input data dari luar yang dibutuhkan oleh model.
	<i>Submodels</i>		Submodel yang digunakan untuk merepresentasikan urutan proses.

2.5. Aplikasi ABMS

ABMS telah digunakan dalam berbagai aplikasi yang mencakup ilmu fisika, biologi, ilmu sosial, dan manajemen. Aplikasi berkisar dari pemodelan peradaban kuno yang telah hilang ratusan tahun lalu, seperti bagaimana merancang model pasar baru yang saat ini tidak ada. Struktur model berbasis agen mencakup sebuah kontinum, dari model akademik minimalis hingga sistem pendukung keputusan skala besar. Model minimalis didasarkan pada satu set asumsi ideal, dirancang untuk hanya sebagian besar fitur yang menonjol dari suatu sistem. Model pendukung keputusan cenderung melayani aplikasi skala besar, dirancang untuk menjawab pertanyaan kebijakan dunia nyata, menyertakan data nyata, dan telah lulus tes validasi yang sesuai untuk membangun kredibilitas.

Pemodelan berbasis agen telah digunakan secara luas dalam biologi, termasuk analisis penyebaran epidemi, dan ancaman *biowarfare*, aplikasi biologis termasuk dinamika populasi, ekspresi gen stokastik, interaksi tumbuhan-hewan, ekologi vegetasi, keanekaragaman lanskap, pertumbuhan dan penurunan peradaban

kuno, evolusi perilaku etnosentris, perpindahan paksa atau migrasi, dinamika pilihan bahasa, pemodelan kognitif, dan aplikasi biomedis termasuk pemodelan 3D pembentukan jaringan payudara / morfogenesis, efek radiasi pengion pada dinamika subpopulasi sel induk mammae, peradangan, dan sistem kekebalan manusia. ABMS juga telah digunakan untuk mengembangkan sistem pendukung keputusan seperti untuk kanker payudara. ABMS ini juga semakin banyak digunakan untuk memodelkan sistem farmakologis pada tahap awal dan penelitian pra-klinis untuk membantu pengembangan obat dan mendapatkan wawasan tentang sistem biologis yang tidak mungkin dilakukan secara apriori. Selain itu, model berbasis agen baru-baru ini digunakan untuk mempelajari sistem biologi tingkat molekuler.

Model berbasis agen telah digunakan sejak pertengahan 1990-an untuk menyelesaikan berbagai masalah bisnis dan teknologi. Contoh aplikasi termasuk pemasaran, perilaku organisasi dan kognisi, kerja tim, optimasi rantai pasok dan logistik, pemodelan perilaku konsumen, termasuk dari mulut ke mulut, efek jaringan sosial, komputasi terdistribusi, manajemen tenaga kerja, dan manajemen portofolio. ABMS juga telah digunakan untuk menganalisis kemacetan lalu lintas. Baru-baru ini, pemodelan dan simulasi berbasis agen telah diterapkan ke berbagai domain seperti mempelajari dampak tempat publikasi oleh para peneliti dalam domain ilmu komputer (jurnal versus konferensi). Selain itu, ABMS telah digunakan untuk mensimulasikan pengiriman informasi dalam lingkungan bantuan ambien. Kemudian ABMS juga diaplikasikan untuk menganalisis simulasi postingan berbasis agen yang tersebar di jaringan sosial online Facebook. Dalam domain *peer-to-peer*, ad-hoc dan jaringan lain yang mengatur diri sendiri dan kompleks, kegunaan pemodelan dan simulasi berbasis agen telah ditunjukkan. Penggunaan kerangka spesifikasi formal berbasis ilmu komputer digabungkan dengan jaringan sensor nirkabel dan simulasi berbasis agen baru-baru ini telah dibuktikan. Algoritma atau pencarian evolusioner berbasis agen adalah topik penelitian baru untuk memecahkan masalah optimasi yang kompleks.

Sebelum, dan setelah krisis keuangan, minat telah tumbuh pada ABMS sebagai alat untuk menganalisis ekonomi. ABMS tidak berasumsi bahwa perekonomian dapat mencapai ekuilibrium dan "agen

perwakilan" digantikan oleh agen dengan perilaku yang beragam, dinamis, dan saling bergantung termasuk menggiring. ABMS mengambil pendekatan "*bottom-up*" dan dapat menghasilkan simulasi ekonomi yang sangat kompleks dan tidak stabil. ABMS dapat merepresentasikan sistem yang tidak stabil dengan *crash* dan *boom* yang berkembang dari respons non-linear (tidak proporsional) menjadi perubahan kecil yang proporsional

ABMS juga telah digunakan dalam arsitektur dan perencanaan kota untuk mengevaluasi desain dan untuk mensimulasikan arus pejalan kaki di lingkungan perkotaan, serta pemeriksaan aplikasi kebijakan publik untuk penggunaan lahan. Ada juga bidang analisis sosio-ekonomi yang sedang berkembang tentang dampak investasi infrastruktur menggunakan kemampuan ABM untuk membedakan dampak sistemik pada jaringan sosio-ekonomi.



BAB III
IMPLEMENTASI ABMS DENGAN
SOFTWARE NETLOGO 6.0

3.1. *Software* dan Bahasa Pemrograman

Setelah model berbasis agen telah diformalkan, keputusan penting adalah apakah lebih baik menulis program komputer khusus (menggunakan bahasa pemrograman seperti Java, C ++, C #, atau Visual Basic) atau menggunakan salah satu paket atau *toolkit* yang telah dibuat khusus untuk membantu dalam pengembangan simulasi. ABMS hampir selalu lebih mudah menggunakan *toolkit* daripada memulai lagi menulis program sendiri. Hal ini terjadi karena banyak terjadi masalah dan membutuhkan waktu yang lebih saat menulis program. Misalnya menulis kode untuk menunjukkan plot dan grafik dari awal dibutuhkan kemampuan yang terampil dan sangat memakan waktu, tetapi sebagian besar *toolkit* menyediakan berbagai fitur semacam fasilitas grafik untuk menampilkan variabel keluaran. Setidaknya beberapa dari *bug* dalam kode *toolkit* akan ditemukan oleh pengembang atau selanjutnya pengguna. Namun, dibalik keunggulan dari sebuah *toolkit* yang digunakan, terdapat beberapa kerugian yaitu fitur terbatas pada apa yang ditawarkan oleh *toolkit* tersebut.

Ada beberapa pilihan *software* atau *toolkit* untuk ABMS yang tersedia secara gratis dan olok untuk ilmuwan sosial. Tabel 3 memberikan perbandingan antara empat *software* pemodelan populer pada sejumlah kriteria (Gilbert, 2007). Pilihan *software* atau alat implementasi bergantung pada beberapa faktor, terutama keahlian seseorang dalam pemrograman dan kompleksitas serta skala model. NetLogo adalah merupakan *software* yang paling cepat untuk dipelajari dan paling mudah digunakan, tetapi mungkin bukan yang paling cocok untuk file berukuran besar dan model yang kompleks. Mason lebih cepat dari RePast, tetapi memiliki dasar pengguna yang jauh lebih kecil, artinya masih sedikit komunitas yang bisa memberikan nasehat dan dukungan.

Tabel 3. Perbandingan Swarm, RePast, Mason, dan NetLogo (Gilbert, 2007)

Aspek	Swarm	RePast	Mason	NetLogo
Lisensi	GPL (<i>General</i>)	GPL (<i>General</i>)	GPL (<i>General</i>)	Free, tetapi tidak <i>open source</i>

Aspek	Swarm	RePast	Mason	NetLogo
	<i>Public Lisence)</i>	<i>Public Lisence)</i>	<i>Public Lisence)</i>	
Dokumentasi	<i>Patchy</i>	Terbatas	Meningkat, tapi terbatas	Baik
<i>User base</i>	Berkurang	Besar	Meningkat	Besar
Bahasa pemodelan	Objective-C, Java	Java, Python	Java	NetLogo
Kecepatan eksekusi	Sedang	Cepat	Tercepat	Sedang
<i>Support for graphical user interface development</i>	Terbatas	Baik	Baik	Sangat mudah dibuat menggunakan 'Tunjuk dan klik'
Kemampuan bawaan untuk berkreasi film dan animasi	Tidak	Ya	Ya	Ya
Dukungan untuk sistematis eksperimen	Beberapa	Ya	Ya	Ya
Mudah dipelajari dan mudah untuk diprogram	Buruk	Sedang	Sedang	Baik
Mudah untuk diinstal	Buruk	Sedang	Sedang	Baik

Aspek	Swarm	RePast	Mason	NetLogo
<i>Link to geographical information system</i>	Tidak	Ya	Ya	Ya

NetLogo saat ini adalah *software* ABMS yang terbaik untuk pemula dan bahkan untuk banyak model ilmiah yang serius. NetLogo adalah turunan jauh dari bahasa pemrograman Logo yang dibuat pada tahun 1960-an. NetLogo masih mempertahankan beberapa aspek warisannya, misalnya, *turtle* yang dapat berpindah pindah. Bahasa pemrogramannya menggunakan sintaks yang sangat sederhana dan menyerupai bahasa Inggris, serta menyediakan bahasa pemrograman yang disederhanakan dan antarmuka grafis yang memungkinkan pengguna membangun, mengamati, dan menggunakan model berbasis agen tanpa perlu mempelajari Bahasa pemrograman yang rumit dan detail. Disamping itu, Netogo memiliki komunitas pengguna dan tim NetLogo di Northwestern University yang menyediakan kumpulan materi dokumentasi dan tutorial yang lengkap, berguna dan profesional.

3.2. NetLogo 6.0

NetLogo adalah platform simulasi dan pemodelan berbasis agen. Platform ini juga memungkinkan pembuatan model dinamika sistem dan simulasi partisipatif. Sangat cocok untuk keperluan penelitian maupun untuk berbagai keperluan pendidikan. Netlogo dapat digunakan untuk mengajarkan pemrograman, pemikiran komputasi, simulasi dan pembangunan model, dan pemahaman fenomena kompleks melalui model di banyak domain yang berbeda.

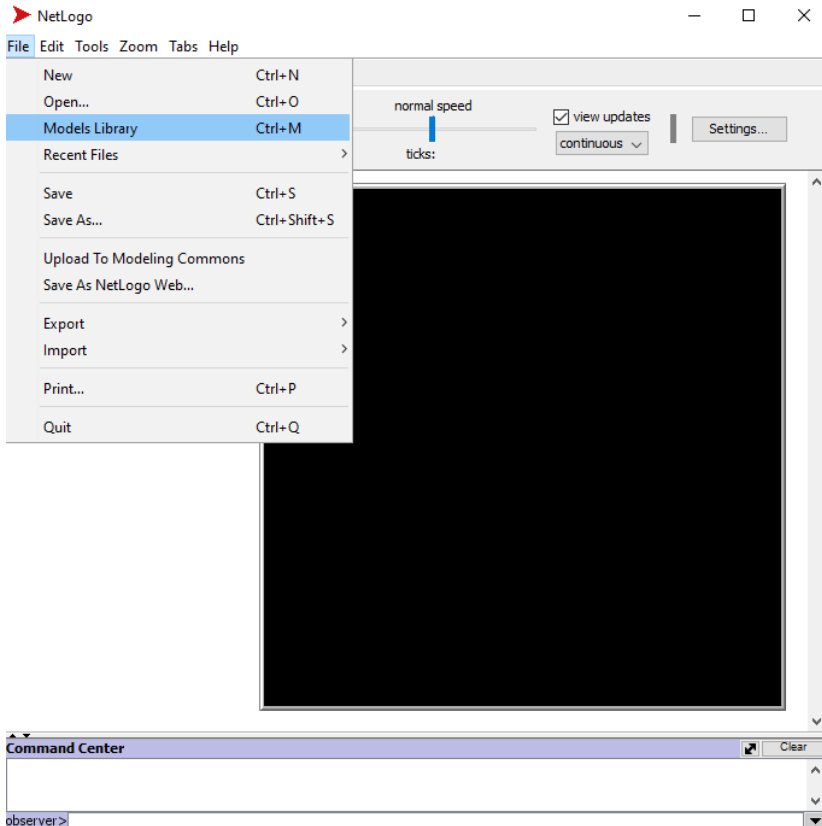
NetLogo dirancang oleh Uri Wilensky pada tahun 1999, dimana platform ini mengajarkan konsep pemrograman menggunakan agen dalam bentuk *turtles*, *patches*, *link* dan *observer*. NetLogo dirancang untuk berbagai audiens, khususnya: mengajar anak-anak dalam komunitas pendidikan, dan pakar domain tanpa latar belakang pemrograman untuk memodelkan fenomena terkait.

3.3. Sample Model: Party

Pernahkah Anda berada di sebuah pesta dan memperhatikan bagaimana orang-orang berkelompok? Anda mungkin juga memperhatikan bahwa orang tidak hanya tinggal dalam kelompok. Saat mereka beredar, kelompok berubah. Jika Anda mengamati perubahan ini dari waktu ke waktu, Anda mungkin memperhatikan polanya. Misalnya, dalam lingkungan sosial, orang mungkin menunjukkan perilaku yang berbeda ketika di tempat kerja atau rumah. Individu yang percaya diri dengan lingkungan kerjanya mungkin menjadi pemalu dan penakut pada sebuah perkumpulan sosial. Dan orang lain yang dicadangkan di tempat kerja mungkin menjadi "*party starter*" dengan teman-temannya. Pola-pola ini dapat bergantung pada jenis pertemuan. Di beberapa tempat, orang dilatih untuk melakukan pengorganisasian diri menjadi kelompok campuran; misalnya, permainan pesta atau kegiatan seperti sekolah. Tapi dalam suasana yang tidak terstruktur, orang cenderung mengelompokkan dengan cara yang lebih acak.

Apakah ada jenis pola untuk pengelompokan semacam ini? Mari kita lihat lebih dekat pertanyaan ini dengan menggunakan komputer untuk membuat model perilaku manusia di *party*. Model "*party*" NetLogo melihat secara khusus pada pertanyaan tentang pengelompokan berdasarkan jenis kelamin, mengapa kelompok cenderung terbentuk yang kebanyakan laki-laki, atau kebanyakan perempuan? Mari gunakan NetLogo untuk menjelajahi pertanyaan tersebut. Caranya adalah:

1. Buka *software* NetLogo
2. Pilih "*Model Library*" dari *File* menu.



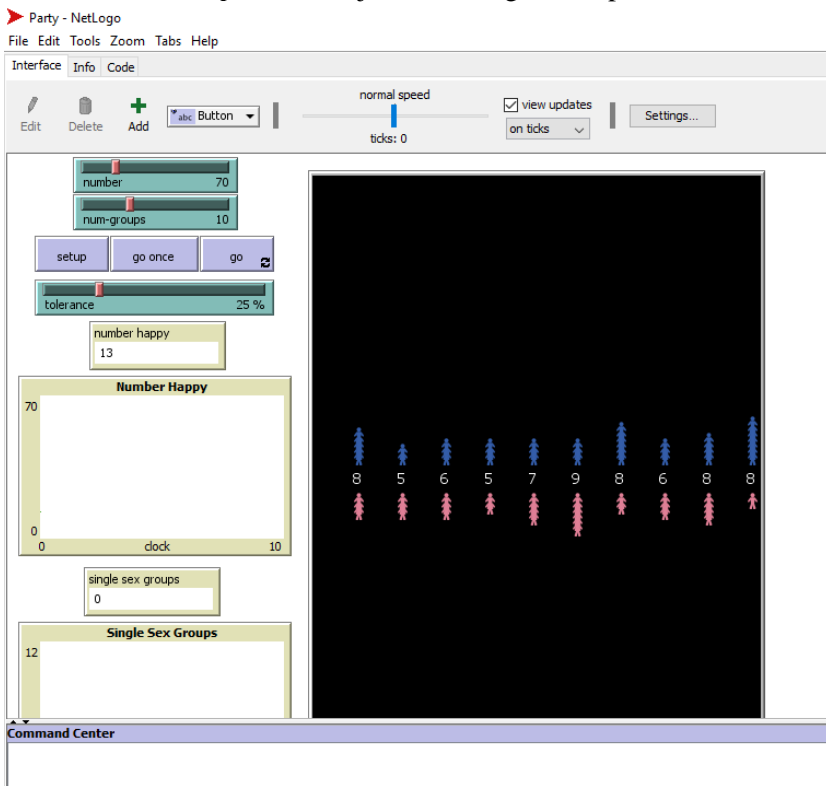
Gambar 10. Model Library pada NetLogo

3. Buka folder “*Social Science*”
4. Pilih model “*Party*”
5. Tekan tombol “*Open*”
6. Tekan tombol “*Setup*”

Pada tampilan model, Anda akan melihat grup merah muda dan biru dengan angka. Garis-garis ini mewakili kelompok yang berbaaur di sebuah pesta. Pria ditampilkan sebagai biru, wanita merah muda. Sedangkan angka yang tertera di situ adalah ukuran kelompok. Apakah semua kelompok memiliki jumlah orang yang hampir sama? Apakah semua kelompok memiliki jumlah yang hampir sama untuk setiap jenis kelamin? Misalkan Anda mengadakan pesta dan mengundang 150 orang. Anda bertanya-tanya bagaimana orang akan berkumpul Bersama

jika 10 kelompok terbentuk di pesta itu. Menurut Anda bagaimana mereka akan berkelompok? Alih-alih meminta 150 teman terdekat Anda untuk berkumpul dan mengelompokkan secara acak, mari kita mensimulasikan situasi ini dengan menggunakan komputer. Hal-hal yang perlu dilakukan adalah:

1. Tekan tombol "go". (Menekan tombol "go" lagi akan menghentikan model secara manual.)
2. Amati pergerakan orang sampai model berhenti.
3. Perhatikan plot untuk melihat apa yang terjadi dengan cara lain.
4. Gunakan *speed slider* jika Anda ingin memperlambat model.



Gambar 11. *Interface Model Party*

Sekarang berapa banyak orang di setiap kelompok? Awalnya, Anda mungkin mengira 150 orang yang terbagi menjadi 10 kelompok, akan menghasilkan sekitar 15 orang di setiap kelompok. Dari model,

kami melihat bahwa orang tidak terbagi rata menjadi 10 kelompok. Sebaliknya, beberapa kelompok menjadi sangat kecil, sedangkan kelompok lain menjadi sangat besar. Selain itu, pesta berubah seiring waktu dari semua kelompok campuran pria dan wanita menjadi semua jenis kelamin tunggal. Apa yang bisa menjelaskan ini? Ada banyak kemungkinan jawaban untuk pertanyaan ini tentang apa yang terjadi di pesta adalah hal yang nyata. Perancang simulasi ini berpendapat bahwa kelompok di pesta tidak terbentuk secara acak. Kelompok ditentukan oleh bagaimana individu di pesta tersebut berperilaku. Perancang model memilih untuk fokus pada suatu variabel tertentu, yang disebut "*tolerance*".



Gambar 12. *Tolerance*

Toleransi di sini didefinisikan sebagai persentase seberapa nyaman pengunjung pesta jika berkelompok dengan lawan jenis. Jika individu berada dalam kelompok yang memiliki persentase toleransi terhadap lawan jenis yang tinggi, maka individu tersebut menjadi "tidak nyaman" dan meninggalkan kelompok untuk mencari kelompok lain. Misalnya, jika tingkat toleransi ditetapkan pada 25%, maka laki-laki hanya "nyaman" dalam kelompok yang kurang dari 25% perempuan, dan perempuan hanya "nyaman" dalam kelompok yang kurang dari 25% laki-laki.

Sebagai individu menjadi "tidak nyaman" dan meninggalkan kelompok, mereka pindah ke kelompok baru, yang dapat menyebabkan beberapa orang dalam kelompok itu menjadi "tidak nyaman" pada gilirannya. Reaksi berantai ini berlanjut sampai semua orang di pesta merasa "nyaman" dalam kelompok mereka.

Perhatikan bahwa dalam model, "toleransi" tidak tetap. Anda, pengguna, dapat menggunakan "slider" toleransi coba persentase toleransi yang berbeda dan lihat apa hasilnya ketika Anda memulai model lagi.

Sintaks model *party* (Wilensky, 1997):

```
globals [  
  group-sites ;; agentset of patches where groups are  
  located  
  boring-groups ;; how many groups are currently  
  single-sex  
  ]  
  
  turtles-own [  
    happy? ;; true or false  
    my-group-site  
  ]  
  
  to setup  
    clear-all  
    set group-sites patches with [group-site?]  
    set-default-shape turtles "person"  
    create-turtles number [  
      choose-sex ;; become a man or a woman  
      set size 3 ;; be easier to see  
      set my-group-site one-of group-sites  
      move-to my-group-site  
    ]  
    ask turtles [ update-happiness ]  
    count-boring-groups  
    update-labels  
    ask turtles [ spread-out-vertically ]  
    reset-ticks  
  end  
  
  to go  
    if all? turtles [happy?]  
      [ stop ] ;; stop the simulation if everyone is happy  
    ask turtles [ move-to my-group-site ] ;; put all people  
  back to their group sites  
    ask turtles [ update-happiness ]  
    ask turtles [ leave-if-unhappy ]  
    find-new-groups  
    update-labels  
    count-boring-groups  
    ask turtles [
```

```

        set my-group-site patch-here
        spread-out-vertically
    ]
    tick
end

to update-happiness ;; turtle procedure
    let total count turtles-here
    let same count turtles-here with [color = [color] of
myself]
    let opposite (total - same)
    ;; you are happy if the proportion of people of the
opposite sex
    ;; does not exceed your tolerance
    set happy? (opposite / total) <= (tolerance / 100)
end

to leave-if-unhappy ;; turtle procedure
    if not happy? [
        set heading one-of [90 270] ;; randomly face right
or left
        fd 1                ;; leave old group
    ]
end

to find-new-groups
    display ;; force display update so we see animation
    let malcontents turtles with [not member? patch-here
group-sites]
    if not any? malcontents [ stop ]
    ask malcontents [ fd 1 ]
    find-new-groups
end

to-report group-site? ;; patch procedure
    ;; if your pycor is 0 and your pxcor is where a group
should be located,
    ;; then you're a group site.
    ;; In this model (0,0) is near the right edge, so pxcor
is usually
    ;; negative.

```

```

;; first figure out how many patches apart the groups
will be
let group-interval floor (world-width / num-groups)
report
  ;; all group sites are in the middle row
  (pycor = 0) and
  ;; leave a right margin of one patch, for legibility
  (pxcor <= 0) and
  ;; the distance between groups must divide evenly
into
  ;; our pxcor
  (pxcor mod group-interval = 0) and
  ;; finally, make sure we don't wind up with too
many groups
  (floor ((- pxcor) / group-interval) < num-groups)
end

to spread-out-vertically ;; turtle procedure
ifelse woman?
  [ set heading 180 ] ;; face south
  [ set heading 0 ] ;; face north
fd 4 ;; leave a gap
while [any? other turtles-here] [
  if-else can-move? 2 [
    fd 1
  ]
  [ ;; else, if we reached the edge of the screen
    set xcor xcor - 1 ;; take a step to the left
    set ycor 0 ;; and move to the base a new stack
    fd 4
  ]
]
end

to count-boring-groups
ask group-sites [
  ifelse boring?
    [ set plabel-color gray ]
    [ set plabel-color white ]
]

```

```

        set boring-groups count group-sites with [plabel-
color = gray]
    end

    to-report boring? ;; patch procedure
    ;; To see whether this group is single sex, we collect
the colors
    ;; of the turtles into a list, then remove all the
duplicates
    ;; from the list. If the result is a list with exactly one
color
    ;; in it, then the group is single sex.
    report length remove-duplicates ([color] of turtles-
here) = 1
    end

    to update-labels
    ask group-sites [ set plabel count turtles-here ]
    end

    ;;
    ;;; color procedures
    ;;

    ;; Blue represents male, pink represents female. No
stereotypes are meant
    ;; to be promoted. Simply change the colors right here
if you'd like.

    to choose-sex ;; turtle procedure
    set color one-of [pink blue]
    end

    to-report woman? ;; turtle procedure
    report color = pink
    end

```

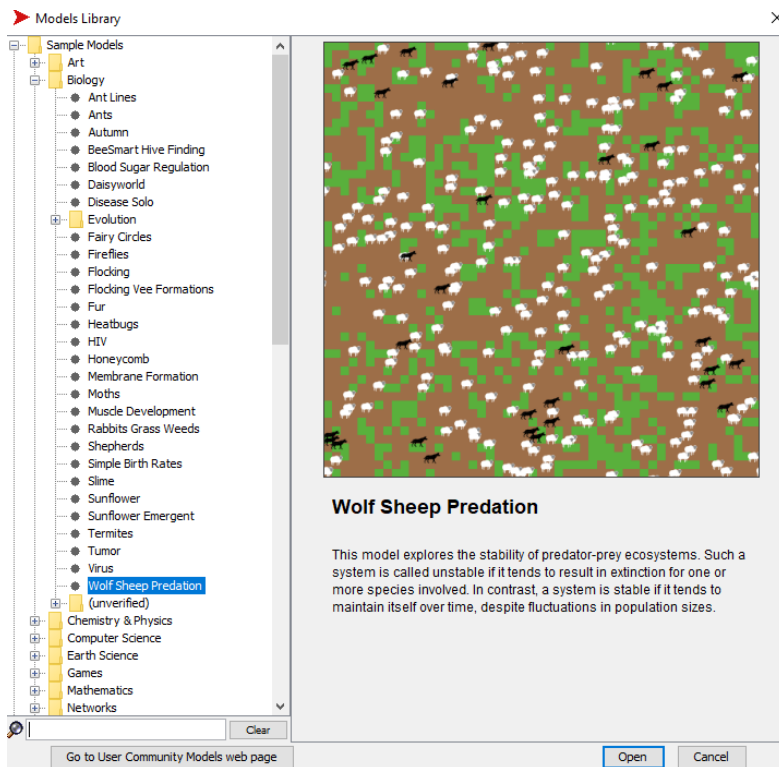
3.4. Tutorial #1 *Models*

Bagian ini akan membahas lebih dalam tentang fitur-fitur yang ada pada model di *Model Library*. Pada bagian ini akan dibahas

menangani model biologi: *wolf sheep predation*, sebuah model populasi predator-mangsa. Caranya adalah:

1. Buka *software* NetLogo
2. Pilih “*Model Library*” dari *File* menu.
3. Buka folder “*Biology*”
4. Pilih model “*Wolf Sheep Predation*”
5. Tekan tombol “*Open*”


Interface akan diisi dengan banyak buttons, switches, sliders and monitors. Elemen-elemen antarmuka tersebut memungkinkan Anda untuk berinteraksi dengan model. Tombol berwarna biru berfungsi untuk mengatur, memulai, dan hentikan modelnya. *Sliders* dan *switches* yang berwarna hijau berfungsi untuk mengubah pengaturan model. *Monitors* dan *plots* yang berwarna krem berfungsi untuk menampilkan data.




Gambar 13. Pilihan dalam *Model Library*

Jika Anda ingin memperbesar jendela agar lebih mudah dilihat, dapan menggunakan Menu zoom. Saat pertama kali membuka model, Anda akan melihat bahwa "view" (tampilan grafis dari file agen dalam model) kosong (semua hitam). Untuk memulai model, Anda harus menyiapkannya terlebih dahulu:

- Tekan tombol "setup"
- Tekan tombol "go" untuk memulai simulasi
- Tekan tombol "go" untuk menghentikan simulasi

Saat tombol ditekan, model merespons dengan tindakan. Tombol bisa menjadi "sekali" tombol, atau tombol "selamanya". Anda dapat membedakan kedua jenis tombol ini dengan simbol di bagian depan tombol. Tombol selamanya memiliki dua panah di kanan bawah sudut, seperti ini (). Sedangkan tombol sekali tidak memiliki panah,

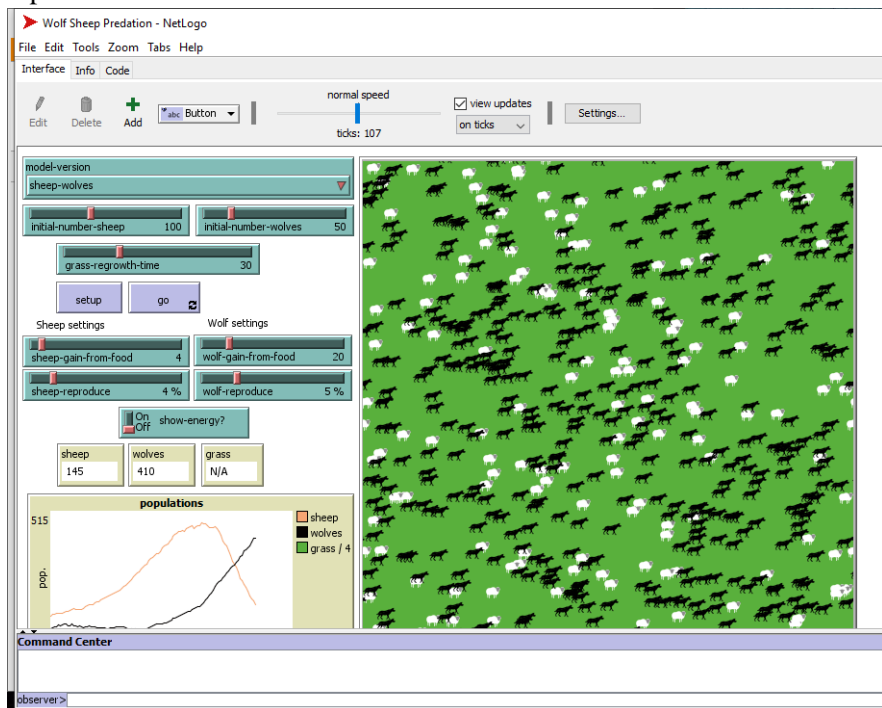
seperti ini (). Tombol ini hanya melakukan satu tindakan dan kemudian berhenti. Saat aksi selesai, tombol akan muncul kembali.

Tombol selamanya melakukan tindakan berulang kali. Saat Anda ingin aksi berhenti, tekan tombol tersebut lagi, maka akan menyelesaikan tindakan saat ini, lalu muncul kembali. Kebanyakan model, termasuk *Wolf Sheep Predation*, memiliki tombol sekali yang disebut "setup" dan tombol selamanya disebut "go". Banyak model juga memiliki tombol sekali yang disebut "go once" atau "step once" yang seperti "go" kecuali ia memajukan model dengan satu tick (langkah waktu), bukan lagi dan lagi. Menggunakan tombol sekali seperti ini memungkinkan Anda melihat kemajuan model lebih rapat.

Menghentikan tombol selamanya adalah cara normal untuk menjeda atau menghentikan model. Setelah berhenti, Anda bisa melanjutkan kembali dengan menekan tombol lagi.

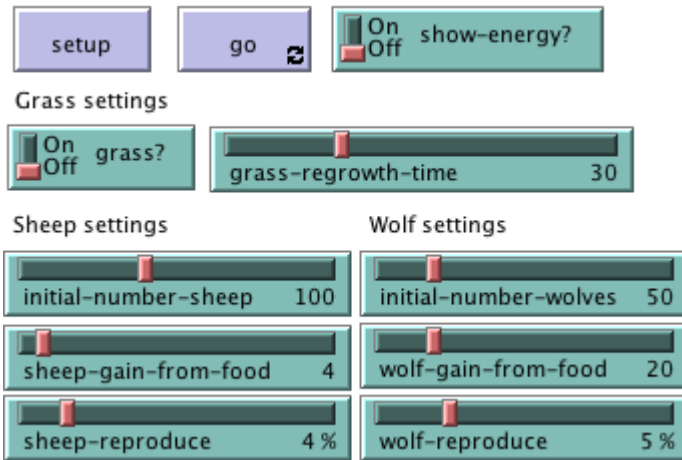
Speed slider memungkinkan Anda untuk mengontrol kecepatan model, yaitu kecepatan *turtles* bergerak, *patches* yang dapat berubah warna, dan seterusnya. Saat Anda memindahkan penggeser ke kiri, model melambat sehingga ada jeda lebih lama di antara setiap *tick* (langkah waktu). Hal ini membuatnya lebih mudah untuk melihat apa

yang sedang terjadi. Anda bahkan mungkin memperlambat model sejauh untuk melihat dengan tepat apa yang dilakukan seekor *turtle*. Saat Anda menggerakkan *speed slider* ke kanan, model akan dipercepat. NetLogo akan mulai melompati *frames*, yaitu, tampilan tidak akan diperbarui di akhir setiap centang, hanya beberapa centang. Memperbarui membutuhkan waktu, jadi lebih sedikit pembaruan tampilan berarti model berkembang lebih cepat. Perhatikan bahwa jika Anda menggeser *speed slider* ke kanan, tampilan mungkin jarang diperbarui



Gambar 14. Interface Wolf Sheep Predation

Setelan model memungkinkan Anda untuk menjelajahi berbagai skenario atau hipotesis. Mengubah pengaturan dan kemudian menjalankan model untuk melihat bagaimana reaksinya dapat memberi Anda pemahaman yang lebih dalam tentang fenomena yang dimodelkan. *Switches and sliders* memberi Anda akses ke setelan model.



Gambar 15. *Switches and Sliders dalam Wolf Sheep Predation*

Sintaks model *Wolf Sheep Predation* (Wilensky, 1997):

```

globals [ max-sheep ] ; don't let sheep population grow too large
; Sheep and wolves are both breeds of turtle.
breed [ sheep a-sheep ] ; sheep is its own plural, so we use "a-sheep"
as the singular.
breed [ wolves wolf ]
turtles-own [ energy ] ; both wolves and sheep have energy
patches-own [ countdown ]

to setup
  clear-all
  ifelse netlogo-web? [set max-sheep 10000] [set max-sheep 30000]

  ; Check model-version switch
  ; if we're not modeling grass, then the sheep don't need to eat to
  survive
  ; otherwise the grass's state of growth and growing logic need to be
  set up
  ifelse model-version = "sheep-wolves-grass" [
    ask patches [

```

```

set pcolor one-of [ green brown ]
ifelse pcolor = green
  [ set countdown grass-regrowth-time ]
  [ set countdown random grass-regrowth-time ] ; initialize grass
regrowth clocks randomly for brown patches
]
]
[
  ask patches [ set pcolor green ]
]

create-sheep initial-number-sheep ; create the sheep, then initialize
their variables
[
  set shape "sheep"
  set color white
  set size 1.5 ; easier to see
  set label-color blue - 2
  set energy random (2 * sheep-gain-from-food)
  setxy random-xcor random-ycor
]

create-wolves initial-number-wolves ; create the wolves, then
initialize their variables
[
  set shape "wolf"
  set color black
  set size 2 ; easier to see
  set energy random (2 * wolf-gain-from-food)
  setxy random-xcor random-ycor
]
display-labels
reset-ticks
end

to go

```

```

; stop the simulation of no wolves or sheep
if not any? turtles [ stop ]
; stop the model if there are no wolves and the number of sheep gets
very large
if not any? wolves and count sheep > max-sheep [ user-message
"The sheep have inherited the earth" stop ]
ask sheep [
  move
  if model-version = "sheep-wolves-grass" [ ; in this version, sheep
eat grass, grass grows and it costs sheep energy to move
    set energy energy - 1 ; deduct energy for sheep only if running
sheep-wolf-grass model version
    eat-grass ; sheep eat grass only if running sheep-wolf-grass
model version
    death ; sheep die from starvation only if running sheep-wolf-
grass model version
  ]
  reproduce-sheep ; sheep reproduce at random rate governed by
slider
]
ask wolves [
  move
  set energy energy - 1 ; wolves lose energy as they move
  eat-sheep ; wolves eat a sheep on their patch
  death ; wolves die if our of energy
  reproduce-wolves ; wolves reproduce at random rate governed by
slider
]
if model-version = "sheep-wolves-grass" [ ask patches [ grow-grass
] ]
; set grass count patches with [pcolor = green]
tick
display-labels
end

to move ; turtle procedure

```

```

rt random 50
lt random 50
fd 1
end

to eat-grass ; sheep procedure
; sheep eat grass, turn the patch brown
if pcolor = green [
  set pcolor brown
  set energy energy + sheep-gain-from-food ; sheep gain energy by
eating
]
end

to reproduce-sheep ; sheep procedure
if random-float 100 < sheep-reproduce [ ; throw "dice" to see if you
will reproduce
  set energy (energy / 2) ; divide energy between parent
and offspring
  hatch 1 [ rt random-float 360 fd 1 ] ; hatch an offspring and move
it forward 1 step
]
end

to reproduce-wolves ; wolf procedure
if random-float 100 < wolf-reproduce [ ; throw "dice" to see if you
will reproduce
  set energy (energy / 2) ; divide energy between parent and
offspring
  hatch 1 [ rt random-float 360 fd 1 ] ; hatch an offspring and move
it forward 1 step
]
end

to eat-sheep ; wolf procedure
let prey one-of sheep-here ; grab a random sheep

```

```

if prey != nobody [
    ; did we get one? if so,
    ask prey [ die ] ; kill it, and...
    set energy energy + wolf-gain-from-food ; get energy from
eating
]
end

to death ; turtle procedure (i.e. both wolf and sheep procedure)
; when energy dips below zero, die
if energy < 0 [ die ]
end

to grow-grass ; patch procedure
; countdown on brown patches: if reach 0, grow some grass
if pcolor = brown [
    ifelse countdown <= 0
    [ set pcolor green
      set countdown grass-regrowth-time ]
    [ set countdown countdown - 1 ]
]
end

to-report grass
ifelse model-version = "sheep-wolves-grass" [
    report patches with [pcolor = green]
]
[ report 0 ]
end

to display-labels
ask turtles [ set label "" ]
if show-energy? [
    ask wolves [ set label round energy ]
    if model-version = "sheep-wolves-grass" [ ask sheep [ set label
round energy ] ]
]

```

```
]
end
```



BAB IV
PENUTUP

Dalam buku ini, telah dijelaskan konsep utama ABMS dan relasi antara metode penelitian yang relatif baru dan penjelasan ilmiah di ilmu sosial. Kemudian juga telah dibahas tentang proses perancangan dan pembuatan model berbasis agen dan telah direkomendasikan serangkaian langkah standar yang dapat digunakan saat membangun model berbasis agen untuk penelitian ilmu sosial. Yang pertama, dan yang terpenting, langkah dalam proses pemodelan adalah mengidentifikasi tujuan model dan menjawab *research question*. Pentingnya menggunakan teori yang ada untuk membenarkan asumsi model dan untuk memvalidasi hasilnya juga telah ditekankan.

ABMS adalah pendekatan baru untuk sistem pemodelan yang terdiri dari agen otonom yang saling berinteraksi. Terdapat banyak aplikasi berbasis agen di berbagai bidang dan disiplin ilmu. ABMS terutama berlaku saat agen adaptasi dan *emergence* adalah pertimbangan penting.

Banyak perangkat lunak dan toolkit berbasis agen telah dikembangkan dan digunakan secara luas. Kombinasi dari beberapa faktor sinergis membuat ABMS bergerak maju dengan cepat. Faktor-faktor ini termasuk pengembangan spesialisasi metode ABMS dan *toolkit* yang berkelanjutan. Perbandingan berbagai macam *software* atau *toolkit* telah dijabarkan pada buku ini. NetLogo saat ini adalah *software* ABMS yang terbaik untuk pemula dan bahkan untuk banyak model ilmiah yang serius. NetLogo adalah turunan jauh dari bahasa pemrograman Logo yang dibuat pada tahun 1960-an. NetLogo masih mempertahankan beberapa aspek warisannya, misalnya, *turtle* yang dapat berpindah pindah. Bahasa pemrogramannya menggunakan sintaks yang sangat sederhana dan menyerupai bahasa Inggris, serta menyediakan bahasa pemrograman yang disederhanakan dan antarmuka grafis yang memungkinkan pengguna membangun, mengamati, dan menggunakan model berbasis agen tanpa perlu mempelajari bahasa pemrograman yang rumit dan detail. Disamping itu, NetLogo memiliki komunitas pengguna dan tim NetLogo di Northwestern University yang menyediakan kumpulan materi dokumentasi dan tutorial yang lengkap, berguna dan profesional.

ABMS memiliki dampak yang luas bagi masa depan tentang bagaimana bisnis menggunakan komputer untuk mendukung

pengambilan keputusan, pemerintah menggunakan model untuk membuat dan mendukung kebijakan, dan peneliti menggunakan laboratorium elektronik untuk lebih jauh penelitian mereka.

DAFTAR PUSTAKA

- Aditya, K., Kotobi, K., & Kesidis, G. (2013). An agent-based framework for performance modeling of an optimistic parallel discrete event simulator. *Complex Adaptive Systems Modeling*, 1(12). <https://doi.org/10.1186/2194-3206-1-12>
- Axelrod, R. (1997). *The Complexity of Cooperation: Agent-Based Models of Competition and Collaboration*. Princeton University Press.
- Bianchi, C., Cirillo, P., Gallegati, M., & Vagliasindi, P. (2007). Validating and calibrating agent-based models: A case study. *Computational Economics*, 30(3), 245–264.
- Bonabeau, E. (2001). Agent-based modeling: Methods and techniques for simulating human systems. *Proc Natl Acad Sci*, 99(3), 7280–7287.
- Casti, J. (1997). *Would-Be Worlds: How Simulation is Changing the World of Science*. Wiley.
- Epstein, J. M., & Axtell, R. (1996). *Growing artificial societies: social science from the bottom up*. Brookings Institution Press.
- Gilbert, N. (2007). *Agent-based models*. Sage Publications Ltd.
- Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., Goss-Custard, J., Grand, T., Heinz, S., Huse, G., Huth, A., Jepsen, J. U., Jørgensen, C., Mooij, W. M., Müller, B., Pe'er, G., Piou, C., Railsback, S. F., ... DeAngelis, D. L. (2006). A standard protocol for describing individual-based and agent-based models. *Ecol. Model*, 198, 115–126.
- Hackl, J., & Dubernet, T. (2019). Epidemic Spreading in Urban Areas Using Agent-Based Transportation Models. *Future Internet*, 1–14. <https://doi.org/10.3390/fi11040092>
- Holland, J. (1995). *Hidden Order: How Adaptation Builds Complexity*. Addison-Wesley.
- Hunter, E., Namee, B. Mac, & Kelleher, J. (2017). A Taxonomy for Agent-Based Models in Human Infectious Disease Epidemiology A Review of Epidemiological Agent-Based Models. *JASSS*, 20(3), 1–17.
- Hunter, E., Namee, B. Mac, & Kelleher, J. (2018). *An open-data-driven agent-based model to simulate infectious disease outbreaks*.
- Ivorra, B., Ferrández, M. R., Vela-pérez, M., & Ramos, A. M. (2020). *Mathematical modeling of the spread of the coronavirus disease 2019 (COVID-19) taking into account the undetected infections . The case of China. 2019.*

- Jennings, N. R. (2000). On agent-based software engineering. *Artif Intell*, 277 – 296.
- Kauffman, S. A. (1993). *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press.
- Kennedy, W. G. (2011). *Modelling Human Behaviour in Agent- Based Models How not to model human behaviour Levels of modelling human behaviour The science behind modelling human behaviour*. 167–179.
- Khair, F. (n.d.). *PENANGGULANGAN BENCANA MENGGUNAKAN GIS DAN AGENT BASED MODELING (ABM)*. XII(2), 249–261.
- Langton, C. G. (1989). Artificial Life. *Artificial Life: The Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems*, 6, 1–47. <https://www.unhcr.org/publications/manuals/4d9352319/unhcr-protection-training-manual-european-border-entry-officials-2-legal.html?query=excom> 1989
- Laskowski, M., Demianyk, B. C. P., Witt, J., Mukhi, S. N., Friesen, M. R., & McLeod, R. D. (2011). Agent-based modeling of the spread of influenza-like illness in an emergency department: A simulation study. *IEEE Transactions on Information Technology in Biomedicine*, 15(6), 877–889. <https://doi.org/10.1109/TITB.2011.2163414>
- Lorek, H., & Sonnenschein, M. (1999). Modelling and simulation software to support individual-oriented ecological modelling. *Ecol. Model*, 115, 199–216.
- Macal, C. M. (2009). Agent based modeling and artificial life. In *Encyclopedia of Complexity and Systems Science*. Springer.
- Macal, C. M., & North, M. J. (2010). *Tutorial on agent-based modelling and simulation*. 151–162. <https://doi.org/10.1057/jos.2010.3>
- Niazi, M. A. K. (2011). *Towards A Novel Unified Framework for Developing Formal, Network and Validated Agent-Based Simulation Models of Complex Adaptive Systems*.
- Nuraini, N., Khairudin, K., & Apri, M. (2020). *Modeling Simulation of COVID-19 in Indonesia based on Early Endemic Data*. 3(1), 1–8. <https://doi.org/10.5614/cbms.2020.3.1.1>
- Ramadhan, F., & Nugraha, C. (2014). *Pemodelan dan Simulasi Berbasis Agen untuk Sistem Industri Kuliner* (. 01(03), 101–113.
- Salgado, M., & Gilbert, N. (2013). Agent based modelling. In *Handbook of Quantitative Methods for Educational Research* (pp. 247–265). <https://doi.org/10.1007/978-94-6209-404-8>
- Saputra, G. W., Irawan, B., & Kusuma, P. D. (2018). MODELING

- AND SIMULATION THE SPREAD OF TUBERCULOSIS BASED ON AGENT SYSTEM. *ISSN : 2355-9365 e-Proceeding of Engineering*, 5(3), 6267–6275.
- Schelling, T. C. (1971). Dynamic Models of Segregation. *Journal of Mathematical Sociology*, 1(2), 143–186. <https://doi.org/10.1080/0022250x.1971.9989794>
- Sun, R. (2006). *Cognition and Multi-Agent Interaction: From Cognitive Modeling to Social Simulation*. Cambridge University Press.
- Susiloningtyas, D., Boer, M., Adrianto, L., & Yulianda, F. (2015). Agent Base Model For Seasonal Migration Activity of Squid Andon Fishers at Salura Island. *J. Sosek KP*, 10(1), 103–114.
- Weisbuch, G. (1991). *Complex Systems Dynamics: An Introduction to Automata Networks (translated from French by S. Ryckebusch)*. Addison-Wesley.
- Wikipedia. (2020a). *John von Neumann*. https://en.wikipedia.org/wiki/John_von_Neumann
- Wikipedia. (2020b). *Stanislaw Ulam*. https://en.wikipedia.org/wiki/Stanislaw_Ulam
- Wikipedia. (2020c). *Thomas Schelling*. https://en.wikipedia.org/wiki/Thomas_Schelling
- Wikipedia. (2020d). *Von Neumann universal constructor*. https://en.wikipedia.org/wiki/Von_Neumann_universal_constructor
- Wilensky, U. (1997). *The NetLogo 6.1.1 User Manual*.
- Worldailmi, E., & Ismianti, I. (2020). Simulation of trends in the use of e-payment using agent-based models. *International Journal of Industrial Optimization*, 1(1), 29–42.

GLOSARIUM

<i>Artificial life</i>	<i>A-life</i> adalah bidang studi di mana para peneliti meneliti sistem yang berkaitan dengan kehidupan alam, proses, dan evolusi, melalui penggunaan simulasi dengan model komputer, robotika, dan biokimia.
<i>Biowarfare</i>	Juga dikenal sebagai <i>biological warfare</i> atau <i>germ warfare</i> adalah penggunaan racun biologis atau agen infeksi seperti bakteri, virus, serangga, dan jamur dengan tujuan untuk membunuh atau melumpuhkan manusia, hewan, atau tumbuhan sebagai tindakan perang.
<i>Cellular automata</i>	Sebuah model komputasi diskrit yang dipelajari dalam teori automata.
<i>Journal of Artificial Societies and Social Simulation (JASSS)</i>	Jurnal akademik yang melalui proses <i>peer-review</i> secara triwulanan dan dibuat oleh Nigel Gilbert (<i>University of Surrey</i>).
Logo	Bahasa pemrograman pendidikan, dirancang pada tahun 1967 oleh Wally Feurzeig, Seymour Papert, dan Cynthia Solomon.
NetLogo	Platform simulasi dan pemodelan berbasis agen dan memungkinkan pembuatan model dinamika sistem dan simulasi partisipatif.
<i>Sugarscape</i>	Model untuk simulasi sosial berbasis agen dengan kecerdasan buatan mengikuti beberapa atau semua aturan yang disajikan oleh Joshua M. Epstein & Robert Axtell.

TENTANG PENULIS



Eko Nursubiyantoro menyelesaikan pendidikan S1 Teknik Industri Universitas Pembangunan Nasional (UPN) "Veteran" Yogyakarta pada Tahun 2002, dan menyelesaikan S2 Teknik Industri pada kampus yang sama Tahun 2011. Mulai menjadi Dosen Jurusan Teknik Industri UPN "Veteran" Yogyakarta tahun 2005. Selain mengajar juga aktif melakukan tridharma lainnya yaitu Penelitian dan Pengabdian Masyarakat. Aktif dalam berbagai kegiatan sosial dilingkungan kampus maupun masyarakat sekitar tempat tinggalnya.



Astrid Wahyu Adventri Wibowo menyelesaikan pendidikan S1 Teknik Industri Universitas Sebelas Maret Surakarta (UNS) pada Tahun 2015, dan menyelesaikan S2 Teknik Industri Universitas Gadjah Mada (UGM) Yogyakarta pada Tahun 2017. Ia bergabung menjadi Dosen Jurusan Teknik Industri UPN "Veteran" Yogyakarta pada Tahun 2019. Seminar nasional dan internasional pernah diikutinya selama menempuh pendidikan formal. Dan sudah beberapa karya yang dihasilkan selama beliau menjadi dosen seperti jurnal ilmiah, HKI, dan buku yang sudah dipublikasikan baik di tingkat nasional maupun internasional.



Ismianti menyelesaikan pendidikan S1 Teknik Industri Universitas Gadjah Mada (UGM) Yogyakarta pada Tahun 2012, dan menyelesaikan S2 Teknik Industri UGM pada Tahun 2018. Ia bergabung menjadi Dosen Jurusan Teknik Industri UPN "Veteran" Yogyakarta pada Tahun 2019. Beberapa kegiatan yang pernah diikutinya adalah seminar nasional dan internasional, menulis jurnal dan buku untuk menunjang proses mengajarnya.

Agent-Based Modelling and Simulation dengan Netlogo 6.0

Buku ini berisi informasi tentang teknologi dan informasi terkait Agent-Based Modelling And Simulation (ABMS) dengan menggunakan NetLogo 6.0. ABMS adalah salah satu teknik komputasi, yang tumbuh seiring dengan pematangan teknologi komputer. Munculnya komputasi yang kuat telah membawa perubahan dramatis di banyak bidang kehidupan, termasuk perubahan signifikan dalam praktik dan konten sains. Ketika akses ke komputasi yang kuat meningkat (dan biayanya menurun), para ilmuwan mampu melakukan penghitungan dan simulasi yang sebelumnya tidak mungkin dilakukan. Bahasa pemrograman yang digunakan dalam buku ini adalah NetLogo, dikembangkan oleh Uri Wilensky. Bahasa pemrograman ini dirancang supaya pemula dapat dengan cepat menggunakannya untuk melakukan hal-hal yang bermakna dan berguna namun juga dapat digunakan oleh ilmuwan dan peneliti untuk merancang model ilmiah mutakhir.

ISBN 978-623-6797-16-7

