

SKRIPSI

**PEMBANGUNAN PERANGKAT LUNAK DAN
PENYELESAIAN PERMAINAN COLORED QUEENS**



Arlo Dante Hananvyasa

NPM: 6182201010

**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS
UNIVERSITAS KATOLIK PARAHYANGAN
2026**

UNDERGRADUATE THESIS

**DEVELOPMENT OF A SOFTWARE APPLICATION AND
SOLVER FOR THE COLORED QUEENS GAME**



Arlo Dante Hananvyasa

NPM: 6182201010

**DEPARTMENT OF INFORMATICS
FACULTY OF SCIENCES
PARAHYANGAN CATHOLIC UNIVERSITY
2026**

LEMBAR PENGESAHAN

PEMBANGUNAN PERANGKAT LUNAK DAN PENYELESAIAN PERMAINAN COLORED QUEENS

Arlo Dante Hananvyasa

NPM: 6182201010

Bandung, «tanggal» «bulan» 2026

Menyetujui,

Pembimbing Utama

Pembimbing Pendamping

Husnul Hakim, M.T.

«pembimbing pendamping/2»

Ketua Tim Penguji

Anggota Tim Penguji

«penguji 1»

«penguji 2»

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng

PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi dengan judul:

PEMBANGUNAN PERANGKAT LUNAK DAN PENYELESAIAN PERMAINAN COLORED QUEENS

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,
Tanggal «tanggal» «bulan» 2026

Meterai Rp. 10000

Arlo Dante Hananvyasa
NPM: 6182201010

ABSTRAK

«Tuliskan abstrak anda di sini, dalam bahasa Indonesia»

Kata-kata kunci: «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Indonesia»

ABSTRACT

«Tuliskan abstrak anda di sini, dalam bahasa Inggris»

Keywords: «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Inggris»

«kepada siapa anda mempersembahkan skripsi ini...?»

KATA PENGANTAR

«Tuliskan kata pengantar dari anda di sini ...»

Bandung, «bulan» 2026

Penulis

DAFTAR ISI

KATA PENGANTAR	xv
DAFTAR ISI	xvii
DAFTAR GAMBAR	xix
1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Tujuan	6
1.4 Batasan Masalah	6
1.5 Metodologi	6
1.6 Sistematika Pembahasan	7
2 LANDASAN TEORI	9
2.1 Skripsi	9
2.2 L ^A T _E X	9
2.3 Template Skripsi FTIS UNPAR	9
2.3.1 Tabel	10
2.3.2 Kutipan	10
2.3.3 Gambar	11
2.3.4 Kode Program	12
2.3.5 Notasi	13
DAFTAR REFERENSI	15
A KODE PROGRAM	17
B HASIL EKSPERIMEN	19

DAFTAR GAMBAR

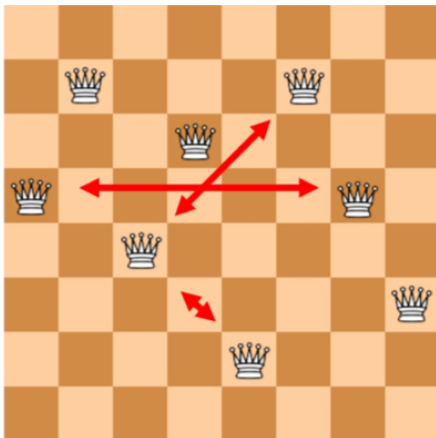
2.1	Gambar <i>Serpentes</i> dalam format png	12
2.2	Ular kecil	12
2.3	<i>Serpentes</i> betina	13
B.1	Hasil 1	19
B.2	Hasil 2	19
B.3	Hasil 3	19
B.4	Hasil 4	19

BAB 1

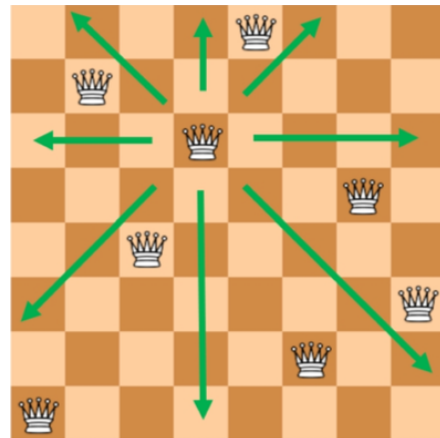
PENDAHULUAN

1.1 Latar Belakang

Masalah n-queens merupakan salah satu permasalahan klasik dalam ilmu komputer yang telah dipelajari secara ekstensif sejak abad ke-19. Dalam bentuk standarnya, masalah n-queens memerlukan penempatan n buah bidak menteri pada papan catur berukuran $n \times n$ sedemikian rupa sehingga tidak ada menteri yang dapat menyerang satu sama lain secara horizontal, vertikal, maupun diagonal. Sebagai contoh, pada papan berukuran 8×8 , terdapat 92 solusi valid yang memenuhi seluruh kendala tersebut. Masalah ini tidak hanya menarik dari segi teoretis, tetapi juga memiliki aplikasi praktis dalam berbagai bidang seperti penjadwalan, alokasi sumber daya, dan desain sirkuit terpadu, sehingga menjadikannya salah satu tolok ukur penting dalam penelitian algoritma pencarian dan pemodelan berbasis kendala.

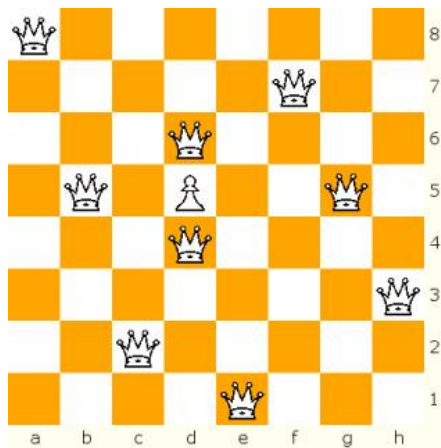


Gambar 1.1(a) Contoh solusi salah masalah N-Queens



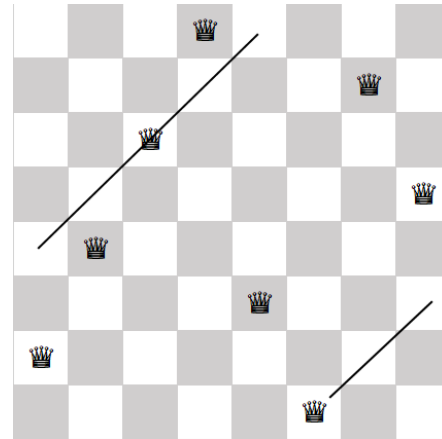
Gambar 1.1(b) Contoh valid permasalahan N-Queens

Seiring perkembangan penelitian, muncul berbagai variasi dari masalah n-queens tradisional yang menawarkan kompleksitas dan tantangan komputasional yang lebih tinggi. Salah satu variasi yang telah diteliti adalah *N+k Queens Problem*, di mana k buah bidak pion ditempatkan sebagai penghalang sehingga memungkinkan penempatan $N + k$ bidak menteri pada papan berukuran $N \times N$. Variasi lainnya adalah *Toroidal N-Queens*, di mana papan catur dibentuk menjadi torus dengan menghubungkan sisi-sisi yang berlawanan, sehingga menteri dapat “melingsar” dari satu sisi ke sisi lainnya. Peningkatan kompleksitas pada variasi-variasi ini terlihat jelas; misalnya pada *N+k Queens Problem* dengan $N = 8$ dan $k = 2$, jumlah kemungkinan konfigurasi yang harus dieksplorasi meningkat secara signifikan dibandingkan masalah standar karena adanya kendala tambahan berupa posisi pion yang tidak dapat dilanggar. Variasi-variasi ini menunjukkan bahwa menambahkan satu atau dua aturan baru saja dapat memperbesar ruang pencarian secara drastis dan mengubah struktur solusi problem secara fundamental.



Gambar 1.2(a) Contoh solusi salah masalah $8+1$ Queens

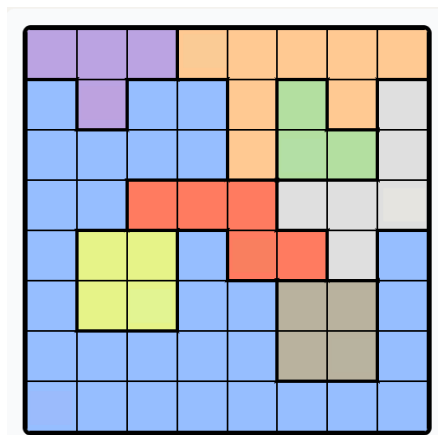
Sumber: <http://www.npluskqueens.info/background.html>



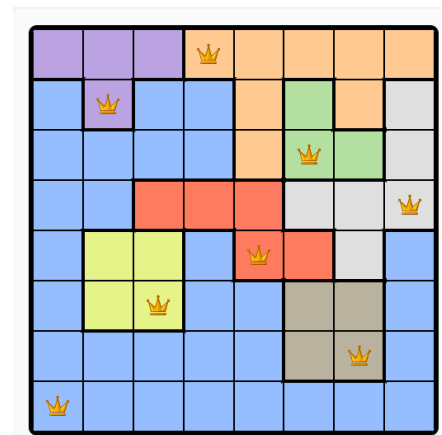
Gambar 1.2(b) Contoh penyerangan yang dapat terjadi pada masalah *Toroidal Queens*

Sumber: <https://www.johndcook.com/blog/2021/08/18/queens-on-a-donut/>

Penelitian ini berfokus pada varian *Colored Queens*, sebuah permasalahan yang hingga saat ini belum memiliki literatur akademis yang memadai dan belum diteliti secara formal dalam publikasi ilmiah. Berbeda dengan masalah n -queens tradisional, permainan *Colored Queens* memiliki aturan yang lebih kompleks: papan permainan dibagi menjadi beberapa sektor berwarna, setiap sektor harus berisi tepat satu bidak menteri, dan tidak ada menteri yang boleh bersebelahan secara langsung, baik horizontal, vertikal, maupun diagonal. Perbedaan fundamental lainnya adalah bahwa bidak menteri pada *Colored Queens* hanya dapat menyerang secara horizontal dan vertikal, sehingga lebih dari satu bidak dapat ditempatkan pada satu garis diagonal yang sama. Kompleksitas komputasional varian ini lebih tinggi dibandingkan permasalahan n -queens tradisional karena terdapat tiga lapis kendala yang harus dipenuhi secara simultan: kendala partisi warna (setiap warna tepat satu menteri), kendala *adjacency* (tidak boleh bersebelahan), dan kendala serangan (horizontal dan vertikal). Sebagai ilustrasi, pada papan berukuran 6×6 dengan 6 sektor warna, ruang pencarian solusi menjadi jauh lebih sempit dan bergantung pada struktur pembagian sektor, sehingga kompleksitasnya umumnya lebih tinggi dibandingkan 6-queens standar yang memiliki ruang solusi yang lebih teratur. Ketidakteraturan struktur sektor warna juga membuat heuristik dan simetri yang biasa dimanfaatkan pada n -queens standar tidak lagi berlaku, sehingga *Colored Queens* merupakan permasalahan yang jauh lebih sulit.



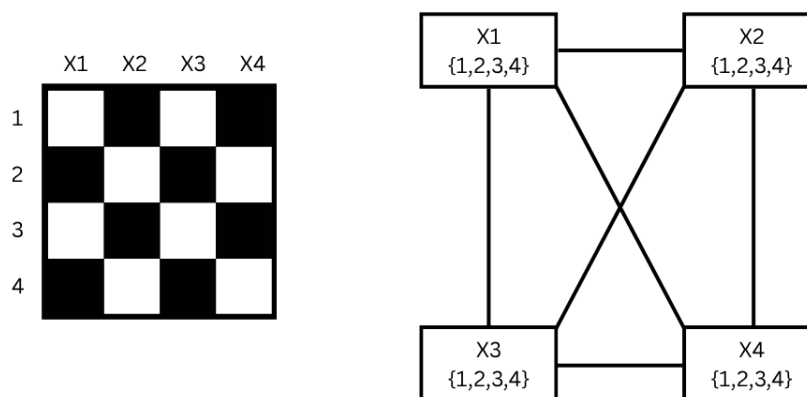
Gambar 1.3(a) Contoh kondisi awal permainan *Colored Queens*



Gambar 1.3(b) Solusi valid permainan *Colored Queens*

Masalah n -queens dan seluruh variasinya, termasuk *Colored Queens*, tergolong ke dalam kategori *Constraint Satisfaction Problem* (CSP). CSP adalah jenis permasalahan yang melibatkan pencarian solusi dengan memberikan nilai pada sejumlah variabel sedemikian rupa sehingga memenuhi seperangkat batasan atau kendala tertentu. Secara formal, sebuah CSP didefinisikan oleh tiga komponen: himpunan variabel $X = X_1, X_2, \dots, X_n$, himpunan domain $D = D_1, D_2, \dots, D_n$ yang berisi nilai-nilai yang mungkin untuk setiap variabel, dan himpunan kendala C yang membatasi kombinasi nilai yang dapat diberikan pada variabel-variabel tersebut. Dalam konteks *Colored Queens*, variabel-variabelnya adalah posisi menteri untuk setiap sektor warna, domain untuk setiap variabel adalah sel-sel yang tersedia pada sektor tersebut, dan kendala-kendalanya mencakup aturan tidak bersebelahan serta tidak saling menyerang. CSP memiliki aplikasi luas dalam penjadwalan, perencanaan, konfigurasi produk, desain jaringan, serta berbagai sistem pengambilan keputusan berbasis kendala.

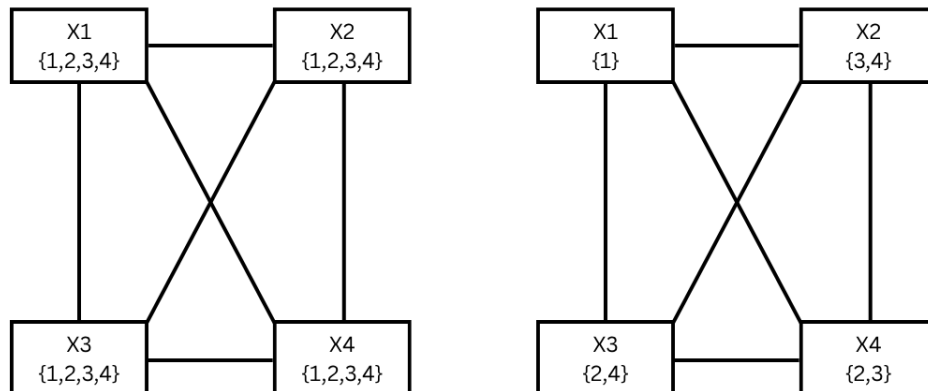
Penyelesaian permasalahan CSP, terutama yang memiliki ruang pencarian besar dan kompleks seperti *Colored Queens*, dapat didekati melalui perspektif optimasi. Optimasi adalah proses pencarian solusi terbaik dari sekumpulan alternatif yang memungkinkan, biasanya dengan memaksimalkan atau meminimalkan suatu fungsi objektif tertentu. Dalam konteks CSP, fungsi objektif dapat berupa jumlah kendala yang dilanggar, sehingga solusi optimal adalah konfigurasi yang tidak melanggar kendala sama sekali (nilai objektif nol). Algoritma optimasi umumnya bekerja dengan menggabungkan strategi eksplorasi dan eksploitasi terhadap ruang solusi: eksplorasi memungkinkan algoritma menjelajahi area solusi baru, sedangkan eksploitasi berfokus pada penyempurnaan solusi yang sudah dianggap menjanjikan. Keseimbangan antara keduanya sangat penting, karena terlalu banyak eksploitasi berpotensi membuat algoritma terjebak pada solusi lokal, yaitu kondisi ketika algoritma menemukan konfigurasi yang tampak optimal di sekitar posisinya saat ini tetapi sebenarnya bukan solusi terbaik secara global. Perspektif optimasi menjadi relevan karena banyak permasalahan CSP tergolong *NP-hard*, yaitu kelas masalah yang tidak diketahui memiliki algoritma yang dapat diselesaikan secara efisien dalam waktu polinomial. Waktu polinomial berarti waktu komputasi yang meningkat secara wajar terhadap ukuran input (misalnya $O(n^2)$ atau $O(n^3)$), sedangkan masalah NP-hard umumnya membutuhkan waktu eksponensial yang meningkat jauh lebih cepat sehingga tidak praktis untuk diselesaikan secara eksak pada ukuran besar, sehingga metode optimasi menjadi alternatif yang lebih realistis.



Gambar 1.3(a) Contoh permasalahan *4-Queens* yang dimodelkan sebagai CSP, dimana variabel X_1 s.d. X_4 adalah baris bidak menteri pada setiap kolomnya dengan nilai kemungkinan 1,2,3,4 dan kendala yang menghubungkan setiap variabel berupa aturan penyerangan bidak menteri pada permainan catur

Untuk menyelesaikan permasalahan CSP seperti *Colored Queens*, salah satu pendekatan klasik yang terbukti efektif adalah algoritma *backtracking*. *Backtracking* bekerja dengan membangun solusi secara bertahap melalui pencarian mendalam, di mana setiap variabel diberi nilai satu per satu sambil memeriksa konsistensi dengan kendala yang ada. Ketika algoritma menemui situasi di mana tidak ada nilai yang valid untuk variabel berikutnya (*dead end*), algoritma akan mundur (*backtrack*) ke langkah sebelumnya dan mencoba alternatif lain. Kekuatan *backtracking* terletak pada sifatnya yang sistematis dan kemampuannya untuk menjamin bahwa solusi akan ditemukan jika memang ada. Namun, pada masalah dengan ruang pencarian yang berkembang secara eksponensial seperti *Colored Queens*, *backtracking* murni dapat menjadi sangat tidak efisien karena harus mengeksplorasi sejumlah besar kemungkinan konfigurasi sebelum menemukan solusi yang valid. Oleh karena itu, diperlukan modifikasi dan teknik tambahan untuk memangkas ruang pencarian agar algoritma tetap praktis digunakan pada instansi masalah *Colored Queens* yang berukuran besar seperti papan 20×20 dan 30×30 .

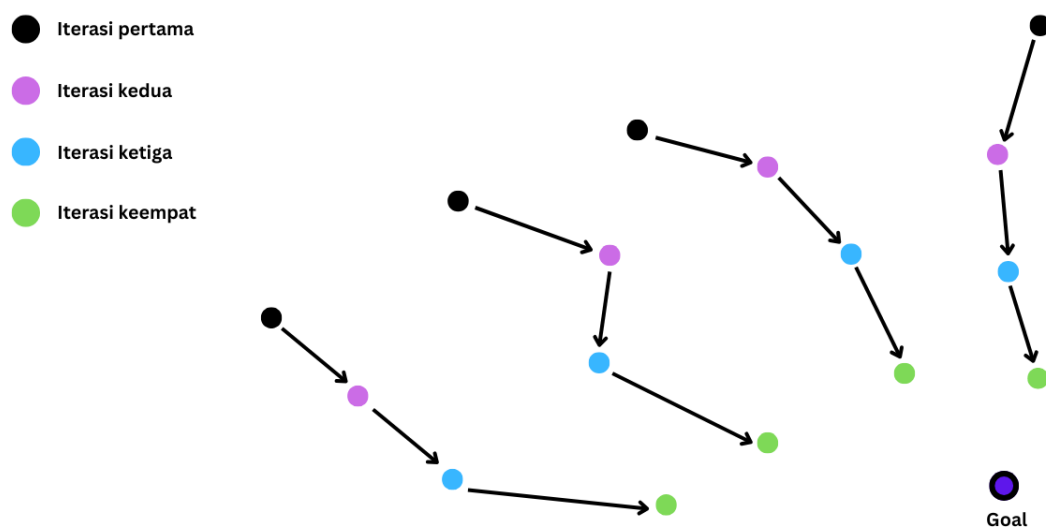
Untuk meningkatkan efisiensi algoritma *backtracking*, penelitian ini mengintegrasikan teknik *Maintaining Arc Consistency* (MAC) yang mengimplementasikan algoritma *Arc Consistency 3* (AC-3). AC-3 adalah teknik propagasi kendala yang berfungsi untuk menyempitkan ruang pencarian dengan mengeliminasi nilai-nilai dalam domain yang tidak mungkin menghasilkan solusi valid berdasarkan kendala antar variabel. Algoritma ini bekerja dengan memeriksa konsistensi antara pasangan variabel dan secara iteratif menghapus nilai-nilai yang tidak memiliki pasangan yang konsisten pada variabel lain. Dengan menjalankan AC-3 pada tahap preprocessing sebelum pencarian dimulai serta setiap kali *backtracking* membuat assignment baru, banyak konfigurasi yang tidak valid dapat dieliminasi lebih awal sebelum algoritma membuang waktu mengeksplorasinya. Pada problem seperti *Colored Queens* yang memiliki ketergantungan antar variabel sangat kuat akibat kendala *adjacency* dan partisi warna, propagasi kendala memberikan dampak reduksi ruang solusi yang signifikan sehingga dapat mengurangi jumlah langkah *backtracking* yang diperlukan secara drastis dan memperpendek waktu yang dibutuhkan untuk mencari solusi yang valid.



Gambar 1.3(a) Contoh ruang solusi *4-Queens* yang dimodelkan sebagai CSP, sebelum(kiri) dan sesudah(kanan) menjalankan AC-3 pada saat menaruh bidak menteri di posisi 1 di kolom X1

Sebagai alternatif dari pendekatan deterministik, penelitian ini juga mengeksplorasi penggunaan *Particle Swarm Optimization* (PSO), sebuah algoritma metaheuristik yang terinspirasi dari perilaku kolektif kawanan burung atau ikan dalam mencari makanan. Metaheuristik adalah strategi pencarian tingkat tinggi yang memandu proses eksplorasi ruang solusi tanpa menjamin menemukan solusi optimal, namun seringkali dapat menemukan solusi yang cukup baik dalam waktu yang lebih

singkat dibandingkan metode eksak seperti backtracking. PSO bekerja dengan mensimulasikan sekumpulan partikel yang bergerak dalam ruang solusi, di mana setiap partikel menyesuaikan posisinya berdasarkan pengalaman terbaiknya sendiri dan pengalaman terbaik kawanannya. Kelebihan PSO terletak pada kemampuannya untuk melakukan eksplorasi ruang solusi secara paralel dan menghindari jebakan solusi lokal dalam beberapa kasus. Namun, karena PSO dirancang untuk ruang solusi kontinu, penerapannya pada *Colored Queens* yang bersifat diskret memerlukan adaptasi khusus. Adaptasi tersebut mencakup representasi setiap partikel sebagai susunan posisi menteri untuk setiap sektor warna pada papan. Konsep velocity dalam PSO juga dimodifikasi menjadi bilangan desimal yang merepresentasikan probabilitas seberapa besar kemungkinan posisi menteri pada suatu sektor akan beralih mengikuti konfigurasi Neighbourhood Best, serta prosedur perbaikan (*repair mechanisms*) untuk menangani solusi yang melanggar kendala hard constraint seperti aturan satu menteri per warna.. Adaptasi ini penting agar PSO tetap relevan dan dapat memberikan performa yang kompetitif pada problem bersifat kombinatorial.



Gambar 1.3(a) Contoh pergerakan partikel-partikle ke arah solusi optimal dengan setiap iterasi

Sumber: Modifikasi dari

https://www.researchgate.net/figure/sualization-of-the-PSO-Algorithm_fig3_334363118

Dari perspektif pengembangan perangkat lunak, penelitian ini tidak hanya berfokus pada aspek algoritmik, tetapi juga pada penyediaan antarmuka pengguna yang intuitif dan edukatif. Akan dikembangkan sebuah aplikasi berbasis web yang memungkinkan pengguna untuk berinteraksi langsung dengan permainan *Colored Queens*, memilih tingkat kesulitan berdasarkan ukuran papan dan kompleksitas pembagian sektor, serta mencoba menyelesaikan *puzzle* secara manual. Aplikasi ini akan dilengkapi sistem peringatan visual yang memberikan umpan balik real-time ketika pengguna menempatkan menteri pada posisi yang melanggar kendala, sehingga membantu pemahaman terhadap aturan permainan. Selain mode permainan manual, aplikasi juga akan menyediakan mode demonstrasi yang menampilkan proses pencarian solusi oleh algoritma backtracking dan PSO secara visual, lengkap dengan metrik performa seperti waktu eksekusi, jumlah iterasi, dan perbandingan efektivitas kedua metode. Pendekatan ini diharapkan dapat menjembatani kesenjangan antara teori algoritma dan aplikasi praktis, serta memberikan alat pembelajaran interaktif bagi pengguna yang tertarik mempelajari teknik penyelesaian CSP.

1.2 Rumusan Masalah

- Bagaimana cara membangun perangkat lunak permainan *Colored Queens*?

- Bagaimana membangun solusi permainan *Colored Queens* menggunakan teknik *Backtracking* dan *Particle Swarm Optimization* (PSO)?
- Bagaimana membangun solver untuk permainan colored queen yang mengimplementasikan *Backtracking* dan PSO yang dapat diintegrasikan dengan perangkat lunak yang dibangun?
- Bagaimana kinerja dari solver yang dibangun dalam mencari solusi permainan *Colored Queens*?

1.3 Tujuan

- Membangun perangkat lunak permainan *Colored Queens*.
- Mempelajari cara membangun solusi permainan Colored Queen menggunakan teknik *Backtracking* dan *Particle Swarm Optimization*.
- Membangun solver untuk permainan colored queen yang mengimplementasikan *Backtracking* dan PSO yang dapat diintegrasikan dengan perangkat lunak yang dibangun.
- Melakukan pengujian untuk mengukur kinerja dari solver yang dibangun dalam mencari solusi permainan *Colored Queens*

1.4 Batasan Masalah

1. Penelitian ini hanya membahas varian *Colored Queens* dengan pembagian sektor warna yang diberikan secara statis.
2. Jumlah sektor warna selalu sama dengan jumlah menteri yang harus ditempatkan pada papan.
3. Pembagian sektor warna tidak berubah selama proses pencarian solusi.
4. Aturan serangan menteri terbatas pada arah horizontal dan vertikal.
5. Kendala *adjacency* diterapkan pada delapan arah di sekitar setiap sel.
6. Algoritma yang dianalisis terbatas pada:
 - (a) *Backtracking* dengan beberapa modifikasi,
 - (b) Teknik propagasi kendala *Arc Consistency* (AC-3) yang akan diterapkan pada backtracking saja.
 - (c) *Particle Swarm Optimization* (PSO) yang telah diadaptasi untuk ruang solusi diskret.
7. Evaluasi performa algoritma hanya mencakup waktu eksekusi, jumlah iterasi, serta tingkat keberhasilan dalam menemukan solusi.
8. Aplikasi yang dibangun bersifat *web-based* dan hanya menyediakan fitur visualisasi papan, penempatan menteri secara manual, pemberian bantuan dan peringatan untuk pemain, dan demonstrasi algoritma.
9. Aplikasi tidak mencakup fitur seperti autentikasi pengguna, penyimpanan basis data, atau mode *multiplayer*
10. Pengujian perangkat lunak hanya dilakukan pada ukuran papan 20×20 dan 30×30 .

1.5 Metodologi

Tentunya akan diisi dengan metodologi yang serius sehingga templatennya terkesan lebih serius.

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetur a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetur. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

1.6 Sistematika Pembahasan

Rencananya Bab 2 akan berisi petunjuk penggunaan template dan dasar-dasar L^AT_EX. Mungkin bab 3,4,5 dapt diisi oleh ketiga jurusan, misalnya peraturan dasar skripsi atau pedoman penulisan, tentu jika berkenan. Bab 6 akan diisi dengan kesimpulan, bahwa membuat template ini ternyata sungguh menghabiskan banyak waktu.

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetur odio sem sed wisi.

BAB 2

LANDASAN TEORI

2.1 Skripsi

Rencananya akan diisi dengan penjelasan umum mengenai buku skripsi.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetur eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetur tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

2.2 L^AT_EX

Mengapa menggunakan L^AT_EX untuk buku skripsi dan apa keunggulan/kerugiannya bagi mahasiswa dan pembuat template.

Aliquam lectus. Vivamus leo. Quisque ornare tellus ullamcorper nulla. Mauris porttitor pharetra tortor. Sed fringilla justo sed mauris. Mauris tellus. Sed non leo. Nullam elementum, magna in cursus sodales, augue est scelerisque sapien, venenatis congue nulla arcu et pede. Ut suscipit enim vel sapien. Donec congue. Maecenas urna mi, suscipit in, placerat ut, vestibulum ut, massa. Fusce ultrices nulla et nisl.

Etiam ac leo a risus tristique nonummy. Donec dignissim tincidunt nulla. Vestibulum rhoncus molestie odio. Sed lobortis, justo et pretium lobortis, mauris turpis condimentum augue, nec ultricies nibh arcu pretium enim. Nunc purus neque, placerat id, imperdiet sed, pellentesque nec, nisl. Vestibulum imperdiet neque non sem accumsan laoreet. In hac habitasse platea dictumst. Etiam condimentum facilisis libero. Suspendisse in elit quis nisl aliquam dapibus. Pellentesque auctor sapien. Sed egestas sapien nec lectus. Pellentesque vel dui vel neque bibendum viverra. Aliquam porttitor nisl nec pede. Proin mattis libero vel turpis. Donec rutrum mauris et libero. Proin euismod porta felis. Nam lobortis, metus quis elementum commodo, nunc lectus elementum mauris, eget vulputate ligula tellus eu neque. Vivamus eu dolor.

2.3 Template Skripsi FTIS UNPAR

Akan dipaparkan bagaimana menggunakan template ini, termasuk petunjuk singkat membuat referensi, gambar dan tabel. Juga hal-hal lain yang belum terpikir sampai saat ini.

Nulla in ipsum. Praesent eros nulla, congue vitae, euismod ut, commodo a, wisi. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean nonummy

magna non leo. Sed felis erat, ullamcorper in, dictum non, ultricies ut, lectus. Proin vel arcu a odio lobortis euismod. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Proin ut est. Aliquam odio. Pellentesque massa turpis, cursus eu, euismod nec, tempor congue, nulla. Duis viverra gravida mauris. Cras tincidunt. Curabitur eros ligula, varius ut, pulvinar in, cursus faucibus, augue.

Nulla mattis luctus nulla. Duis commodo velit at leo. Aliquam vulputate magna et leo. Nam vestibulum ullamcorper leo. Vestibulum condimentum rutrum mauris. Donec id mauris. Morbi molestie justo et pede. Vivamus eget turpis sed nisl cursus tempor. Curabitur mollis sapien condimentum nunc. In wisi nisl, malesuada at, dignissim sit amet, lobortis in, odio. Aenean consequat arcu a ante. Pellentesque porta elit sit amet orci. Etiam at turpis nec elit ultricies imperdiet. Nulla facilisi. In hac habitasse platea dictumst. Suspendisse viverra aliquam risus. Nullam pede justo, molestie nonummy, scelerisque eu, facilisis vel, arcu.

2.3.1 Tabel

Berikut adalah contoh pembuatan tabel. Penempatan tabel dan gambar secara umum diatur secara otomatis oleh \LaTeX , perhatikan contoh di file bab2.tex untuk melihat bagaimana cara memaksa tabel ditempatkan sesuai keinginan kita.

Perhatikan bawa berbeda dengan penempatan judul gambar gambar, keterangan tabel harus diletakkan di atas tabel!! Lihat Tabel 2.1 berikut ini:

Tabel 2.1: Tabel contoh

	v_{start}	\mathcal{S}_1	v_{end}
τ_1	1	12	20
τ_2	1		20
τ_3	1	9	20
τ_4	1		20

Tabel 2.2 dan Tabel 2.3 berikut ini adalah tabel dengan sel yang berwarna dan ada dua tabel yang bersebelahan.

Tabel 2.2: Tabel bewarna(1)

	v_{start}	\mathcal{S}_2	\mathcal{S}_1	v_{end}
τ_1	1	5	12	20
τ_2	1	8		20
τ_3	1	2/8/17	9	20
τ_4	1			20

Tabel 2.3: Tabel bewarna(2)

	v_{start}	\mathcal{S}_1	\mathcal{S}_2	v_{end}
τ_1	1	12	5	20
τ_2	1		8	20
τ_3	1	9	2/8/17	20
τ_4	1			20

2.3.2 Kutipan

Berikut contoh kutipan dari berbagai sumber, untuk keterangan lebih lengkap, silahkan membaca file referensi.bib yang disediakan juga di template ini. Contoh kutipan:

- Buku: [?]
- Bab dalam buku: [?]
- Artikel dari Jurnal: [?]
- Artikel dari prosiding seminar/konferensi: [?]
- Skripsi/Thesis/Disertasi: [?] [?] [?]
- Technical/Scientific Report: [?]
- RFC (Request For Comments): [?]

- Technical Documentation/Technical Manual: [?] [?] [?]
- Paten: [?]
- Tidak dipublikasikan: [?] [?]
- Laman web: [?]
- Lain-lain: [?]

2.3.3 Gambar

Pada hampir semua editor, penempatan gambar di dalam dokumen L^AT_EX tidak dapat dilakukan melalui proses *drag and drop*. Perhatikan contoh pada file bab2.tex untuk melihat bagaimana cara menempatkan gambar. Beberapa hal yang harus diperhatikan pada saat menempatkan gambar:

- Setiap gambar **harus** diacu di dalam teks (gunakan *field LABEL*)
- *Field CAPTION* digunakan untuk teks pengantar pada gambar. Terdapat dua bagian yaitu yang ada di antara tanda [dan] dan yang ada di antara tanda { dan }. Yang pertama akan muncul di Daftar Gambar, sedangkan yang kedua akan muncul di teks pengantar gambar. Untuk skripsi ini, samakan isi keduanya.
- Jenis file yang dapat digunakan sebagai gambar cukup banyak, tetapi yang paling populer adalah tipe PNG (lihat Gambar 2.1), tipe JPG (Gambar 2.2) dan tipe PDF (Gambar 2.3)
- Besarnya gambar dapat diatur dengan *field SCALE*.
- Penempatan gambar diatur menggunakan *placement specifier* (di antara tanda [dan] setelah deklarasi gambar. Yang umum digunakan adalah **H** untuk menempatkan gambar **sesuai** penempatannya di file .tex atau **h** yang berarti "kira-kira" di sini.

Jika tidak menggunakan *placement specifier*, L^AT_EX akan menempatkan gambar secara otomatis untuk menghindari bagian kosong pada dokumen anda. Walaupun cara ini sangat mudah, hindarkan terjadinya penempatan dua gambar secara berurutan.

- Gambar 2.1 ditempatkan di bagian atas halaman, walaupun penempatannya dilakukan setelah penulisan 3 paragraf setelah penjelasan ini.
- Gambar 2.2 dengan skala 0.5 ditempatkan di antara dua buah paragraf. Perhatikan penulisannya di dalam file bab2.tex!
- Gambar 2.3 ditempatkan menggunakan *specifier h*.

Curabitur tellus magna, porttitor a, commodo a, commodo in, tortor. Donec interdum. Praesent scelerisque. Maecenas posuere sodales odio. Vivamus metus lacus, varius quis, imperdiet quis, rhoncus a, turpis. Etiam ligula arcu, elementum a, venenatis quis, sollicitudin sed, metus. Donec nunc pede, tincidunt in, venenatis vitae, faucibus vel, nibh. Pellentesque wisi. Nullam malesuada. Morbi ut tellus ut pede tincidunt porta. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam congue neque id dolor.

Donec et nisl at wisi luctus bibendum. Nam interdum tellus ac libero. Sed sem justo, laoreet vitae, fringilla at, adipiscing ut, nibh. Maecenas non sem quis tortor eleifend fermentum. Etiam id tortor ac mauris porta vulputate. Integer porta neque vitae massa. Maecenas tempus libero a libero posuere dictum. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aenean quis mauris sed elit commodo placerat. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Vivamus rhoncus tincidunt libero. Etiam elementum pretium justo. Vivamus est. Morbi a tellus eget pede tristique commodo. Nulla nisl. Vestibulum sed nisl eu sapien cursus rutrum.

Nulla non mauris vitae wisi posuere convallis. Sed eu nulla nec eros scelerisque pharetra. Nullam varius. Etiam dignissim elementum metus. Vestibulum faucibus, metus sit amet mattis rhoncus, sapien dui laoreet odio, nec ultricies nibh augue a enim. Fusce in ligula. Quisque at magna et nulla commodo consequat. Proin accumsan imperdiet sem. Nunc porta. Donec feugiat mi at justo. Phasellus facilisis ipsum quis ante. In ac elit eget ipsum pharetra faucibus. Maecenas viverra nulla in massa.

Nulla ac nisl. Nullam urna nulla, ullamcorper in, interdum sit amet, gravida ut, risus. Aenean ac enim. In luctus. Phasellus eu quam vitae turpis viverra pellentesque. Duis feugiat felis ut enim.



Gambar 2.1: Gambar *Serpentes* dalam format png

Phasellus pharetra, sem id porttitor sodales, magna nunc aliquet nibh, nec blandit nisl mauris at pede. Suspendisse risus risus, lobortis eget, semper at, imperdiet sit amet, quam. Quisque scelerisque dapibus nibh. Nam enim. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc ut metus. Ut metus justo, auctor at, ultrices eu, sagittis ut, purus. Aliquam aliquam.



Gambar 2.2: Ular kecil

Etiam pede massa, dapibus vitae, rhoncus in, placerat posuere, odio. Vestibulum luctus commodo lacus. Morbi lacus dui, tempor sed, euismod eget, condimentum at, tortor. Phasellus aliquet odio ac lacus tempor faucibus. Praesent sed sem. Praesent iaculis. Cras rhoncus tellus sed justo ullamcorper sagittis. Donec quis orci. Sed ut tortor quis tellus euismod tincidunt. Suspendisse congue nisl eu elit. Aliquam tortor diam, tempus id, tristique eget, sodales vel, nulla. Praesent tellus mi, condimentum sed, viverra at, consectetur quis, lectus. In auctor vehicula orci. Sed pede sapien, euismod in, suscipit in, pharetra placerat, metus. Vivamus commodo dui non odio. Donec et felis.

Etiam suscipit aliquam arcu. Aliquam sit amet est ac purus bibendum congue. Sed in eros. Morbi non orci. Pellentesque mattis lacinia elit. Fusce molestie velit in ligula. Nullam et orci vitae nibh vulputate auctor. Aliquam eget purus. Nulla auctor wisi sed ipsum. Morbi porttitor tellus ac enim. Fusce ornare. Proin ipsum enim, tincidunt in, ornare venenatis, molestie a, augue. Donec vel pede in lacus sagittis porta. Sed hendrerit ipsum quis nisl. Suspendisse quis massa ac nibh pretium cursus. Sed sodales. Nam eu neque quis pede dignissim ornare. Maecenas eu purus ac urna tincidunt congue.

2.3.4 Kode Program

Kode program dalam bahasa tertentu seringkali harus ditulis di dalam bab, bukan hanya dilampirkan di bagian Lampiran. Kode 2.1 menampilkan penggunaan karakter-karakter yang umum digunakan dalam sebuah program yang ditulis dengan bahasa C.

Gambar 2.3: *Serpentes* jantan

Kode 2.1: Kode untuk menampilkan karakter-karakter aneh

```

1 // This does not make algorithmic sense,
2 // but it shows off significant programming characters.
3
4 #include<stdio.h>
5
6 void myFunction( int input, float* output ) {
7     switch ( array[i] ) {
8         case 1: // This is silly code
9             if ( a >= 0 || b <= 3 && c != x )
10                 *output += 0.005 + 20050;
11             char = 'g';
12             b = 2^n + ~right_size - leftSize * MAX_SIZE;
13             c = (--aaa + &daa) / (bbb++ - ccc % 2 );
14             strcpy(a,"hello_$@?");
15         }
16         count = ~mask | 0x00FF00AA;
17     }
18
19 // Fonts for Displaying Program Code in LATEX
20 // Adrian P. Robson, nepsweb.co.uk
21 // 8 October 2012
22 // http://nepsweb.co.uk/docs/progfonts.pdf

```

2.3.5 Notasi

Simbol-simbol (matematika) yang sering digunakan sepanjang penulisan skripsi, dapat dimasukkan ke dalam “Daftar Notasi”. Daftar ini ada di halaman depan sebelum Bab 1. Cara memasukkan sebuah simbol ke dalam Daftar Notasi adalah menggunakan perintah `\nomenclature`. Contoh:

```
\nomenclature[]{$A$}{luas kandang ular}
```

Argumen opsional digunakan untuk mengurutkan notasi. Silahkan lihat sendiri dokumentasi package `nomencl`

DAFTAR REFERENSI

LAMPIRAN A

KODE PROGRAM

Kode A.1: MyCode.c

```

1 // This does not make algorithmic sense,
2 // but it shows off significant programming characters.
3
4 #include<stdio.h>
5
6 void myFunction( int input, float* output ) {
7     switch ( array[i] ) {
8         case 1: // This is silly code
9             if ( a >= 0 || b <= 3 && c != x )
10                 *output += 0.005 + 20050;
11             char = 'g';
12             b = 2^n + ~right_size - leftSize * MAX_SIZE;
13             c = (--aaa + &daa) / (bbb++ - ccc % 2 );
14             strcpy(a,"hello_$@?");
15         }
16         count = ~mask | 0x00FF00AA;
17     }
18 }
19
20 // Fonts for Displaying Program Code in LATEX
21 // Adrian P. Robson, nepswb.co.uk
22 // 8 October 2012
23 // http://nepswb.co.uk/docs/progfonts.pdf

```

Kode A.2: MyCode.java

```

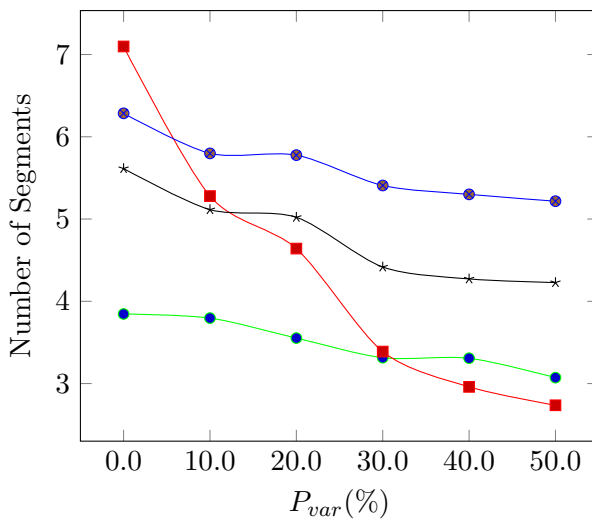
1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.HashSet;
4
5 //class for set of vertices close to furthest edge
6 public class MyFurSet {
7     protected int id; //id of the set
8     protected MyEdge FurthestEdge; //the furthest edge
9     protected HashSet<MyVertex> set; //set of vertices close to furthest edge
10    protected ArrayList<ArrayList<Integer>> ordered; //list of all vertices in the set for each trajectory
11    protected ArrayList<Integer> closeID; //store the ID of all vertices
12    protected ArrayList<Double> closeDist; //store the distance of all vertices
13    protected int totaltrj; //total trajectories in the set
14
15    /*
16     * Constructor
17     * @param id : id of the set
18     * @param totaltrj : total number of trajectories in the set
19     * @param FurthestEdge : the furthest edge
20     */
21    public MyFurSet(int id,int totaltrj,MyEdge FurthestEdge) {
22        this.id = id;
23        this.totaltrj = totaltrj;
24        this.FurthestEdge = FurthestEdge;
25        set = new HashSet<MyVertex>();
26        ordered = new ArrayList<ArrayList<Integer>>();
27        for (int i=0;i<totaltrj;i++) ordered.add(new ArrayList<Integer>());
28        closeID = new ArrayList<Integer>(totaltrj);
29        closeDist = new ArrayList<Double>(totaltrj);
30        for (int i = 0;i <totaltrj;i++) {
31            closeID.add(-1);
32            closeDist.add(Double.MAX_VALUE);
33        }
34    }
35
36 }

```

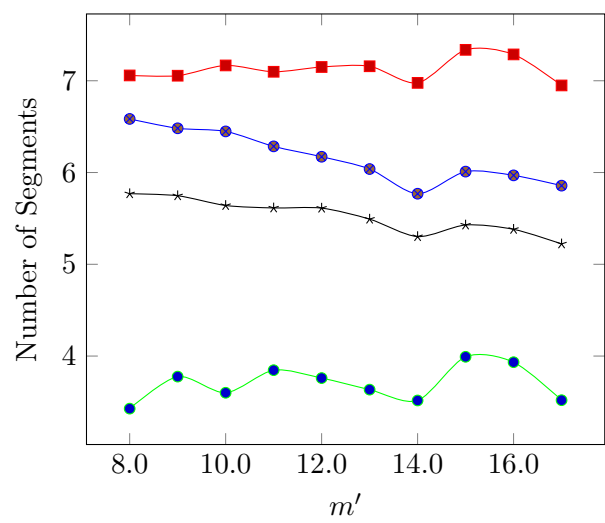

LAMPIRAN B

HASIL EKSPERIMEN

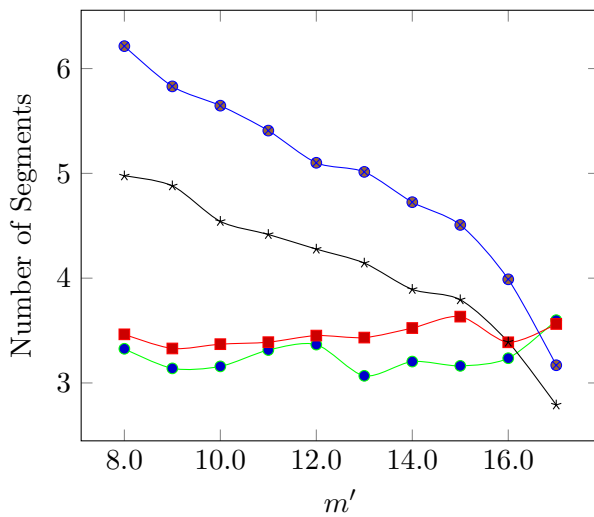
Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



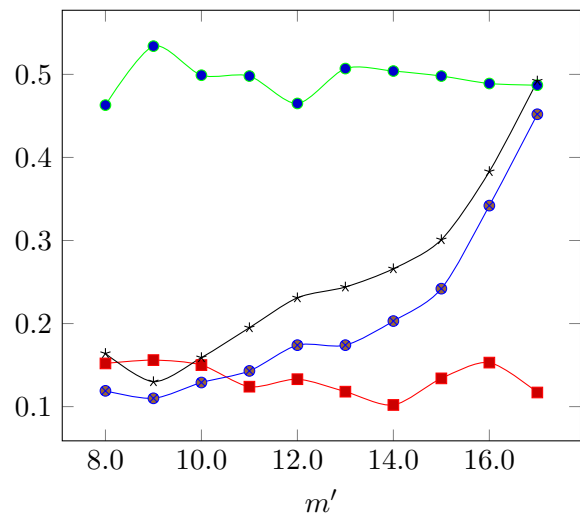
Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4