



INSTITUT TEKNOLOGI DEL

**Implementasi Algoritme *Backtracking* CSP dan *Steepest Ascent Hill Climbing* pada Penjadwalan Kapal
Penyeberangan
(Studi Kasus: Ajibata)**

TUGAS AKHIR

**Diajukan sebagai salah satu syarat untuk memperoleh gelar
Diploma III
Program Studi Teknologi Informasi**

Oleh:

NIM	NAMA
11317014	Yohana Purba
11317017	Grace Anastasya M Sihombing
11317046	Amsal Marulitua Sianipar

**FAKULTAS INFORMASI DAN TEKNIK ELEKTRO
PROGRAM STUDI DIII TEKNOLOGI INFORMASI**

LAGUBOTI

2020

HALAMAN PERNYATAAN ORISINALITAS

Tugas Akhir ini adalah hasil karya saya sendiri, dan semua sumber baik yang dikutip maupun dirujuk telah saya nyatakan dengan benar.

NAMA : Yohana Purba

NIM : 11317014

TANDA TANGAN : 

TANGGAL : 3 Agustus 2020

NAMA : Grace Anastaya M Sihombing

NIM : 11317017

TANDA TANGAN : 

TANGGAL : 3 Agustus 2020

NAMA : Amsal Marulitua Sianipar

NIM : 11317046

TANDA TANGAN : 

TANGGAL : 3 Agustus 2020

HALAMAN PENGESAHAN

Tugas Akhir ini diajukan oleh :

1. Nama : Yohana Purba
Nim : 11317014
 2. Nama : Grace Anastasya Sihombing
Nim : 11317017
 3. Nama : Amsal Marulitua Sianipar
Nim : 11317046
- Program studi : Diploma III Teknologi Informasi
- Judul Tugas Akhir : Implementasi Algoritma
Backtracking CSP dan *Steepest Ascent Hill Climbing* pada
Penjadwalan Kapal Penyeberangan
(Studi Kasus: Ajibata)

Telah berhasil dipertahankan dihadapannya dewan penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Diploma Ahli Madya Komputer pada program Studi Diploma III Teknologi Informasi, Fakultas Informatika dan Teknik Elektro Institut Teknologi Del.

DEWAN PENGUJI

- Pembimbing : Yohanssen Pratama, S.Si, M.T. ()
- Pembimbing : Monalisa Pasaribu, SS., M.Ed. ()
- Penguji : Lit Malem Ginting, S.Si, MT. ()
- Penguji : Hernawati Samosir, S.ST., ()
- Ditetapkan : Laguboti
- Tanggal : 3 Agustus 2020

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa, karena atas kasih karuniaNya, penulis dapat menyelesaikan Tugas Akhir 2 (TA2) dengan judul ” **Implementasi Algoritme *Backtracking* CSP dan *Steepest Ascent Hill Climbing* pada Penjadwalan Kapal Penyeberangan**” ini. Penulis Tugas Akhir ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Diploma Tiga Teknologi Informasi Fakultas Teknik Informatika dan Teknik Elektro Institut Teknologi Del. Penulis menyadari bahwa, tanpa bantuan, dukungan, dan bimbingan dari berbagai pihak, dimulai dari masa perkuliahan sampai pada penyusunan Tugas Akhir ini, sangatlah sulit bagi penulis untuk menyelesaikan Tugas Akhir ini. Oleh karena itu, penulis menyampaikan rasa terimakasih kepada:

1. Terimakasih **Allah Bapa, Tuhan Yesus Kristus, dan Roh Kudus**, semua ini bisa terlewati hanya karena anugerah dan kekuatan dariMu saja. TanpaMu, penulis hanya butiran debu yang lemah dan tak berarti. Terimakasih karena selalu ada, terimakasih karena hadirMu hanya sejauh doa. Tugas Akhir ini penulis persembahkan untuk memuliakan namaMu, ya Tuhan.
2. Terimakasih atas semua dukungan dari **Keluarga**, karena keluarga adalah motivasi terbesar penulis.
3. Terimakasih Dosen-Dosen Institut Teknologi Del, terkhusus **Yuniarta Basani, S.Si., M.Si.**, selaku dosen pembimbing pertama di Tugas Akhir 1, bapak **Yohanssen Pratama, S.Si, M.T**, selaku dosen pembimbing pertama di Tugas Akhir 2 dan ibu **Monalisa Pasaribu, SS., M.Ed (TESOL)**, selaku dosen pembimbing kedua yang telah menyediakan waktu, tenaga dan pikiran untuk membantu penulis dalam mengarahkan dalam penyusunan Tugas Akhir ini, tanpa bapak dan ibu ulat seperti penulis ini takkan berubah menjadi kupu-kupu yang indah.

4. Terimakasih bapak **Lit Malem, S.SI, MT**, dan ibu **Hernawati Samosir, S.ST., M.kom**, selaku dosen penguji mulai dari seminar proposal hingga sidang Tugas Akhir 2, yang telah meluangkan waktu, tenaga dan pikiran untuk memberikan kritikan dan masukan yang membangun dalam penyusunan Tugas Akhir ini.
5. Terimakasih **teman** dan **sahabat** untuk setiap doa dan dukungan, untuk setiap motivasi dan penghiburan. Penulis teramat sayang kepada kalian.
6. *Last but not least, to favorite human being, **Joko Widodo, Fiersa Besari, Tulus, and Jerome Polin**. You guys give a tons of positive energy till this time. Thankyou, thankyou, thankyou for teaching to believe in ourself. Wishing can meet you, someday. God bless you. Loves.*

Akhir kata, penulis berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah ikut membantu penulis dalam menyusun Tugas Akhir ini. Semoga Tugas Akhir ini bermanfaat bagi pengembangan ilmu dimasa yang akan datang.

Sitoluama, 3 Agustus 2020

Penulis

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Institut Teknologi Del, saya yang bertanda tangan di bawah ini:

Nama : Yohana Purba
NIM : 11317014
Fakultas/Program Studi : FITE/DIII Teknologi Informasi

Nama : Grace Anastasya Megawati Sihombing
NIM : 11317017
Fakultas/Program Studi : FITE/DIII Teknologi Informasi

Nama : Amsal Marulitua Sianipar
NIM : 11317046
Fakultas/Program Studi : FITE/DIII Teknologi Informasi

Jenis Karya : Tugas Akhir

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Institut Teknologi Del **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul:

IMPLEMENTASI ALGORITME *BACKTRACKING* CSP dan
STEEPEST ASCENT HILL CLIMBING PADA PENJADWALAN
KAPAL PENYEBERANGAN (STUDI KASUS: AJIBATA)

Beserta perangkat yang ada (jika di perlukan). Dengan Hak Bebas Royalti Noneksklusif ini Institut Teknologi Del berhak menyimpan, mengalih/media-format dalam bentuk pangkalan data (*database*), merawat, dan memublikasikan tugas akhir kami selama tetap mencantumkan nama kami sebagai penulis/pencipta dan sebagai pemilik hak cipta.

Demikian pernyataan ini kami buat dengan sebenarnya.

Dibuat di: Laguboti
Pada tanggal: 3 Agustus 2020

Yang menyatakan



(Yohana Purba)



(Grace Sihombing)



(Amsal Sianipar)

ABSTRAK

Penjadwalan diartikan sebagai pengalokasian sejumlah sumber daya (resource) untuk melakukan sejumlah tugas atau operasi dalam jangka waktu tertentu. Dalam kegiatan di pelabuhan sangat diperlukan sistem penjadwalan kapal penyeberangan terkhususnya di pelabuhan Ajibata. Sistem penjadwalan ini merupakan hal yang kompleks dikarenakan perlunya pertimbangan terkait keterbatasan dan syarat yang dimiliki. Keterbatasan dan syarat dalam hal ini berupa nahkoda yang mengemudi kapal, dan jumlah muatan kapal yang disesuaikan dengan jumlah penumpang yang ada didalam kapal tersebut. Penjadwalan kapal juga harus memperhatikan jarak antar pelabuhan untuk meminimalisasikan penggunaan waktu. Kasus penjadwalan kapal yang diambil untuk penelitian Tugas Akhir ini adalah penjadwalan kapal di pelabuhan Ajibata.

Oleh karena itu, untuk mengatasi permasalahan penjadwalan kapal di Pelabuhan Ajibata, maka pada penelitian ini dilakukan pembangunan aplikasi yang menerapkan penggabungan Algoritme *Backtracking* CSP (*Constraint Satisfaction Problem*) dan Algoritme *Steepest Hill Climbing*. Algoritme *Backtracking* dengan *Constraint Satisfaction Problem* dan *Steepest Ascent Hill Climbing*, dimana Algoritme *Backtracking* akan mencoba semua kemungkinan solusi dalam bentuk pohon solusi (*tree*) berdasarkan Constraints yang didapatkan dengan menggunakan pendekatan CSP. Pohon solusi akan ditelusuri secara DFS (*Depth First Search*) untuk mendapatkan solusi terbaik yang diinginkan. Sedangkan Algoritme *Steepest Ascent Hill Climbing* merupakan algoritme yang digunakan untuk penyelesaian masalah jalur terpendek. Gabungan kedua algoritme ini diharapkan dapat mengimplementasikan penugasan setiap sumber daya dan memperoleh jarak antar pelabuhan yang lebih efektif dalam waktu penjadwalan yang beruntung. Sehingga dapat menghasilkan penjadwalan penyeberangan kapal yang tidak bentrok lagi, dengan cara tidak adanya nama kapal yang sama pada keberangkatan yang sama.

Kata kunci: *Penjadwalan kapal, Algoritme backtracking, metode Constraint Satisfaction, Algoritme Steepest Hill Climbing.*

ABSTRACT

Scheduling is defined as the allocation of a number of resources (resources) to perform a number of tasks or operations within a certain period. In the activities at the port, it is very necessary to schedule a ship crossing system, especially at the port of Ajibata. This scheduling system is a complex thing because of the need for consideration related to the limitations and conditions they have. Limitations and conditions in this case in the form of a captain who drives the ship, and the number of cargo that is adjusted to the number of passengers on the ship. Ship scheduling must also pay attention to the distance between ports to minimize the use of time. The case of ship scheduling taken for this Final Project research is ship scheduling at the port of Ajibata. Therefore, to overcome the problem of ship scheduling at the Ajibata Port, this research was carried out in the development of an application that applies the merging of the CSP (Constraint Satisfaction Problem) Algorithm and the Steepest Hill Climbing Algorithm. Backtracking Algorithm with Constraint Satisfaction Problem and Steepest Ascent Hill Climbing, where Backtracking Algorithm will try all possible solutions in the form of a tree based on constraints obtained using the CSP approach. The solution tree will be traced in DFS (Depth First Search) to get the best desired solution. While the Steepest Ascent Hill Climbing Algorithm is an algorithm used to solve the shortest path problem. The combination of these two algorithms is expected to be able to implement the assignment of each resource and obtain a more effective distance between ports in lucky scheduling time. So that it can result in scheduling crossings of ships that do not clash anymore, by means of the absence of the same ship name on the same departure.

Keywords: *Ship Scheduling, Backtracking Algorithm, Constraint Satisfaction Method, Steepest Hill Climbing Algorithm*

DAFTAR ISI

KATA PENGANTAR	4
ABSTRAK	ii
ABSTRACT	iv
DAFTAR ISI	v
DAFTAR TABEL	vii
DAFTAR GAMBAR	viii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	3
1.4 Lingkup	3
1.5 Pendekatan	3
1.6 Sistematika Penyajian	4
BAB II TINJAUAN PUSTAKA	5
2.1 Kapal	5
2.2 Penjadwalan.....	5
2.3 Pelabuhan Ajibata.....	5
2.4 <i>Constraint Satisfaction Problem</i>	6
2.5 Algoritme <i>Backtracking Constraint Satisfaction Problem (CSP)</i>	7
2.6 Lintasan Terpendek.....	8
2.7 Algoritme <i>Steepest Ascent Hill Climbing</i>	9
2.8 Penelitian Terdahulu.....	10
2.9 Kesimpulan	12
BAB III ANALISIS PERMASALAHAN DAN PENGGUNAAN ALGORITME.....	13
3.1 Analisis Penjadwalan Kapal.....	13
3.1.1 Penjadwalan Kapal Penyeberangan di Ajibata	13
3.1.2 Batasan-batasan penjadwalan kapal di pelabuhan Ajibata	14
3.1.3 Data Penjadwalan Kapal	15
3.2 Analisis Algoritme <i>Backtracking (Constraint Satisfaction Problem)</i>	20
3.2.1 Pemodelan Variabel dan <i>Constraints</i>	20
3.2.2 Pewarnaan <i>Graph</i>	28
3.3 Analisis Algoritme <i>Backtracking</i>	29
3.4 Perancangan Sistem	33
3.4.1 Perhitungan jarak menggunakan Algoritme <i>Steepest Ascent Hill Climbing</i>	36
3.4.2 <i>Graph</i> Pencarian Rute Terpendek	37
3.5 Analisis Kebutuhan Sistem.....	41
3.6 <i>Class Diagram</i>	42
3.7 <i>Entity Relationship Diagram (ERD)</i>	42
3.8 Struktur <i>Database</i> Penjadwalan	43
3.10 Antarmuka Perangkat Lunak	45
3.10.1 Antarmuka Pengguna	45
3.10.2 Lingkungan Pengembangan	46
3.11 Deskripsi Fungsional	46
3.11.1 <i>Use Case Diagram</i> Penjadwalan Kapal Penyeberangan	46
3.11.2 <i>Use Case Scenario</i>	49
3.11.2.1 <i>Use Case Scenario</i> Mengelola Data.....	49
3.11.2.2 <i>Use Case Scenario</i> Mengunduh Jadwal	53
3.12 Desain	53
BAB IV IMPLEMENTASI dan TESTING.....	59
4.1 Kebutuhan Implementasi	59
4.1.1 Spesifikasi Perangkat Keras	59
4.1.2 Spesifikasi Perangkat Lunak.....	59
4.2 Hasil Aplikasi.....	59
4.2.1 Halaman Login.....	60

4.2.2	Menu Utama	60
4.2.3	Kelola Kapal	61
4.2.4	Kelola Jadwal	62
4.2.5	Generate Jadwal	63
4.2.6	Kelola Wilayah	64
4.2.7	Kode Program Pemodelan <i>Backtracking Constraint Satisfaction Problem (CSP)</i>	65
4.2.8	Algoritme <i>Steepest Hill Climbing</i>	66
4.3	Testing	67
4.3.1	Metode Test	67
4.3.2	Tujuan Pengujian	67
4.3.3	Test Plan	67
BAB V HASIL DAN PEMBAHASAN		73
5.1	Hasil Jadwal	73
5.2	Hasil Penjadwalan Kapal	73
5.3	Hasil Generate Jadwal Menggunakan Algoritme <i>Steepest Hill Climbing</i>	74
BAB VI KESIMPULAN dan SARAN		77
6.1	Kesimpulan	77
6.2	Saran	77
DAFTAR PUSTAKA		78
LAMPIRAN		vii

DAFTAR TABEL

Table 1. Penelitian terlebih dahulu	11
Table 2. Data Waktu Penyeberangan Kapal Ajibata tahun 2019	15
Table 3. Data Kapal Pelabuhan Ajibata	17
Table 4. Data Nahkoda Kapal Ajibata	18
Table 5. Data Pelabuhan/Wilayah	19
Table 6. Matriks dari Kapal dengan Nahkoda.....	24
Table 7. Relasi antara sesi dengan keberangkatan kapal di Ajibata.....	27
Table 8. Tipe Data Kapal	43
Table 9. Tipe Data Nahkoda	44
Table 10. Tipe Data Sesi.....	44
Table 11. Tipe Data Wilayah	45
Table 12. Tipe Data pelabuhan	45
Table 13. Kebutuhan User Interface.....	45
Table 14. Use Case Scenario Mengelola Data Kapal	49
Table 15. Use Case Scenario Mengelola Data wilayah.....	50
Table 16. Use Case Scenario Mengelola Data Jadwal	51
Table 17. Use Case Scenario Mengelola Generate Jadwal	52
Table 18. Use Case Scenario Membuat Jadwal	53
Table 19. Spesifikasi Perangkat Keras.....	59
Table 20. Spesifikasi Perangkat Lunak	59
Table 21. Test Plan Pengujian Mengelola Data Kapal.....	68
Table 22. Test Plan Pengujian Mengelola Data Wilayah	69
Table 23. Test Plan Pengujian Mengelola Data Jadwal	70
Table 24. Test Plan Pengujian Generate Jadwal	71
Table 25. Pengujian setiap constraint	72

DAFTAR GAMBAR

Gambar 1. Alur Program Algoritma Backtracking CSP	8
Gambar 2. Alur program Algoritme Steepest Ascent Hill Climbing CSP	10
Gambar 3. Pemodelan Tree	29
Gambar 4. Tree Algoritme Backtracking Pertama	31
Gambar 5. Tree Algoritme Backtracing Kedua	32
Gambar 6. Jarak antar pelabuhan pada Google Map	33
Gambar 7. Data Jarak Antar Pelabuhan	34
Gambar 8. Jarak dari Pelabuhan Ajibata ke Tomok.....	35
Gambar 9. Jarak dari Pelabuhan Ajibata ke Tigaras	35
Gambar 10. Jarak dari Pelabuhan Ajibata ke Onanrunggu	36
Gambar 11. Rute Perjalanan dari Ajibata ke Tigaras	37
Gambar 12. Rute Terpendek Ajibata Tigaras.....	38
Gambar 13. Rute Perjalanan dari Ajibata ke Tomok.....	38
Gambar 14. Rute Terpendek Ajibata-Tomok.....	39
Gambar 15. Rute Perjalanan dari Ajibata ke Onanrunggu	40
Gambar 16. Rute Terpendek Ajibata-Onanrunggu	40
Gambar 17. Class Diagram.....	42
Gambar 18. ER-Diagram	43
Gambar 19. Use Case Diagram.....	47
Gambar 20. Mockup Login	54
Gambar 21. Mockup Home	55
Gambar 22. Mockup Kelola Kapal	55
Gambar 23. Mockup Kelola Jadwal	56
Gambar 24. Mockup Kelola Wilayah	57
Gambar 25. Mockup Generate Jadwal	57
Gambar 26. Tampilan Menu Login.....	60
Gambar 27. Tampilan Menu Utama.....	61
Gambar 28. Tampilan Menu Kelola Kapal	62
Gambar 29. Tampilan Menu Kelola Jadwal	63
Gambar 30. Tampilan Menu Generate Jadwal	64
Gambar 31. Tampilan Menu Kelola Wilayah	65
Gambar 32. Kode program Backtracking deklarasi variabel	66
Gambar 33. Kode program <i>Steepest Hill Climbing</i>	66
Gambar 34. Hasil penambahan jadwal kapal	73
Gambar 35. Generate Steepest Hill Climbing	74
Gambar 36. Hasil Export Generate Jadwal	76
Gambar 37. hasil wawancara dari DISHUB	vii
Gambar 38. hasil perhitungan jarak	viii

BAB I

PENDAHULUAN

1.1 Latar Belakang

Kecamatan Ajibata merupakan salah satu kecamatan di Kabupaten Toba Samosir dengan luas wilayah 72, 8 km². Berada pada 2⁰32'- 2⁰40'LU dan 98⁰56' - 99⁰04' BT 908 meter dari permukaan laut. Kecamatan Ajibata memiliki pelabuhan yaitu Pelabuhan Ajibata. Dimana Pelabuhan Ajibata memiliki 3 tujuan, yaitu menuju ke Pulau Samosir, Balige, dan juga Tigaras. Dimana transportasi yang digunakan adalah kapal yang biasa digunakan oleh warga untuk melakukan penyeberangan.

Di Ajibata ada dua jenis pelabuhan yakni regular (untuk kapal-kapal kayu tradisional pengangkut penumpang) dan pelabuhan *ferry* yang menyeberangkan kendaraan, barang maupun orang. Ajibata berbatasan langsung dengan kota Parapat, dimana jalur penyeberangan menuju Pulau Samosir dapat ditempuh melalui tiga rute yakni melalui Pelabuhan Ajibata menuju Tomok, Ajibata menuju Tigaras, Ajibata menuju Onanrunggu.

Pelabuhan Ajibata menyediakan 4 jenis kapal diantaranya kapal ferry, kapal KMP Tao Toba I dan KMP Tao Toba II, dan kapal penumpang. Pada penelitian ini kami akan menggunakan data data kapal penumpang. Pelabuhan Ajibata menuju pelabuhan Tomok beroperasi dengan jumlah 15 trip, memakan waktu sekitar 30 menit. Tapi jika dalam keadaan libur jadwal ini bisa berubah sesuai dengan kepadatan dan jumlah penumpang. Dari pelabuhan Ajibata ke Tigaras menggunakan kapal KMP Sumut I dan Elnusa Pertamina, dengan total trip sebanyak 8 kali dan memakan waktu 1½ jam. Pelabuhan lainnya yaitu Onanrunggu yang juga melayani perjalanan 8 kali, dan menempuh waktu perjalanan sekitar 1½ jam.

Salah satu tugas penting dari pelabuhan adalah membuat jadwal pemberangkatan kapal yang beroperasi setiap hari, dimana jumlah trip kapal yang beroperasi di Pelabuhan Ajibata menuju Tomok, Ajibata menuju Tigaras, dan Ajibata menuju Onanrunggu sebanyak 37 kali dalam sehari. Penjadwalan pemberangkatan kapal yang baik tentu akan menghasilkan pelayanan yang baik dan memuaskan para

penumpang pengguna jasa transportasi air yaitu kapal. Semua kapal yang tersedia di setiap pelabuhan jumlahnya banyak dan terjadwal secara rutin oleh para pegawai di setiap pelabuhan. Akan tetapi, masih ada kendala yang dihadapi dalam proses penyusunan penjadwalan pemberangkatan kapal, dikarenakan pembuatan jadwal masih dilakukan secara manual, dan dengan jumlah kapal yang banyak dan dermaga yang terbatas. Hal tersebut sangat rumit dan tentunya membutuhkan suatu ketelitian dan waktu yang cukup lama. Terlebih pada hari-hari besar (Misalnya perayaan Imlek, perayaan Natal, Tahun Baru, Idul Fitri) yang penumpangnya lebih banyak dibandingkan hari biasa. Sehingga memungkinkan ada nama kapal yang sama dalam sehari beroperasi, ini membuat ketidakadilan pada pembagian jadwal kapal.

Oleh karena itu diperlukan solusi untuk permasalahan penjadwalan di pelabuhan Ajibata saat ini. Pada penelitian ini akan dilakukan penerapan Algoritme Backtracking dengan Constraint Satisfaction Problem dan Steepest Ascent Hill Climbing, dimana Algoritme Backtracking akan mencoba semua kemungkinan solusi dalam bentuk pohon solusi (tree) berdasarkan Constraints yang didapatkan dengan menggunakan pendekatan CSP. Pohon solusi akan ditelusuri secara DFS (Depth First Search) untuk mendapatkan solusi terbaik yang diinginkan. Sedangkan Algoritme Steepest Ascent Hill Climbing merupakan algoritme yang digunakan untuk penyelesaian masalah jalur terpendek. Gabungan kedua algoritme ini diharapkan dapat mengimplementasikan penugasan setiap sumber daya dan memperoleh jarak antar pelabuhan yang lebih efektif dalam waktu penjadwalan yang beruntung. Sehingga dapat menghasilkan penjadwalan penyeberangan kapal yang tidak bentrok lagi, dengan cara tidak adanya nama kapal yang sama pada keberangkatan yang sama.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang ada, rumusan masalah dalam pengerjaan Tugas Akhir ini adalah apakah dengan menggunakan Algoritme *Backtracking CSP* dan *Steepest Ascent Hill Climbing* dapat menghasilkan sistem penjadwalan penyeberangan kapal yang tidak bentrok dan menghasilkan jarak terpendek?

1.3 Tujuan

Tujuan penulisan Tugas Akhir yang akan dicapai dalam Tugas Akhir ini adalah mengimplementasikan Algoritme *Backtracking CSP* dan *Steepest Ascent Hill Climbing* untuk penyusunan penjadwalan penyeberangan kapal yaitu penjadwalan tidak bentrok, dengan cara tidak adanya nama kapal yang sama pada keberangkatan yang sama, dan juga mendapatkan jarak terpendek antar pelabuhan.

1.4 Lingkup

Batasan dari pelaksanaan penelitian ini adalah:

1. Algoritme yang akan digunakan dalam penelitian ini adalah kombinasi dari Algoritme *Backtracking CSP* dan *Steepest Ascent Hill Climbing*.
2. Data yang akan digunakan dalam Tugas Akhir ini adalah data yang diperoleh dari Dinas Perhubungan Toba Samosir. Adapun data tersebut adalah nama kapal, kecepatan kapal, muatan kapal, nama perusahaan, jarak tempuh, durasi waktu, nama nahkoda, dan slot waktu.
3. Penjadwalan kapal tidak memperhatikan cuaca yang terjadi.

1.5 Pendekatan

Beberapa pendekatan yang dilakukan selama pengerjaan Tugas Akhir ini adalah:

1. Studi Literatur

Tahapan studi literatur adalah proses menemukan variable-variabel yang akan diteliti dan mencari sumber kepustakaan yang akan digunakan sebagai referensi penelitian.

2. Pengumpulan Data

Tahapan pengumpulan data adalah proses pengumpulan data yang dibutuhkan, seperti: nama kapal, kecepatan kapal, muatan kapal, nama perusahaan, jarak tempuh, durasi waktu, nama nahkoda, dan slot waktu. Data-data tersebut dibutuhkan yaitu dalam rangka mencapai tujuan penelitian. Teknik pengumpulan data akan dilakukan dengan melakukan wawancara. Wawancara akan dilakukan dengan Dinas Perhubungan Ajibata.

3. Metode Analisis

Tahapan analisis adalah proses pengkajian kebutuhan apa saja yang diperlukan untuk mengatasi permasalahan yang akan diselesaikan dalam pembuatan perangkat lunak.

4. Merancang Sistem

Merancang sistem penjadwalan penyeberangan kapal di Pelabuhan Ajibata yang akan dibuat berdasarkan kebutuhan pengguna dari hasil wawancara.

5. Implementasi Algoritme

Tahapan implementasi algoritme adalah proses menerapkan algoritme pada kasus penjadwalan kapal dalam kode program yang ditulis menggunakan bahasa pemrograman java.

1.6 Sistematika Penyajian

Untuk mempermudah pemahaman dalam pembahasan Tugas Akhir ini, maka penulisan Tugas Akhir ini dibagi menjadi enam Bab.

Bab I Pendahuluan menjelaskan apa yang menjadi topik, lingkup, tujuan, pendekatan yang dilakukan pada Tugas Akhir ini.

Bab II Tinjauan Pustaka menjelaskan mengenai dasar-dasar teori dan tinjauan pustaka yang berhubungan dengan topik yang dikerjakan.

Bab III Analisis menjelaskan mengenai analisis terhadap algoritme yang digunakan serta analisis terhadap permasalahan pada penjadwalan penyeberangan kapal.

Bab IV Implementasi dan Testing menjelaskan uraian hasil implementasi yang telah dilakukan sebelumnya serta proses testing yang telah dilakukan.

Bab V Hasil dan Pembahasan menjelaskan mengenai hasil penjadwalan penyeberangan kapal yang dilakukan dengan menggunakan Algoritma *Backtracking CSP* dan *Steepest Ascent Hill Climbing*.

Bab VI Kesimpulan dan Saran berisi kesimpulan dan saran mengenai Tugas Akhir yang sudah dilakukan dan saran-saran untuk masa yang akan datang.

BAB II

TINJAUAN PUSTAKA

Pada bab ini akan dijelaskan mengenai rangkuman dari teori-teori yang mendukung penelitian yang dibagi menjadi beberapa bagian yaitu mengenai Panjadwalan, Kapal, Pelabuhan Ajibata, Algoritme *Backtracking Constraint Satisfaction Problem (CSP)*, Jalur Terpendek, dan Algoritme *Steepest Ascent Hill Climbing*.

2.1 Kapal

Menurut Wismulyani (2008) kapal adalah suatu alat transportasi yang sering digunakan masyarakat Indonesia untuk dapat saling berhubungan dari satu pulau ke pulau yang lain di seluruh wilayah Indonesia karena Indonesia itu sendiri berbentuk sebagai negara kepulauan yang terdiri dari beberapa pulau. Perahu dan kapal merupakan alat transportasi yang utama di negara Indonesia. Kapal memiliki bentuk, fungsi, dan warna yang beraneka ragam[1].

2.2 Penjadwalan

Penjadwalan dapat diartikan sebagai pengalokasian sejumlah sumber daya (*resource*) untuk melakukan sejumlah tugas atau operasi dalam jangka waktu tertentu dan merupakan proses pengambilan keputusan yang peranannya sangat penting dalam industri manufaktur dan jasa yaitu mengalokasikan sumber-sumber daya yang ada agar tujuan dan sasaran perusahaan lebih optimal[2].

Tujuan penjadwalan adalah untuk mengurangi waktu keterlambatan dari batas waktu yang ditentukan agar dapat memenuhi batas waktu yang telah disetujui dengan konsumen, penjadwalan juga dapat meningkatkan produktifitas mesin dan mengurangi waktu menganggur.

2.3 Pelabuhan Ajibata

Pelabuhan Ajibata adalah pelabuhan menuju Pulau Samosir dengan 3 Rute yakni Ajibata menuju Tomok, Ajibata menuju Tigaras, dan Ajibata menuju Onan runggu. Di Ajibata terdapat dua pelabuhan yaitu reguler untuk kapal-kapal kayu tradisional pengangkut penumpang dan pelabuhan ferry yang menyeberangkan mobil, barang, dan orang dari dan ke Terdapat hubungan antara pelabuhan dengan kota yang digunakan oleh masyarakat setempat untuk bertransaksi maupun bersosialisasi dalam hal memajukan perkembangan pelabuhan itu. Oleh karena kemajuan suatu

pelabuhan maupun kota itu sendiri tidak bisa dilepaskan dari kegiatan perdagangan maupun penyeberangan Pulau Samosir. Pelabuhan disini tidak hanya sebagai tempat bersandarnya modal transportasi kapal. Akan tetapi, menjadikan Pelabuhan Ajibata sebagai pusat kegiatan pemenuhan kebutuhan hidup masyarakat.

Sejak tahun 1972 Pelabuhan Ajibata dikelola secara permanen oleh Dinas Perhubungan Ajibata. Keberadaan pelabuhan ini sangat menunjang perekonomian ataupun perdagangan bagi perkembangan Kecamatan Ajibata dan didukung dengan sarana transportasi darat untuk memperlancar kegiatan pelabuhan, seperti pengangkutan hasil pertanian serta turun naiknya penumpang dari kapal yang berlabuh untuk berbagai kepentingan. Ada dua hal yang disumbangkan oleh pelabuhan untuk meningkatkan perekonomian nasional yaitu berupa pajak yang diberikan kepada pemerintah pusat atau daerah.

Demikian juga secara langsung berupa perolehan pendapatan pada jenis-jenis usaha lain yang dapat dikelola oleh masyarakat di lokasi pelabuhan. Begitu juga tumbuhnya usaha-usaha lain di daerah pelabuhan yang digerakkan oleh adanya aktivitas pelabuhan dan pada gilirannya akan memberikan nilai tambah ekonomi pada daerah sekitar atau belakang pelabuhan. Kegiatan ekonomi yang berlangsung di sekitar pelabuhan Ajibata sejak tahun 1972 hingga tahun 1992 memberikan lapangan pekerjaan bagi masyarakat sekitar khususnya masyarakat Desa Pardamean, Desa Parsaoran, Desa Motung, Desa Lumban Sirait, dan Desa Lumban Gurning. Desa-desa ini terletak di Kecamatan Ajibata. Ada yang bekerja sebagai awak kapal, pedagang, buruh pelabuhan, karyawan dan juga pemberi jasa lainnya seperti calo.

2.4 Constraint Satisfaction Problem

Constraint Satisfaction Problem (CSP) adalah sebuah teknik untuk mendapatkan suatu penyelesaian dari sebuah persoalan melalui pencarian objek atau kondisi yang memenuhi satu atau lebih kriteria

Constraint Satisfaction Problem memiliki tiga komponen yang perlu diperhatikan dalam pendekatannya, antara lain *Constraints*, merupakan suatu aturan yang ditentukan untuk mengatur nilai yang boleh diisikan ke dalam variabel atau kombinasi variabel. Terdapat beberapa jenis *constraints*, diantaranya *unary* (menyatakan persyaratan sepasang variabel), *binary* (menyatakan persyaratan

sepasang variabel), *nary* (menyatakan persyaratan tiga atau lebih variabel), dan *preference* (syarat yang sebaiknya dipenuhi, tetapi tidak harus).

1. Domain, merupakan kumpulan nilai legal diisi ke dalam variabel. Dengan kata lain, sebuah domain akan membatasi nilai suatu variabel.
2. Variabel, merupakan suatu penampung yang dapat diisi dengan berbagai nilai. Biasanya persoalan dimulai di sini, yaitu ketika variabel harus diisi oleh domain yang telah memenuhi *constraints*.

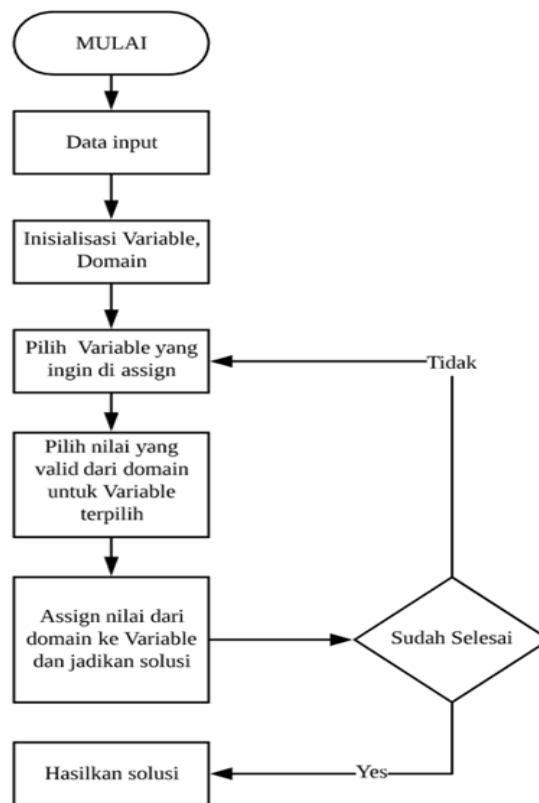
Untuk menentukan isi sebuah variabel yang hendak diisi, dapat dilakukan melalui dua cara yaitu *Most Constrained Variable* Penentuan variabel yang pertama diisi dan berikutnya dimulai dari variabel yang paling banyak mengandung *constraints*.

1. *Least Constrained Variable* Penentuan variabel yang pertama diisi dan berikutnya dimulai dari variabel yang paling sedikit mengandung *constraints*.

2.5 Algoritme *Backtracking Constraint Satisfaction Problem* (CSP)

Algoritme *backtracking* melakukan pencarian solusi persoalan secara sistematis pada semua kemungkinan solusi yang ada pada setiap node dengan berbasis pada *Depth First Search* (DFS) rekursif. DFS merupakan suatu metode pencarian yang dilakukan pada satu node dalam setiap level dari yang paling kiri. Jika solusi belum juga ditemukan, maka pencarian dilanjutkan pada node sebelah kanan. Demikian seterusnya sampai ditemukan solusi atau jika menemukan jalan buntu akan *backtrack* ke posisi sebelumnya. Jika solusi telah ditemukan maka pencarian akan berhenti meskipun ada node yang belum ditelusuri.

Skema umum untuk langkah-langkah pencarian sesuai Algoritma *Backtracking* dapat dilihat pada Gambar 1.



Gambar 1. Alur Program Algoritma Backtracking CSP

2.6 Lintasan Terpendek

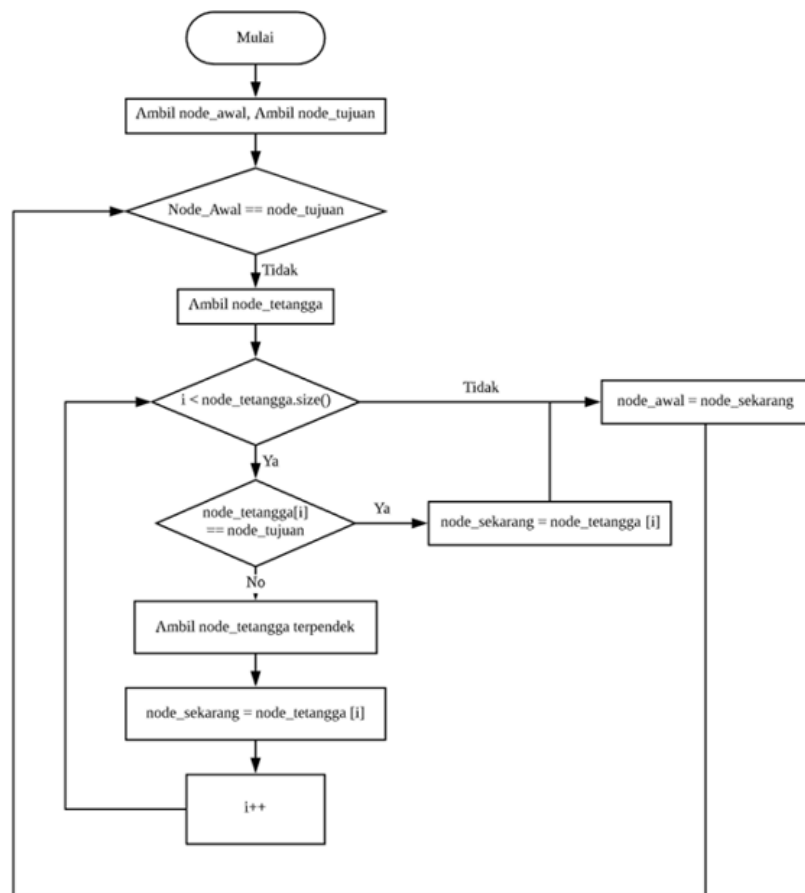
Lintasan atau jalur terpendek adalah jalur minimum yang diperlukan untuk mencapai suatu lokasi ke lokasi lainnya (Yusaputra, 2013). Sedangkan menurut (Hayati & Yohanes, 2014) lintasan terpendek adalah lintasan atau jalur yang dilalui dari satu titik atau node ke titik lainnya dengan bobot yang paling kecil. Tujuan dari pencarian lintasan atau jalur terpendek adalah untuk memperoleh biaya atau jarak yang paling dekat dalam sebuah perjalanan dari lokasi awal ke lokasi tujuan (Hayati & Yohanes, 2014). Dalam pencarian lintasan atau jalur terpendek, terdapat metode atau algoritme yang dapat digunakan. Algoritme pencarian jalur atau rute bertujuan untuk memilih rute paling dekat dari awal ke tempat yang dituju. Algoritme yang digunakan penulis dalam penyelesaian pencarian jalur atau lintasan terpendek pada pengerjaan tugas akhir ini yaitu *Steepest Ascent Hill Climbing*.

2.7 Algoritme *Steepest Ascent Hill Climbing*

Steepest Ascent Hill Climbing merupakan metode algoritme yang banyak digunakan untuk permasalahan optimasi. Salah satu penerapannya adalah untuk mencari rute yang terpendek dengan cara memaksimumkan atau meminimumkan nilai dari fungsi optimasi yang ada. Secara harafiah *steepest* berarti paling tinggi, sedangkan *ascent* berarti kenaikan. Dengan demikian *Steepest Ascent* berarti kenaikan paling tinggi.

Algoritme *Steepest Ascent Hill Climbing*:

1. Evaluasi keadaan awal (*Initial State*). Jika keadaan awal sama dengan tujuan (*Goal state*) maka kembali pada *initial state* dan berhenti berproses. Jika tidak maka *initial state* tersebut jadikan sebagai *current state*.
2. Mulai dengan *current state = initial state*.
3. Dapatkan semua pewaris (*successor*) yang dapat dijadikan *next state* pada *current statenya* dan evaluasi *successor* tersebut dengan fungsi evaluasi dan beri nilai pada setiap *successor* tersebut. Jika salah satu dari *successor* tersebut mempunyai nilai yang lebih baik dari *current state* maka jadikan *successor* dengan nilai yang paling baik tersebut sebagai *new current state*. Lakukan operasi ini terus menerus hingga tercapai *current state = goal state* atau tidak ada perubahan pada *current state* nya.



Gambar 2. Alur program Algoritme Steepest Ascent Hill Climbing CSP

2.8 Penelitian Terdahulu

Pada bagian ini terdapat daftar dari related work atau penelitian yang dilakukan sebelumnya, dapat dilihat pada Tabel 1.

Table 1. Penelitian terlebih dahulu

No	Penulis	Tahun	Tujuan	Algoritme dan Dataset	Hasilnya	Perbedaan dengan TA1D3TI-19	Saran
1.	Rio Febriyana, Wayan Firdaus Mahmudy	2016	Meneliti lebih lanjut mengenai Optimasi Penjadwalan Kapal Penyeberangan Ketapang-Gilimanuk menggunakan Algoritme Genetika	Implementasi menggunakan metode algoritme genetika dengan menggunakan bahasa pemrograman java.	Penelitian ini mampu menghasilkan sebuah aplikasi penjadwalan kapal Ketapang Gilimanuk dan dapat menyelesaikan permasalahan penjadwalan kapal penyeberangan Ketapang-Gilimanuk dengan optimal.	Membuat Penjadwalan kapal di Ajibata dengan menggunakan metode gabungan algoritme Backtracking CSP dan Stepest Ascent Hill Climbing	<ol style="list-style-type: none"> 1. Modifikasi Algoritme Genetika pada fungsi mutasi untuk meningkatkan akurasi yang lebih baik. 2. Tampilan dirancang agar lebih menarik, sederhana, dan konsisiten.
2.	Chandra Ari Gunawan, Hapnes Toba	2016	Penelitian ini bertujuan untuk melakukan penyusunan jadwal otomatis pada pengolaan staf dengan menggunakan pendekatan CSP untuk menghasilkan pembuatan jadwal yang fleksibel.	Pendekatan CSP yaitu pendekatan yang menerapkan sistem pencarian suatu solusi dari masalah dengan cara mencari objek yang dapt memenuhi kriteria dalam penjadwalan tersebut.	<ol style="list-style-type: none"> 1. Menghasilkan penanangan bentrok jadwal dengan memberikan prioritas sesuai jadwal itu sendiri. 2. Memberikan alternatif utnuk menghindari penggunaan teknik optimasi. Implementsi CSP . 		<ol style="list-style-type: none"> 1. Memperbaiki implentasi kelas-kelas dalam aplikasi agar generalisai pembentukan <i>Constraints</i> agar dapat digunakan pada kasus lain. 2. Tampilan dirancang agar lebih menarik, sederhana, dan konsisiten. 3. Apliaksi diharapkan mampu terhubung langsungn dengan sistem akademik .

2.9 Kesimpulan

Penggunaan Algoritme *Backtracking* dalam pencarian solusi terbaik, dengan cara membentuk lintasan dari akar ke daun. *Expand node* yang tidak mengarah ke solusi akan menjadi simpul mati, yang simpulnya tidak akan diperluas lagi. Jika tidak ada simpul *child* maka dilakukan *backtrack* ke simpul induk. Pencarian akan berhenti, apabila telah menemukan solusi atau tidak ada simpul hidup yang diperlukan, sehingga algoritme ini dapat melakukan pencarian solusi penjadwalan dengan seluruh kemungkinan solusi. Algoritme *Steepest Ascent Hill Climbing* digunakan untuk memperoleh solusi pencarian jarak terpendek antar pelabuhan.

BAB III

ANALISIS PERMASALAHAN DAN PENGGUNAAN ALGORITME

3.1 Analisis Penjadwalan Kapal

Pada bab ini dijelaskan analisis dan pengamatan data pada pembuatan jadwal penyeberangan kapal di Ajibata.

3.1.1 Penjadwalan Kapal Penyeberangan di Ajibata

Penjadwalan penyeberangan kapal di Ajibata masih dilakukan secara manual oleh Dinas Perhubungan Ajibata. Selama ini Penjadwalan penyeberangan kapal ditentukan berdasarkan hasil dari perundingan masyarakat pada waktu lampau, sehingga telah ditentukan bahwa 1 hari terdapat 15 sesi penyeberangan kapal dari Ajibata – Tomok mulai dari jam 05.00 – 22.00, dari Ajibata - Tigaras. Namun yang menjadi kendala yaitu pembagian porsi pada setiap kapal tidaklah merata, dikarenakan pemberangkatan kapal masih dilakukan secara acak. Contohnya pada sesi 1 pukul 06.00-08.00 disediakan 2 kapal yang telah ditentukan oleh Dinas Perhubungan Ajibata untuk melakukan penyeberangan kapal pada sesi tersebut, jika kapal sudah penuh, maka Dinas perhubungan Ajibata memanggil kapal yang lain untuk mengangkut penumpang yang masih ada.

Maka dari itu pembagian porsi pada setiap kapal tidaklah merata, dikarenakan 48 kapal yang ada di Ajibata tidaklah digunakan semua oleh Dinas Perhubungan Ajibata. Bahkan ada kapal yang melakukan penyeberangan kapal atau mengangkut penumpang dari Ajibata lebih dari 1 kali, dan ada kapal yang sama sekali tidak melakukan penyeberangan kapal dikarenakan tidak mendapatkan penumpang dari Ajibata. Sehingga dia melakukan penyeberangan kapal di Pelabuhan lain. Di Pelabuhan Ajibata juga memperbolehkan nahkoda memakai kapal nya untuk pengambilan penumpang diluar jadwal yang sudah ditentukan namun jenis penumpangnya yakni rombongan. Dalam hal ini, untuk mengatur penjadwalan kapal di pelabuhan Ajibata. Terdapat sumber daya yang harus diperhatikan, antara lain:

1. Kapal kayu / Kapal penumpang

Kapal kayu / Kapal penumpang merupakan suatu alat transportasi yang digunakan masyarakat Ajibata untuk dapat saling berhubungan dari satu pelabuhan ke pelabuhan yang lain di kabupaten Toba Samosir melewati Danau Toba.

2. Nahkoda

Nahkoda merupakan orang yang dapat mengendarai kapal yang ada di Pelabuhan Ajibata. Sebagian besar Nahkoda yang mengendarai kapal adalah masyarakat Kecamatan Ajibata.

3. Pelabuhan

Pelabuhan merupakan sebuah tempat yang digunakan untuk memuat dan membongkar muatan kapal yang berlabuh. Penambahan Pelabuhan yang ditentukan yaitu pelabuhan dengan jarak terdekat dari pelabuhan sebelumnya.

4. Slot Waktu

Slot Waktu merupakan durasi yang dimiliki dalam proses penyeberangan kapal. Waktu efektif Penyeberangan kapal di Ajibata yaitu 7 hari setiap minggunya dengan sesi perharinya terdiri dari 4 sesi. Dimana 1 sesi terdapat 4 keberangkatan kapal dengan waktu yang berbeda, misalnya sesi 1 keberangkatan 1 dengan kapal DOS ROHA I pada pukul 05.30. sesi 1 keberangkatan 2 dengan kapal DOS ROHA II pada pukul 06.00.

3.1.2 Batasan-batasan penjawalan kapal di pelabuhan Ajibata

Pada bagian ini menjelaskan tentang *hard constraint* pada penjadwalan kapal di pelabuhan Ajibata.

1. Setiap kapal yang berbeda tidak boleh berada pada pelabuhan yang sama dan waktu yang sama. Contohnya, di Pelabuhan Tomok ada kapal KM Horas dan kapal KM Star 03 pada hari senin pukul 06.45
2. Nahkoda tidak boleh mengemudi pada dua kapal atau lebih.
3. Setiap kapal yang berbeda tidak boleh berada pada keberangkatan yang sama dan rute yang sama. Contohnya Kapal Soluna 1 dan Kapal Soluna 2 sama sama berangkat pada pukul 06.00 ke Tomok.
4. Pada setiap sesi tidak boleh terdapat lebih kapal yang berangkat dari yang sudah ditentukan.

3.1.3 Data Penjadwalan Kapal

Data yang digunakan dalam pembuatan jadwal kapal di pelabuhan Ajibata adalah nama kapal, durasi waktu, kecepatan kapal, muatan kapal, dan jarak tempuh. Tugas akhir ini menggunakan data pada tahun 2019 sebagai sumber data yang valid dalam membangun sebuah *prototype* penjadwalan penyeberangan kapal di pelabuhan Ajibata. Selanjutnya akan dilakukan pengujian terhadap sumber daya, sehingga menghasilkan jadwal penyeberangan kapal yang tidak bentrok. Berikut data penyeberangan kapal Ajibata tahun 2019:

1. Data Waktu

Table 2. Data Waktu Penyeberangan Kapal Ajibata tahun 2019

Slot Waktu	Sesi	Keberangkatan	Waktu
1	Sesi 1 Senin	Keberangkatan 1	05.30 – 05.50 WIB
		Keberangkatan 2	06.00 – 06.50 WIB
		Keberangkatan 3	07.00 – 07.50 WIB
		Keberangkatan 4	08.00 – 08.50 WIB
2	Sesi 2 Senin	Keberangkatan 1	09.00 – 09.50 WIB
		Keberangkatan 2	10.00 – 10.50 WIB
		Keberangkatan 3	11.00 – 11.50 WIB
		Keberangkatan 4	12.00 – 12.50 WIB
3	Sesi 3 Senin	Keberangkatan 1	13.00 – 13.50 WIB
		Keberangkatan 2	14.00 – 14.50 WIB
		Keberangkatan 3	15.00 – 15.50 WIB
		Keberangkatan 4	16.00 – 16.50 WIB
4	Sesi 4 Senin	Keberangkatan 1	17.00 – 17.50 WIB
		Keberangkatan 2	18.00 – 20.50 WIB
5	Sesi 1 Selasa	Keberangkatan 1	05.30 – 05.50 WIB
		Keberangkatan 2	06.00 – 06.50 WIB
		Keberangkatan 3	07.00 – 07.50 WIB
		Keberangkatan 4	08.00 – 08.50 WIB
6	Sesi 2 Selasa	Keberangkatan 1	09.00 – 09.50 WIB
		Keberangkatan 2	10.00 – 10.50 WIB
		Keberangkatan 3	11.00 – 11.50 WIB
		Keberangkatan 4	12.00 – 12.50 WIB
7	Sesi 3 Selasa	Keberangkatan 1	13.00 – 13.50 WIB
		Keberangkatan 2	14.00 – 14.50 WIB
		Keberangkatan 3	15.00 – 15.50 WIB
		Keberangkatan 4	16.00 – 16.50 WIB
8	Sesi 4 Selasa	Keberangkatan 1	17.00 – 17.50 WIB
		Keberangkatan 2	18.00 – 20.50 WIB
9	Sesi 1 Rabu	Keberangkatan 1	05.30 – 05.50 WIB
		Keberangkatan 2	06.00 – 06.50 WIB
		Keberangkatan 3	07.00 – 07.50 WIB
		Keberangkatan 4	08.00 – 08.50 WIB

Slot Waktu	Sesi	Keberangkatan	Waktu
10	Sesi 2 Rabu	Keberangkatan 1	09.00 – 09.50 WIB
		Keberangkatan 2	10.00 – 10.50 WIB
		Keberangkatan 3	11.00 – 11.50 WIB
		Keberangkatan 4	12.00 – 12.50 WIB
11	Sesi 3 Rabu	Keberangkatan 1	13.00 – 13.50 WIB
		Keberangkatan 2	14.00 – 14.50 WIB
		Keberangkatan 3	15.00 – 15.50 WIB
		Keberangkatan 4	16.00 – 16.50 WIB
12	Sesi 4 Rabu	Keberangkatan 1	17.00 – 17.50 WIB
		Keberangkatan 2	18.00 – 20.50 WIB
13	Sesi 1 Kamis	Keberangkatan 1	05.30 – 05.50 WIB
		Keberangkatan 2	06.00 – 06.50 WIB
		Keberangkatan 3	07.00 – 07.50 WIB
		Keberangkatan 4	08.00 – 08.50 WIB
14	Sesi 2 Kamis	Keberangkatan 1	09.00 – 09.50 WIB
		Keberangkatan 2	10.00 – 10.50 WIB
		Keberangkatan 3	11.00 – 11.50 WIB
		Keberangkatan 4	12.00 – 12.50 WIB
15	Sesi 3 Kamis	Keberangkatan 1	13.00 – 13.50 WIB
		Keberangkatan 2	14.00 – 14.50 WIB
		Keberangkatan 3	15.00 – 15.50 WIB
		Keberangkatan 4	16.00 – 16.50 WIB
16	Sesi 4 Kamis	Keberangkatan 1	17.00 – 17.50 WIB
		Keberangkatan 2	18.00 – 20.50 WIB
17	Sesi 1 Jumat	Keberangkatan 1	05.30 – 05.50 WIB
		Keberangkatan 2	06.00 – 06.50 WIB
		Keberangkatan 3	07.00 – 07.50 WIB
		Keberangkatan 4	08.00 – 08.50 WIB
18	Sesi 2 Jumat	Keberangkatan 1	09.00 – 09.50 WIB
		Keberangkatan 2	10.00 – 10.50 WIB
		Keberangkatan 3	11.00 – 11.50 WIB
		Keberangkatan 4	12.00 – 12.50 WIB
19	Sesi 3 Jumat	Keberangkatan 1	13.00 – 13.50 WIB
		Keberangkatan 2	14.00 – 14.50 WIB
		Keberangkatan 3	15.00 – 15.50 WIB
		Keberangkatan 4	16.00 – 16.50 WIB
20	Sesi 4 Jumat	Keberangkatan 1	17.00 – 17.50 WIB
		Keberangkatan 2	18.00 – 20.50 WIB
21	Sesi 1 Sabtu	Keberangkatan 1	05.30 – 05.50 WIB
		Keberangkatan 2	06.00 – 06.50 WIB
		Keberangkatan 3	07.00 – 07.50 WIB
		Keberangkatan 4	08.00 – 08.50 WIB
22	Sesi 2 Sabtu	Keberangkatan 1	09.00 – 09.50 WIB
		Keberangkatan 2	10.00 – 10.50 WIB
		Keberangkatan 3	11.00 – 11.50 WIB
		Keberangkatan 4	12.00 – 12.50 WIB

23	Sesi 3 Sabtu	Keberangkatan 1	13.00 – 13.50 WIB
		Keberangkatan 2	14.00 – 14.50 WIB

Slot Waktu	Sesi	Keberangkatan	Waktu
		Keberangkatan 3	15.00 – 15.50 WIB
		Keberangkatan 4	16.00 – 16.50 WIB
24	Sesi 4 Sabtu	Keberangkatan 1	17.00 – 17.50 WIB
		Keberangkatan 2	18.00 – 20.50 WIB
25	Sesi 1 Minggu	Keberangkatan 1	05.30 – 05.50 WIB
		Keberangkatan 2	06.00 – 06.50 WIB
		Keberangkatan 3	07.00 – 07.50 WIB
		Keberangkatan 4	08.00 – 08.50 WIB
26	Sesi 2 Minggu	Keberangkatan 1	09.00 – 09.50 WIB
		Keberangkatan 2	10.00 – 10.50 WIB
		Keberangkatan 3	11.00 – 11.50 WIB
		Keberangkatan 4	12.00 – 12.50 WIB
27	Sesi 3 Minggu	Keberangkatan 1	13.00 – 13.50 WIB
		Keberangkatan 2	14.00 – 14.50 WIB
		Keberangkatan 3	15.00 – 15.50 WIB
		Keberangkatan 4	16.00 – 16.50 WIB
28	Sesi 4 Minggu	Keberangkatan 1	17.00 – 17.50 WIB
		Keberangkatan 2	18.00 – 20.50 WIB

Tabel 2 merupakan data waktu yang digunakan untuk penyeberangan kapal di Ajibata yaitu 28 *slot* waktu. Jumlah *slot* waktu sebanyak 28 diperoleh dari 4 *slot* waktu dari hari senin sampai hari jumat.

Catatan: Sesi 4 hari Senin sampai hari Minggu hanya ada 2 keberangkatan kapal.

2. Data Kapal

Table 3. Data Kapal Pelabuhan Ajibata

No.	Nama Kapal	Muatan Kapal	Nama Perusahaan
1.	DOS ROHA I	50 orang	OPS. Tomok Tour
2.	DOS ROHA II	50 orang	OPS. Tomok Tour
3.	DOS ROHA III	50 orang	OPS. Tomok Tour
4.	DOS ROHA V	50 orang	OPS. Tomok Tour
5.	RODAME III	50 orang	OPS. Tomok Tour
6.	LEO START	50 orang	OPS. Tomok Tour
7.	RODAME I	50 orang	OPS. Tomok Tour
8.	GLORIA	50 orang	OPS. Tomok Tour
9.	RUDI	50 orang	OPS. Tomok Tour
10.	MURNI	50 orang	OPS. Tomok Tour
11.	SILIMA TALI	50 orang	OPS. Tomok Tour
12.	TIO TOUR	50 orang	OPS. Tomok Tour
13.	PULO HORAS	50 orang	OPS. Tomok Tour
14.	ROGANDA	50 orang	OPS. Tomok Tour
15.	Soluna 01	50 orang	PT.ASDP

16.	Soluna 02	50 orang	PT.ASDP
17.	Lamhot	50 orang	PT.Dok Bahari Nusantara
18.	Doruli 02	50 orang	PT. Aquafarm
19.	Doruli 03	50 orang	PT. Aquafarm
20.	Holden 02	50 orang	Putu Sumarjaya
21.	Holden 01	50 orang	Putu Sumarjaya
22.	Horas 01	50 orang	PT.ASDP
23.	Horas 02	50 orang	PT.ASDP
24.	Yosuaris 01	50 orang	PT.Dok Bahari Nusantara
25.	Yosuaris 02	50 orang	PT.Dok Bahari Nusantara
26.	Grace 3	50 orang	PT. Gunung Hijau Megah
27.	HORAS	50 orang	OPS. Marihat Permai
28.	STAR 03	50 orang	OPS. Marihat Permai
29.	AUSTIN 01	50 orang	OPS. Marihat Permai
30.	TELAGA BIRU	50 orang	OPS. Marihat Permai
31.	STAR 04	50 orang	OPS. Marihat Permai
32.	AUSTIN 17	50 orang	OPS. Marihat Permai
33.	GLANTER	50 orang	OPS. Marihat Permai
34.	SINAR TOBA 7	50 orang	OPS. Marihat Permai
35.	SINAR TOBA 6	50 orang	OPS. Marihat Permai
36.	METHA	50 orang	OPS. Marihat Permai
37.	EKO 92	50 orang	OPS. Marihat Permai
38.	NATIO 1	50 orang	OPS. Marihat Permai
39.	KANRO	50 orang	OPS. Marihat Permai
40.	SARINA	50 orang	OPS. Marihat Permai
41.	ADINDA	50 orang	OPS. Marihat Permai
42.	ARIMBI	50 orang	OPS. Marihat Permai

Tabel 3 merupakan data kapal yang digunakan dalam pembuatan *prototype* di Ajibata. Data kapal terdiri dari 42 kapal pada tahun 2019.

3. Data Nahkoda

Table 4. Data Nahkoda Kapal Ajibata

No.	Nama Nahkoda	Inisial	Nama kapal yang dikemudi
1.	Arie Gultom	AG	Else
2.	Parulian Sinaga	PS	Marsada Holong
3.	Putra Ambarita	PA	Sinta Maju
4.	Maulana Limbong	ML	Toba Odeboer
5.	Rico Lumbanraja	RL	Marta Gultom
6.	Raja Malau	RM	Demos
7.	Krintianto Hutahaeen	KH	Pardomuan
8.	Betman Malau	BM	Sapadia
9.	Budiman Rumahorbo	BR	Rilu
10.	Pudan Parapat	PP	Nainggolan 8A
11.	Febi Samosir	FS	Petrus Siantury
12.	Soon Parapat	SP	Bunda
13.	Budianto Sidauruk	BS	Roma Parsauran

14.	Joko Simalango	JS	Sampurna
15.	Dikki Sidabutar	DS	Rogabe
16.	Septa Simalango	SS	Soluna 01
17.	Martin Gultom	MG	Soluna 02
18.	Kristian Saragi	KS	Lamhot
19.	Andika Samosir	AS	Doruli 02
20.	Joni Hutagaol	JH	Doruli 03
21.	Jojo Naibaho	JN	Holden 02
22.	Lamhot Nainggolan	LN	Holden 01
23.	Dohar Nadapdap	DN	Horas 01
24.	Manahan Nainggolan	MN	Horas 02
25.	Andi Gultom	NG	Yosuaris 01
26.	Bornok Simbolon	BB	Yosuaris 02
27.	Dedi Situngkir	DT	Grace 3
28.	Hasudungan Sihotang	HS	Dosroha 01
29.	Kevin Sigalingging	KG	Dosroha 02
30.	Jecky Sitinjak	JT	Dosroha 03
31.	Bastian Tamba	BT	Dosroha 05
32.	Johan Simarmata	JM	Leostar
33.	Jojo Situmorang	JO	Murni
34.	Pangihutan Simbolon	PS	Rudy Star
35.	Linggom Sitinjak	LS	Pulo Horas
36.	Charles Situngkir	CS	Tio Tour
37.	Yosef Malau	YM	Gloria
38.	Lambok Siboro	LB	Leostar
39.	Danang Sidauruk	DD	Rodame 01
40.	Mikael Simanjuntak	MS	Rodame 02
41.	Josua Simarmata	JM	Tioly
42.	Torang Gultom	TG	Roganda

Tabel 4 merupakan data nahkoda kapal yang digunakan dalam pembuatan *prototype* di Ajibata. Data Nahkoda dengan inisialnya terdiri dari 42 kapal pada tahun 2019.

4. Data Pelabuhan/Wilayah

Table 5. Data Pelabuhan/Wilayah

No.	Wilayah awal	Wilayah tujuan	Jarak tempuh	Durasi
1.	Ajibata	Tomok	8,43 km	30 menit
3.	Ajibata	Onanrunggu	23,76	1 jam 25 menit
4.	Ajibata	Tigaras	22,60	1 jam 21 menit

Tabel 5 merupakan data wilayah yaitu rute perjalanan kapal yang digunakan dalam pembuatan *prototype* di Ajibata pada tahun 2019, dan data tersebut didapat dari hasil wawancara langsung terhadap Dinas Perhubungan Ajibata. Hasil wawancara dapat dilihat didalam lampiran.

3.2 Analisis Algoritme *Backtracking* (*Constraint Satisfaction Problem*)

CSP merupakan metode yang digunakan dalam pembuatan solusi-solusi yang memungkinkan dan tidak melanggar *constraints* yang ada, agar tidak ada *constraints* yang dilanggar dalam pembuatan jadwal. CSP menggunakan sistem penggolongan *constraints* berdasarkan aturan, bahwa *constraints* yang ada, akan dipisahkan antara *constraint* yang sejenis. *Constraint* yang dibuat, bertujuan untuk menghindari kemungkinan bentrok. Berikut dibahas mengenai *constraints* yang telah dibentuk menggunakan CSP.

1. Membuat batas jam sesi dalam sehari, sejumlah tujuh belas jam.
2. Memisahkan kapal berdasarkan tujuan pelabuhannya.
3. Kapal dengan tujuan pelabuhan akan dipisahkan berdasarkan *constraints* yang telah ditentukan. Misalnya, dalam 1 sesi tersebut terdapat 4 kapal yang akan melakukan penyeberangan, tidak boleh lebih dari 4 kapal yang akan melakukan penyeberangan.
4. Kapal tidak boleh berada pada Keberangkatan dengan tujuan pelabuhan yang sama.

3.2.1 Pemodelan Variabel dan *Constraints*

Pemodelan variable pada penjadwalan penyeberangan kapal merupakan penampung yang terdiri domain (*value*) yang sesuai dengan variabel itu sendiri. Variabel ini akan diinput oleh *user* untuk menentukan jumlah domain setiap variabel. Domain akan disimpan dalam bentuk *array*.

1. Kapal

$K = \{ \text{DOS ROHA I, DOS ROHA II, DOS ROHA III, DOS ROHA IV, RODAME I, RODAME III, LEO START, GLORIA, RUDI, MURNI} \}$

$K[0] = \text{DOS ROHA I}$

$K[6] = \text{LEO START}$

$K[1] = \text{DOS ROHA II}$

$K[7] = \text{GLORIA}$

$K[2] = \text{DOS ROHA III}$

$K[8] = \text{RUDI}$

$K[3] = \text{DOS ROHA V}$

$K[9] = \text{MURNI}$

$K[4] = \text{RODAME I}$

$K[5] = \text{RODAME III}$

2. Nahkoda

$N = \{\text{HS, KS, JS, BT, DS, MS, LS, YM, PS, JSG}\}$

$N[0] = \text{HS}$

$N[6] = \text{LS}$

$N[1] = \text{KS}$

$N[7] = \text{YM}$

$N[2] = \text{JS}$

$N[8] = \text{PS}$

$N[3] = \text{BT}$

$N[9] = \text{JSG}$

$N[4] = \text{DS}$

$N[5] = \text{MS}$

3. Wilayah

$W = \{\text{Tomok, Tigaras, Onan runggu}\}$

$W[0] = \text{Tomok}$

$W[1] = \text{Tigaras}$

$W[2] = \text{Onan Runggu}$

4. Hari

$H = \{\text{Senin, Selasa, Rabu, Kamis, Jumat, Sabtu, Minggu}\}$

$H[0] = \text{Senin}$

$H[1] = \text{Selasa}$

$H[2] = \text{Rabu}$

$H[3] = \text{Kamis}$

$H[4] = \text{Jumat}$

$H[5] = \text{Sabtu}$

$H[6] = \text{Minggu}$

5. Sesi

$$S = \{1, 2, 3, 4\}$$

$$S[0] = 1$$

$$S[1] = 2$$

$$S[2] = 3$$

$$S[3] = 4$$

6. Keberangkatan

$$KB = \{\text{Keberangkatan I, Keberangkatan II, Keberangkatan III, Keberangkatan IV}\}$$

$$KB[0] = \text{Keberangkatan I}$$

$$KB[1] = \text{Keberangkatan II}$$

$$KB[2] = \text{Keberangkatan III}$$

$$KB[3] = \text{Keberangkatan IV}$$

Constraints pada penjadwalan terbentuk dari keterkaitan antar domain yang terdiri dari batasan-batasan yang terletak pada domain. Relasi domain akan bernilai satu jika memiliki keterkaitan, dan akan bernilai nol jika tidak memiliki keterkaitan.

1. Kapal-Nahkoda (KN)

Pada bagian ini relasi Nahkoda dengan kapal dijelaskan menggunakan tabel matriks yang berisi array masing-masing nahkoda dan kapal yang dikendarai.

Berikut merupakan contoh relasi domain yang bernilai satu dan bernilai nol:

a. Relasi domain yang bernilai satu:

Nahkoda dengan inisial HS pada array ke 0 dimodelkan sebagai $N[0] = \text{HS}$, dan Kapal DOS ROHA I pada array ke 0 dimodelkan $K[0] = \text{DOS ROHA I}$. Sehingga hasil pemodelan yang diperoleh antara nahkoda dengan kapal yaitu $KN[0][0] = 1 = KN[\text{HS}][\text{DOS ROHA I}]$, artinya relasi domain bernilai satu

karena memiliki keterkaitan yaitu Nahkoda HS mengendarai kapal DOS ROHA I.

b. Relasi domain yang bernilai nol:

Nahkoda dengan inisial HS pada array ke 0 dimodelkan sebagai $N[0] = HS$, dan Kapal DOS ROHA II pada array ke 1 dimodelkan $K[1] = DOS ROHA II$. Sehingga hasil pemodelan yang diperoleh antara nahkoda dengan kapal yaitu $KN[0][0] = 0 = KN[HS][DOS ROHA II]$, artinya relasi domain bernilai satu karena memiliki keterkaitan yaitu Nahkoda HS mengendarai kapal DOS ROHA II.

Berikut merupakan relasi secara keseluruhan antara nahkoda yang mengendarai kapal di Ajibata.

Table 6. Matriks dari Kapal dengan Nahkoda

Kapal	Nahkoda																			
	HS [0]	KS [1]	JS [2]	BT [3]	DS [4]	MS [5]	LS [6]	YM [7]	PS [8]	JSG [9]	LSK [10]	CS [11]	SS [12]	MG [13]	KSI [14]	AS [15]	JH [16]	LG [17]	JNO [18]	DN [19]
DOS ROHA I [0]	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DOS ROHA II [1]	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DOS ROHA III [2]	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DOS ROHA V [3]	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RODAME III [4]	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LEO START [5]	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
RODAME I [6]	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
GLORIA [7]	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
RUDI [8]	0	0	0	0	0	0	0	0	1		0	0	0	0	0	0	0	0	0	0
MURNI [9]	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
PULO HORAS [10]	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
TIO TOUR [11]	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
SOLUNA 1 [12]	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
SOLUNA 2 [13]	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
LAMHOT [14]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
DORULI 2 [15]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
DORULI 3 [16]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

HOLDEN 1 [17]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Kapal	Nahkoda																			
	HS [0]	KS [1]	J S [2]	BT [3]	DS [4]	MS [5]	L S [6]	Y M [7]	PS [8]	JSG [9]	LSK [10]	CS [11]	SS [12]	MG [13]	KSI [14]	A S [1 5]	J H [1 6]	LG [17]	JNO [18]	DN [19]
HOLDEN 2 [18]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
HORAS 1 [19]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
HORAS 2 [20]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ROGANDA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Berdasarkan tabel matrik antara nahkoda dan kapal diatas, maka berikut merupakan keterkaitan atau bernilai 1:

$KN[0][0] = 1 = KN[DOS ROHA I][HS]$
 $KN[1][1] = 1 = KN[DOS ROHA II][KS]$
 $KN[2][2] = 1 = KN[DOS ROHA III][JS]$
 $KN[3][3] = 1 = KN[DOS ROHA IV][BT]$
 $KN[4][4] = 1 = KN[RODAME III][DS]$
 $KN[5][5] = 1 = KN[LEO START][MS]$
 $KN[6][6] = 1 = KN[RODAME I][LS]$
 $KN[7][7] = 1 = KN[GLORIA][YM]$
 $KN[8][8] = 1 = KN[RUDI][PS]$
 $KN[9][9] = 1 = KN[MURNI][JSG]$
 $KN[10][10] = 1 = KN[PULO HORAS][LSK]$
 $KN[11][11] = 1 = KN[TIO TOUR][CS]$
 $KN[12][12] = 1 = KNSOLUNA 1][SS]$
 $KN[13][13] = 1 = KN[SOLUNA 2][MG]$
 $KN[14][14] = 1 = KN[LAMHOT][KSI]$
 $KN[15][15] = 1 = KN[DORULI 2][AS]$
 $KN[16][16] = 1 = KN[DORULI 3][JH]$
 $KN[17][17] = 1 = KN[HOLDEN 1][LG]$
 $KN[18][18] = 1 = KN[HOLDEN 2][JNO]$
 $KN[19][19] = 1 = KN[HORAS 1][DN]$
 $KN[20][20] = 1 = KN[HORAS 2][MN]$
 $KN[21][21] = 1 = KN[ROGANDA][TG]$

2. Sesi – Keberangkatan kapal (SK)

Pada bagian ini relasi antara sesi dan keberangkatan kapal dijelaskan menggunakan tabel matriks yang berisi array masing-masing sesi dengan keberangkatan kapal. Berikut merupakan contoh relasi kapal dengan keberangkatan kapal dengan domain bernilai satu dan bernilai nol:

- Relasi domain yang bernilai satu :

Sesi dengan inisial Sesi 1 pada array ke 0 dimodelkan sebagai $S[0]$ = Sesi 1, dan Keberangkatan kapal dengan array ke 0 dimodelkan sebagai $K[0]$ = Keberangkatan I Sehingga hasil pemodelan yang diperoleh antara Sesi dengan Keberangkatan kapal yaitu $SK[0][0] = 1 = SK[\text{Sesi 1}][\text{Keberangkatan I}]$, artinya relasi domain bernilai satu karena memiliki keterkaitan yaitu Sesi 1 dengan Keberangkatan kapal I.

- Relasi domain yang bernilai nol:

Sesi dengan inisial Sesi 4 pada array ke 3 dimodelkan sebagai $S[3]$ = Sesi 4, dan Keberangkatan kapal dengan array ke 3 dimodelkan sebagai $K[2]$ = Keberangkatan III Sehingga hasil pemodelan yang diperoleh antara Sesi dengan Keberangkatan kapal yaitu $SK[4][2] = 0 = SK[\text{Sesi 4}][\text{Keberangkatan III}]$, artinya relasi domain bernilai 0 karena tidak memiliki keterkaitan antara Sesi 4 dengan Keberangkatan kapal III.

Berikut merupakan relasi secara keseluruhan antara sesi dengan Keberangkatan kapal di Ajibata:

Table 7. Relasi antara sesi dengan keberangkatan kapal di Ajibata

SESI	Keberangkatan kapal			
	Keberangkatan I [0]	Keberangkatan III [1]	Keberangkatan III [2]	Keberangkatan IV [3]
SESI 1 [0]	1	1	1	1
SESI 2 [1]	1	1	1	1
SESI 3 [2]	1	1	1	1
SESI 4 [3]	1	1	0	0

Berdasarkan tabel matriks antara sesi dan keberangkatan kapal diatas, maka berikut merupakan keterkaitan atau bernilai 1:

$SK[0][0] = 1 = SK[\text{Sesi 1}][\text{Keberangkatan I}]$

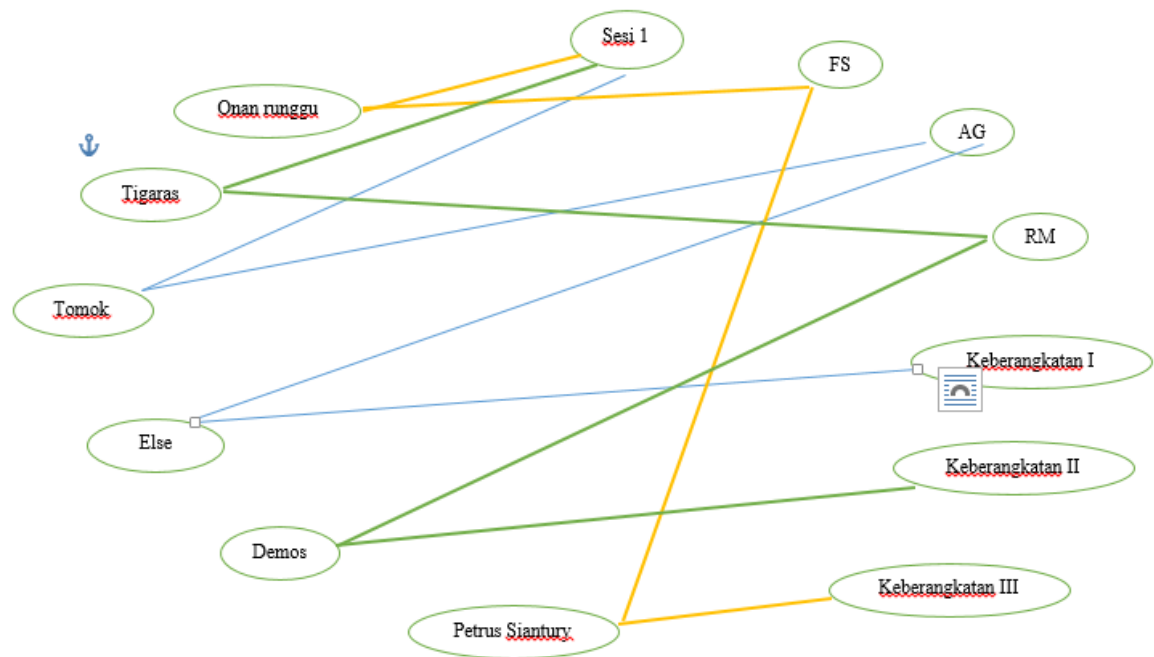
$SK[0][1] = 1 = SK[\text{Sesi 1}][\text{Keberangkatan II}]$

$SK[0][2] = 1 = SK[\text{Sesi 1}][\text{Keberangkatan III}]$

$SK[0][3] = 1 = SK[\text{Sesi 1}][\text{Keberangkatan IV}]$
 $SK[1][0] = 1 = SK[\text{Sesi 2}][\text{Keberangkatan I}]$
 $SK[1][1] = 1 = SK[\text{Sesi 2}][\text{Keberangkatan II}]$
 $SK[1][2] = 1 = SK[\text{Sesi 2}][\text{Keberangkatan III}]$
 $SK[1][3] = 1 = SK[\text{Sesi 2}][\text{Keberangkatan IV}]$
 $SK[2][0] = 1 = SK[\text{Sesi 3}][\text{Keberangkatan I}]$
 $SK[2][1] = 1 = SK[\text{Sesi 3}][\text{Keberangkatan II}]$
 $SK[2][2] = 1 = SK[\text{Sesi 3}][\text{Keberangkatan III}]$
 $SK[2][3] = 1 = SK[\text{Sesi 3}][\text{Keberangkatan IV}]$
 $SK[3][0] = 1 = SK[\text{Sesi 4}][\text{Keberangkatan I}]$
 $SK[3][1] = 1 = SK[\text{Sesi 4}][\text{Keberangkatan II}]$

3.2.2 Pewarnaan *Graph*

Constraints yang ada dapat dijelaskan menggunakan pewarnaan graph. Pewarnaan graph memiliki syarat yang harus diperhatikan. Setiap pewarnaan yang dimiliki harus merujuk pada *constraints* yang dimiliki. Untuk pewarnaan *constraints*, setiap node yang dihubungkan dengan node lain untuk tujuan yang sama harus memiliki warna yang berbeda. Graph tersebut menjelaskan antara keterkaitan antar sumber daya dan pembentukan *constraints* berdasarkan tugas yang dimiliki. Berikut merupakan pewarnaan graph untuk *constraints* yang telah dibentuk berdasarkan contoh pemodelan tree pada Gambar 3 berikut.



Gambar 3. Pemodelan Tree

3.3 Analisis Algoritme *Backtracking*

Algoritme *Backtracking* digunakan untuk menghasilkan jadwal yang tidak mengalami bentrok antara kapal yang satu dengan yang lain. Algoritme *Backtracking* dibuat dalam bentuk pohon (*tree*) solusi. Untuk mendapatkan solusi terbaik, pohon solusi tersebut akan ditelusuri secara DFS (*Depth First Search*). Penggunaan Algoritme *Backtracking* dilakukan dengan tahapan logika sebagai berikut:

1. Penelusuran keseluruhan *node* dimulai dari akar hingga ke daun. Simpul yang lahir disebut simpul hidup dan dapat diperluas dengan diberi nama simpul-E. Penelusuran ini dilakukan berdasarkan *tree* yang telah tercipta dari solusi-solusi yang terkumpul dari metode CSP yang telah ada sebelumnya.
2. Apabila lintasan yang didapatkan dari perluasan simpul-E tidak mengarah ke solusi, maka simpul tersebut akan menjadi simpul mati dan tidak akan dilakukan perluasan lagi.

3. Apabila posisi saat ini berada pada *node* yang merupakan simpul mati, maka akan dilakukan pencarian selanjutnya dengan membangkitkan simpul anak yang lain. Namun, jika tidak ada simpul *child* maka akan dilakukan *Backtracking* ke simpul induk.
4. Proses ini akan berhenti apabila telah menemukan solusi terbaik atau tidak ada lagi simpul hidup yang diperlukan.

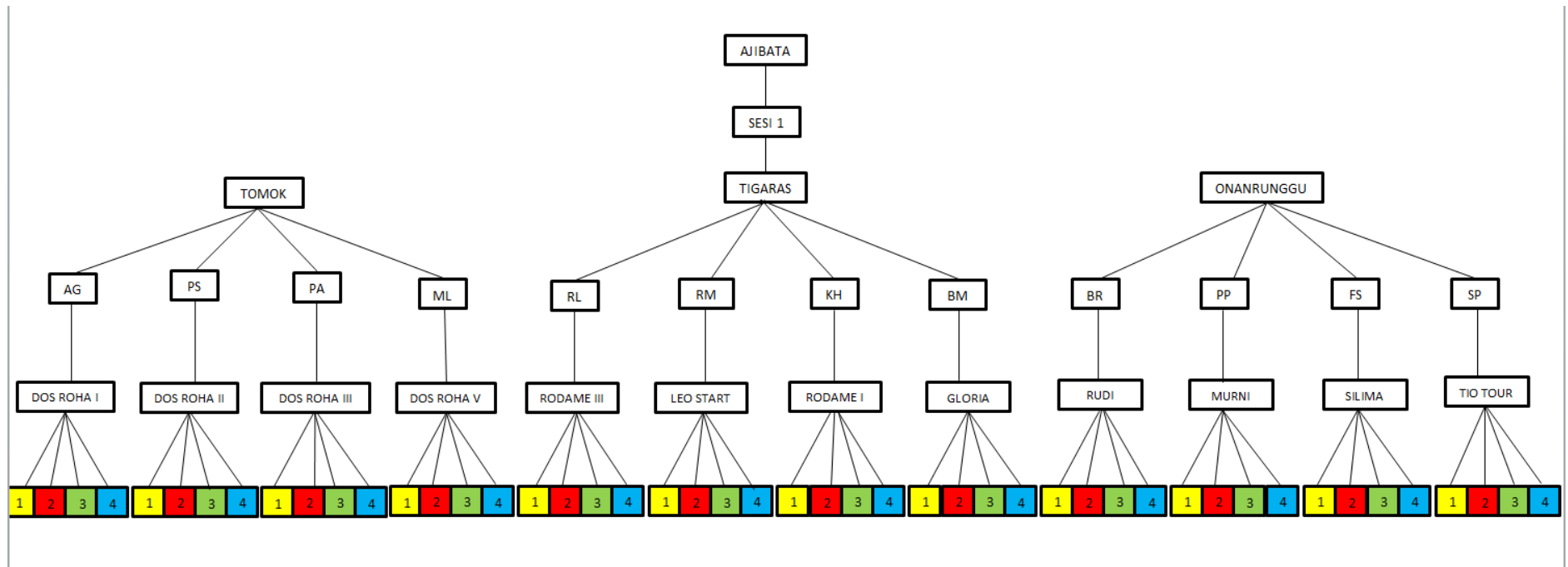
Untuk penjadwalan kapal penyeberangan, akan dilakukan terlebih dahulu pencari solusi. Pencarian solusi dapat disimulasikan sebagai berikut:

- SESI I,TOMOK,AG,DOS ROHA I
- SESI I,TOMOK,PS,DOS ROHA II
- SESI I,TOMOK,PA,DOS ROHA III
- SESI I,TOMOK,ML,DOS ROHA V
- SESI I,TIGARAS,RL,RODAME III
- SESI I,TIGARAS,RM,LEO START
- SESI I,TIGARAS,KH,RODAME I
- SESI I,TIGARAS,BM,GLORIA
- SESI I,ONANRUNGGU,BR,RUDI
- SESI I,ONANRUNGGU,PP,MURNI
- SESI I,ONANRUNGGU,FS,SILIMA TALI
- SESI I,ONANRUNGGU,SP,TIO TOUR

Dalam menghasilkan jadwal yang sesuai, maka Algoritme *Backtracking* harus memperhatikan *constraints* yang telah dibentuk menggunakan CSP, untuk simulasi yang dilakukan, perlu memperhatikan *constraints* sebagai berikut:

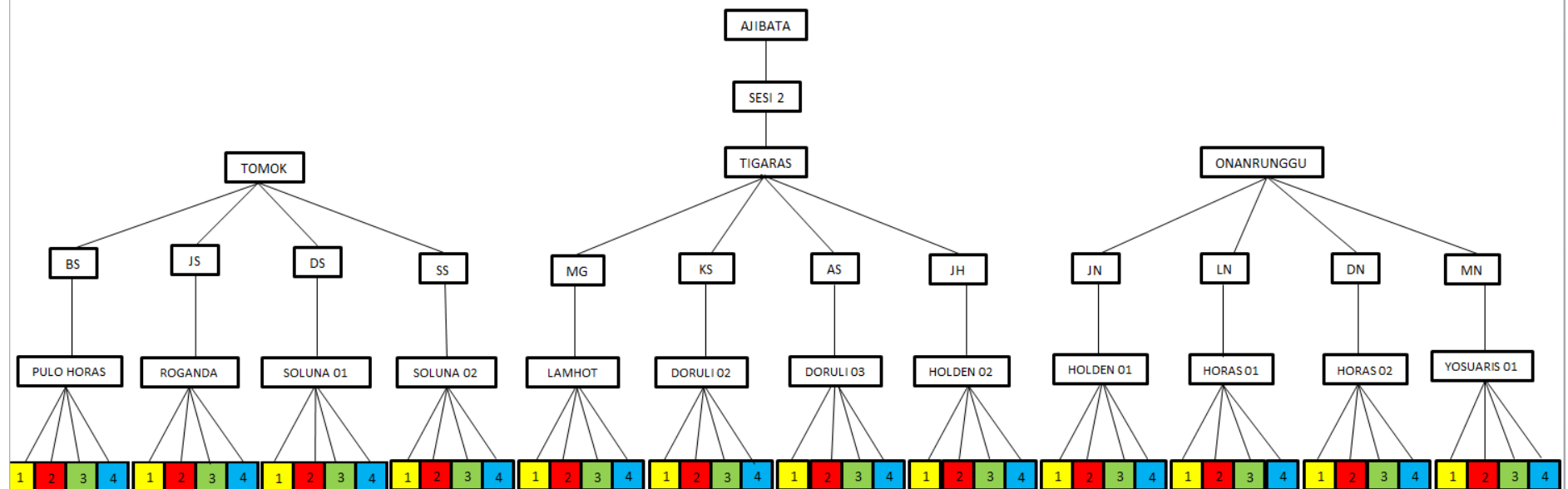
- Hari beroperasi, menggunakan 7 hari yaitu senin, selasa, rabu, kamis, jumat, sabtu, dan minggu.
- Jumlah sesi sehari, yaitu sejumlah 4 sesi.
- Kapal maksimal yang dapat digunakan sehari sejumlah 48 kapal.

Berikut merupakan *tree* yang menggambarkan algoritme *Backtracking* dalam pengalokasian kapal dan waktu:



Gambar 4. Tree Algoritme Backtracking Pertama

Tree Algoritme terdiri dari 6 level, diantaranya *root* merupakan Ajibata yang memiliki satu percabangan yaitu Sesi 1, *node* Sesi 1 memiliki percabangan yaitu, pelabuhan yang dituju, percabangan dari *node* pelabuhan yang dituju yaitu inisial nahkoda. Percabangan selanjutnya yaitu nama kapal yang dioperasikan, dan percabangan terakhir adalah jam keberangkatan, jam keberangkatan perharinya sejumlah empat keberangkatan. Dapat dijelaskan bahwa Sesi 1 memiliki 3 tempat tujuan, yaitu Tomok, Tigaras, dan Onanrunggu. Proses *Backtracing* ini akan berjalan dengan mengikuti *constraints* yang ada. Melalui *tree* ini akan ditelusuri solusi yang memungkinkan, seluruh *node* akan ditelusuri untuk memastikan keseluruhan *node* yang ada. Ketika pencarian tiba di *child* maka akan dilakukan 6 proses *backtrack*.



Gambar 5. Tree Algoritme Backtracing Kedua

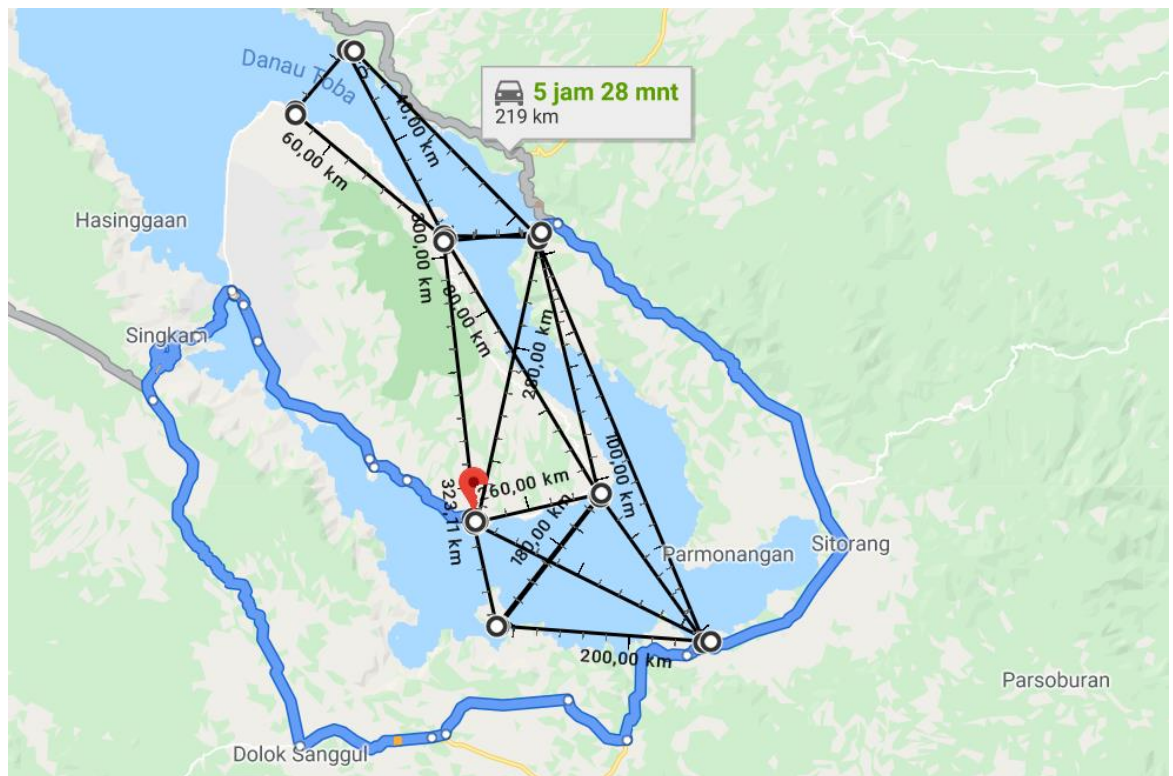
Keterangan:

- Keberangkatan 1
- Keberangkatan 2
- Keberangkatan 3
- Keberangkatan 4

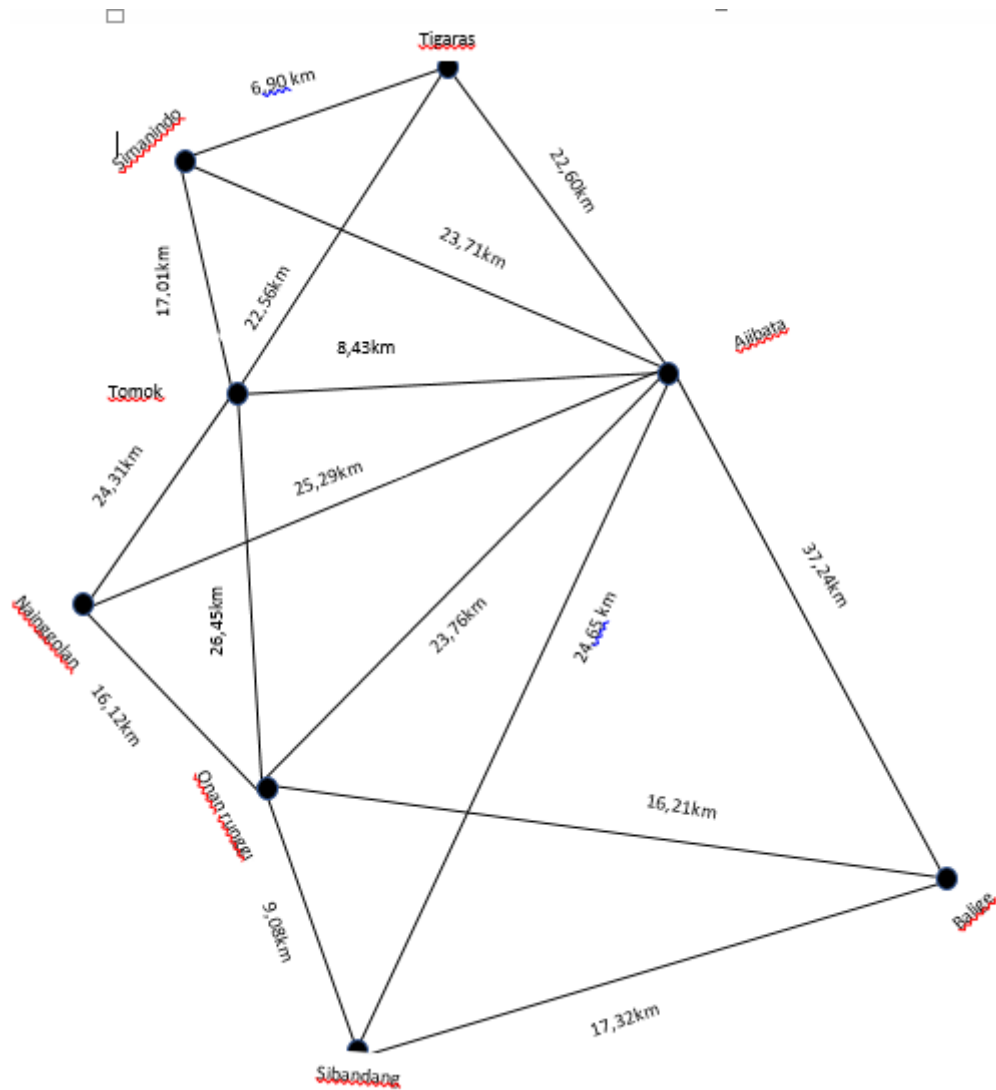
Pada gambar 5 dijelaskan bahwa Sesi 2 memiliki 3 tempat tujuan, yaitu Tomok, Tigaras, dan Onanrunggu. Percabangan selanjutnya adalah inisial nama nahkoda. Berikutnya adalah nama kapal, dan percabangan terakhir adalah jam keberangkatan.

3.4 Perancangan Sistem

Pada bab ini dijelaskan analisis dan pengamatan terhadap pencarian jarak terpendek dengan cara menggunakan algoritme *Steepest Ascent Hill Climbing*. Data yang kami gunakan merupakan data *dummy* yang kami peroleh dari *Google Map*, dikarenakan tidak tersedianya data jarak antar pelabuhan di Dinas Perhubungan Ajibata. Seperti yang ditunjukkan pada Gambar 6 berikut :



Gambar 6. Jarak antar pelabuhan pada Google Map



Gambar 7. Data Jarak Antar Pelabuhan

Dari Gambar 7 diperoleh jarak dari Pelabuhan Ajibata ke Pelabuhan Tomok adalah 8,43 km, Pelabuhan Ajibata ke Pelabuhan Tigaras adalah 22,60 km, dan Pelabuhan Ajibata ke Pelabuhan Onanrunggu adalah 23,76 km.

Berikut beberapa gambar untuk memperjelas jarak Ajibata – Tomok, Ajibata – Tigaras, dan Ajibata – Onanrunggu.

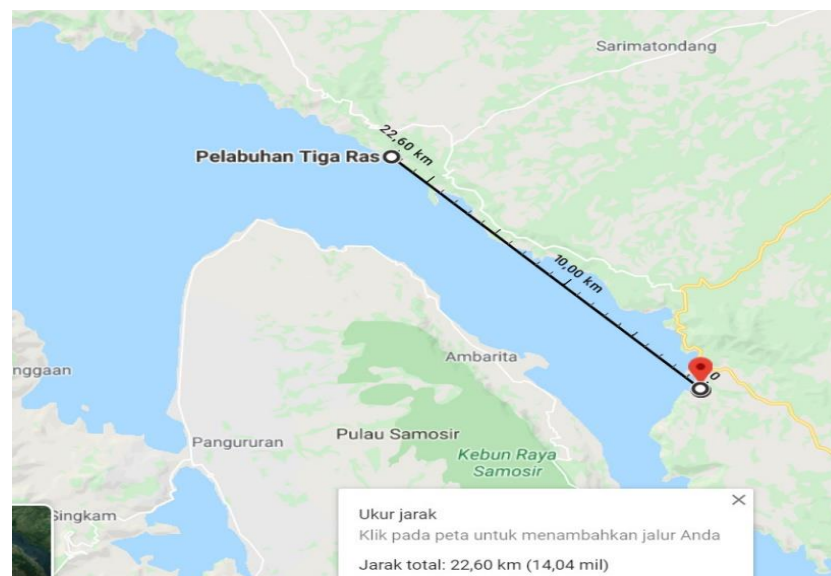
1. Ajibata – Tomok



Gambar 8. Jarak dari Pelabuhan Ajibata ke Tomok

Melalui Gambar 8 dapat kita ketahui bahwa jarak dari Pelabuhan Ajibata menuju Tomok adalah 8,43 km.

2. Ajibata – Tigaras



Gambar 9. Jarak dari Pelabuhan Ajibata ke Tigaras

Melalui Gambar 9 dapat kita ketahui bahwa jarak dari Pelabuhan Ajibata menuju Tigaras adalah 22, 60 km.

3. Ajibata – Onanrunggu



Gambar 10. Jarak dari Pelabuhan Ajibata ke Onanrunggu

Melalui Gambar 10 dapat kita ketahui bahwa jarak dari Pelabuhan Ajibata menuju Onanrunggu adalah 22, 60 km.

3.4.1 Perhitungan jarak menggunakan Algoritme *Steepest Ascent Hill Climbing*

Algoritme *Steepest Ascent Hill Climbing* digunakan untuk mendapatkan suatu jarak yang paling pendek. Pada kasus pencarian jalur terpendek, dibutuhkan algoritme *Steepest Hill Climbing* untuk penentuan pelabuhan yang paling tepat, berdasarkan jarak antar pelabuhan.

Berikut merupakan urutan logika algoritme *Steepest Ascent Hill Climbing* pada pencarian jarak terpendek antar pelabuhan:

1. Evaluasi keadaan awal, jika keadaan awal sama dengan tujuan maka kembali pada keadaan awal dan berhenti berproses. Jika tidak maka keadaan awal tersebut jadikan sebagai keadaan saat ini (*current state*)
2. Mulai dengan *current state* = keadaan awal
3. Dapatkan semua pewaris yang dapat dijadikan keadaan selanjutnya pada *current statenya* dan evaluasi pewaris tersebut dengan fungsi evaluasi dan

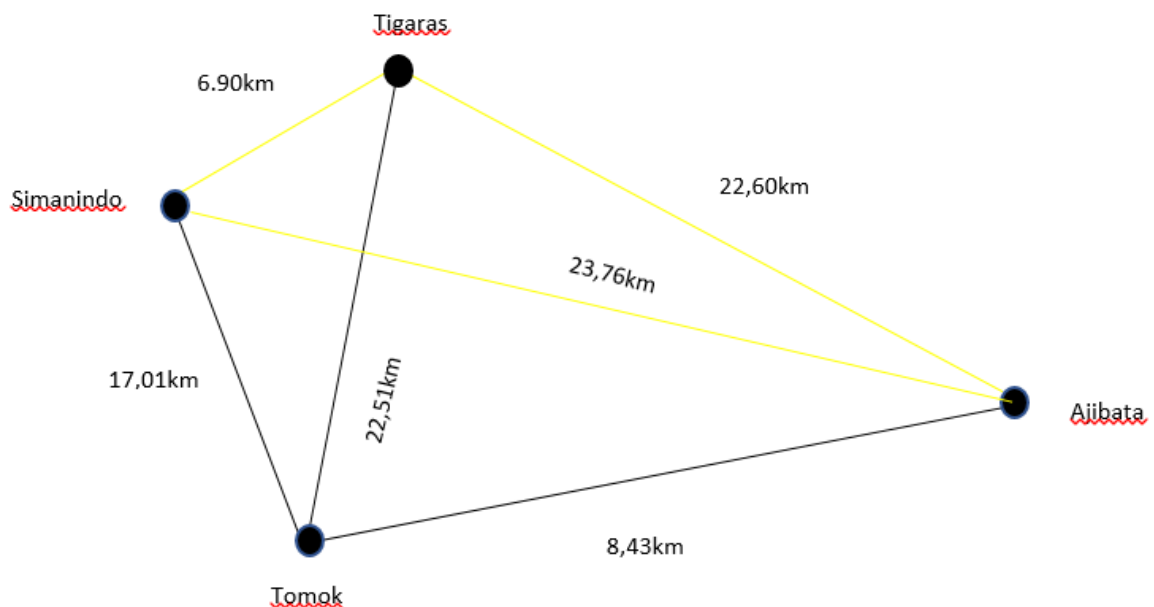
beri nilai pada setiap pewaris tersebut. Jika salah satu dari pewaris tersebut mempunyai nilai yang lebih baik dari *current state* maka jadikan pewaris dengan nilai yang paling baik tersebut sebagai *new current state*. Lakukan operasi ini terus menerus hingga tercapai *current state* = tujuan atau tidak ada perubahan pada *current state* nya.

3.4.2 Graph Pencarian Rute Terpendek

Pembangunan sistem penjadwalan dengan jarak terpendek di Pelabuhan Ajibata menggunakan *Graph*. *Graph* digunakan untuk menggambarkan rute yang akan dilalui oleh kapal. Adapun hasil graph yang telah diperoleh merupakan dari pencarian jarak yang rumusnya kecepatan dikali dengan waktu tempuh ($s = v \times t$), hitungan nya dapat dilihat didalam lampiran. Berikut hasil graph yang telah dihitung,

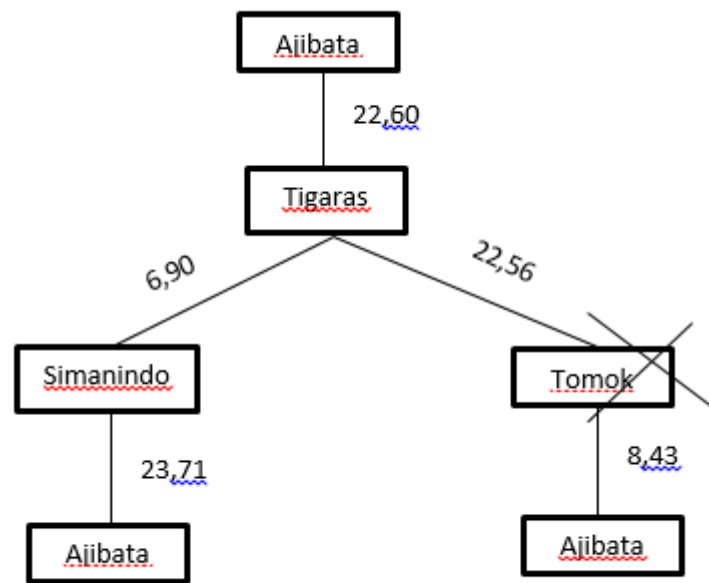
1. Ajibata – Tigaras

Berikut rute perjalanan yang dapat dilalui dari Ajibata ke Tigaras.



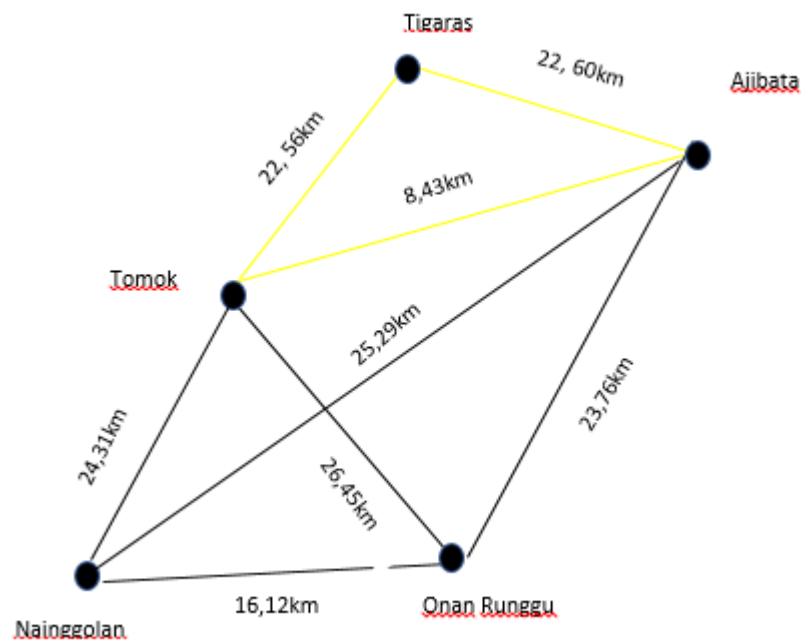
Gambar 11. Rute Perjalanan dari Ajibata ke Tigaras

Maka dengan Algoritme *Steepest Ascent Hill Climbing* didapat jalur terpendek dengan rute Ajibata – Tigaras yaitu Ajibata – Tigaras – Simanindo – Ajibata dengan total jarak = 53,21 km.



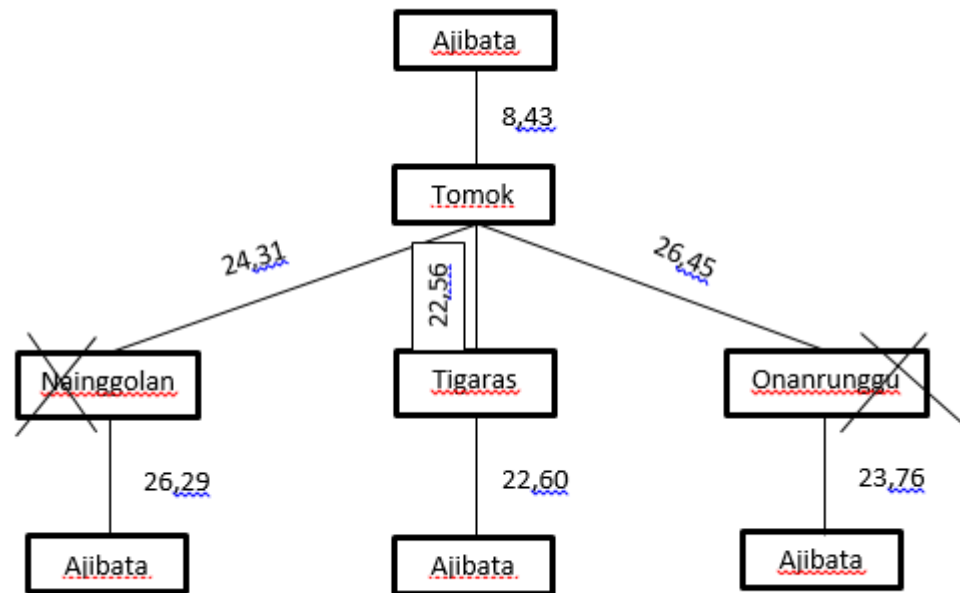
Gambar 12. Rute Terpendek Ajibata Tigaras

2. Ajibata – Tomok



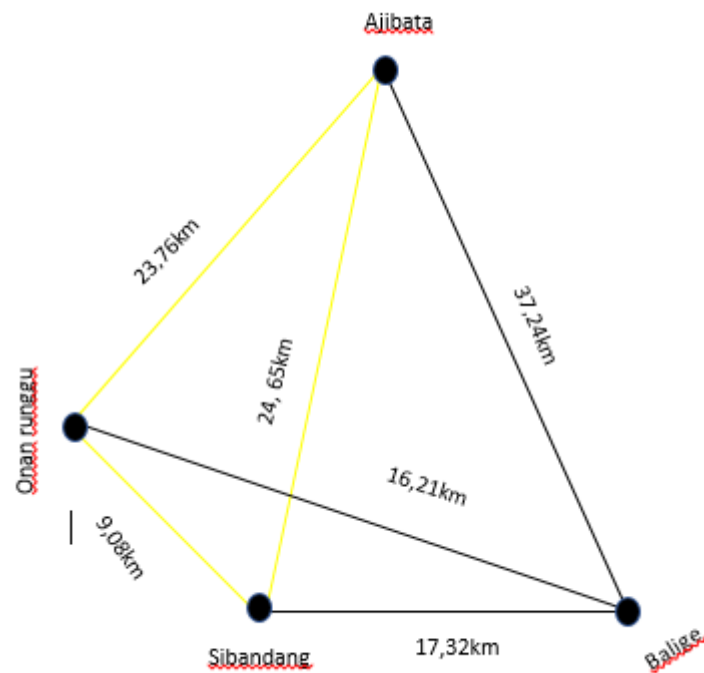
Gambar 13. Rute Perjalanan dari Ajibata ke Tomok

Maka dengan Algoritme *Steepest Ascent Hill Climbing* didapat jalur terpendek dengan rute Ajibata – Tomok yaitu Ajibata – Tomok – Tigaras – Ajibata dengan total jarak = 53,59 km.



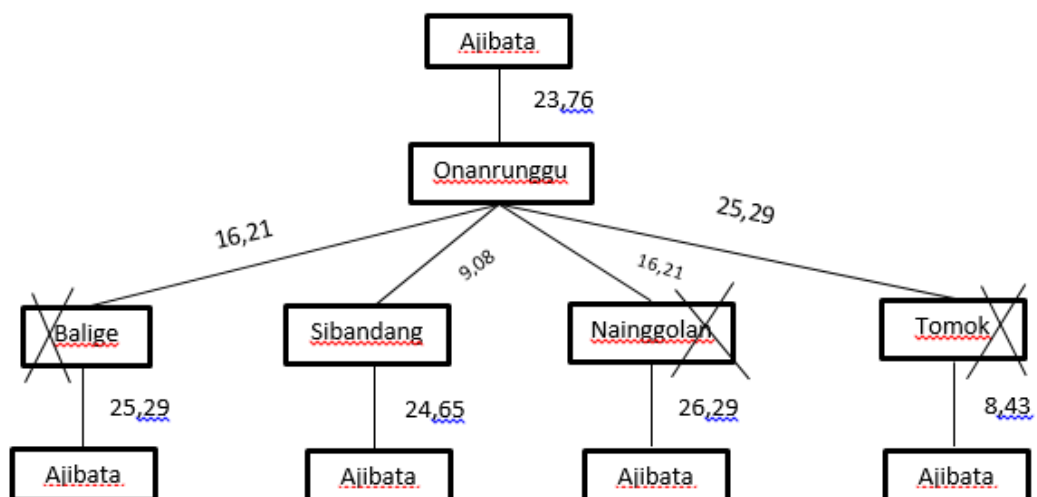
Gambar 14. Rute Terpendek Ajibata-Tomok

3. Ajibata – Onan runggu



Gambar 15. Rute Perjalanan dari Ajibata ke Onanrunggu

Maka dengan Algoritme *Steepest Ascent Hill Climbing* didapat jalur terpendek dengan rute Ajibata – Onanrunggu yaitu Ajibata – Onanrunggu – Sibandang – Ajibata dengan total jarak = 57,49 km



Gambar 16. Rute Terpendek Ajibata-Onanrunggu

3.5 Analisis Kebutuhan Sistem

Pada bab ini akan dijelaskan kebutuhan dari system yang akan dibangun. Sistem yang akan dihasilkan diharapkan dapat menyelesaikan permasalahan pembuatan jadwal penyeberangan kapal dengan jarak terpendek. Kebutuhan sistem yang akan dibangun secara garis besar adalah sebagai berikut:

1. User

User yang akan mengelola sistem yang akan dibangun yaitu Dinas Perhubungan di Ajibata. Pengguna dapat melakukan akses terhadap modul yang ada pada sistem yang akan dibangun serta mengoperasikanny. Hal yang dapat dilakukan oleh pengguna sistem yaitu memberikan masukan berdasarkan batasan yang telah ada sebelumnya adalah sebagai berikut :

- a. Mengelola data
- b. Membuat jadwal
- c. Mengunduh jadwal

2. Input

Input-an yang digunakan dalam pembuatan jadwal kapal penyeberangan ialah berupa data penjadwalan seperti data kapal, nahkoda, wilayah, waktu, dan wilayah.

3. Proses

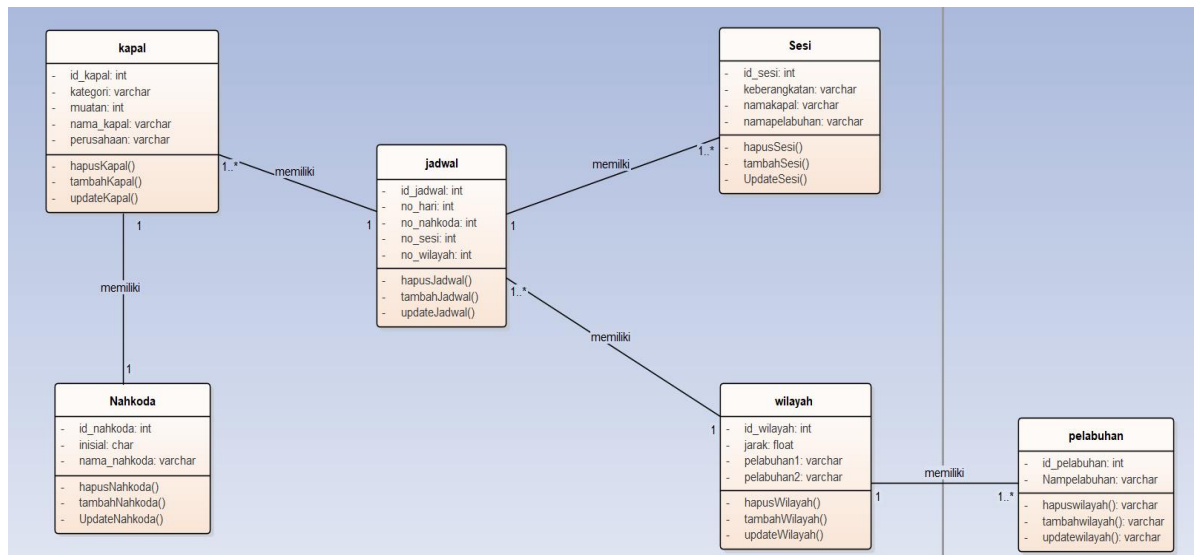
Bagian ini dijelaskan mengenai proses terhadap data yang telah di input pada sistem dalam pembuatan jadwal kapal penyeberangan dengan pendekatan CSP menggunakan algoritme *Backtracking* dengan rute terpendek menggunakan algoritme *Steepest Ascent Hill Climbing*.

4. Output

Output dari sistem yang akan dibangun akan menghasilkan tampilan jadwal kapal penyeberangan pada sistem serta dapat di *export* menjadi tabel *excel*.

3.6. Class Diagram

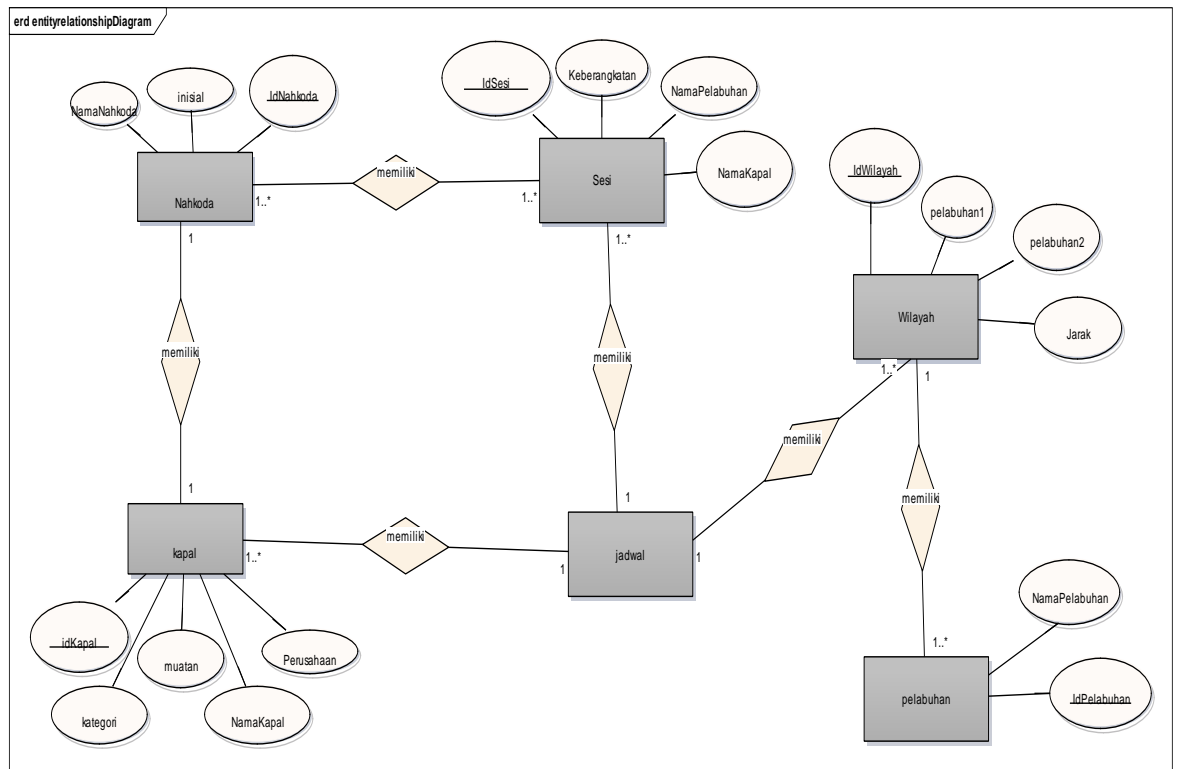
Berikut merupakan gambar *class diagram* dari aplikasi penjadwalan kapal. Pada diagram terdapat enam kelas yaitu Jadwal, Kapal, Nahkoda, sesi, pelabuhan, dan wilayah.



Gambar 17. Class Diagram

3.7 Entity Relationship Diagram (ERD)

Pada bab ini penulis menggunakan *Entity Relationship Diagram* (ERD) dengan notasi Chen yaitu untuk menggambarkan relasi antar *entity* dalam basis data pada sistem penjadwalan mata kapal seperti pada Gambar 18:



Gambar 18. ER-Diagram

3.8 Struktur Database Penjadwalan

1. Tabel Kapal

Digunakan untuk menyimpan data kapal.

Nama Tabel : kapal

Primary Key : IdKapal

Table 8. Tipe Data Kapal

No.	Nama Field	Tipe	Keterangan
1.	IdKapal	Int	Id dari tabel kapal
2.	NamaKapal	varchar	Nama kapal
3.	Perusahaan	varchar	Nama perusahaan pemilik kapal
4.	Muatan	Int	Muatan dari kapal
5.	Kategori	varchar	Kategori Pelabuhan

2. Tabel Nahkoda

Digunakan untuk menyimpan data nahkoda.

Nama Tabel : Nahkoda

Primary Key : IdNahkoda

Table 9. Tipe Data Nahkoda

No.	Nama Field	Tipe	Keterangan
1.	IdNahkoda	Int	Id dari table nahkoda
2.	NamaNahkoda	varchar	Nama dari nahkoda
3.	Inisial	char	Inisial dari si nahkoda

3. Tabel Sesi

Digunakan untuk menyimpan data sesi.

Nama Tabel : sesi

Primary Key : id_sesi

Table 10. Tipe Data Sesi

No.	Nama Field	Tipe	Keterangan
1.	id_sesi	Int	Id dari tabel sesi
2.	keberangkatan	varchar	waktu dari sesi
3.	NamaKapal	varchar	Nama dari sesi
4.	NamaPelabuhan	varchar	Pelabuhan dari sesi

4. Tabel Wilayah

Digunakan untuk menyimpan data wilayah.

Nama Tabel : wilayah

Primary Key : id_wilayah

Table 11. Tipe Data Wilayah

No.	Nama Field	Tipe	Keterangan
1	Id_wilayah	int	Id dari tabel wilayah
2	Pelabuhan1	varchar	Nama wilayah
3	Pelabuhan2	varchar	Nama wilayah
4	Jarak	Int	Jarak dari antar wilayah

5. Tabel pelabuhan

Digunakan untuk menyimpan data pelabuhan

Nama tabel: pelabuhan

Primary Key : id_pelabuhan

Table 12. Tipe Data pelabuhan

No.	Nama Field	Tipe	Keterangan
1.	IdPelabuhan	Int	Id dari tabel pelabuhan
2.	NamaPelabuhan	Varchar	Nama pelabuhan

3.10 Antarmuka Perangkat Lunak

Pada bagian ini penulis Antarmuka Perangkat Lunak (*software interface*) yang dibutuhkan dalam mendukung pembangunan sistem yaitu menggunakan *Xampp* karena mendukung banyak sistem operasi yang merupakan kompilasi dari beberapa program yang digunakan sebagai penerjemahan bahasa yang ditulis dengan bahasa pemrograman Java.

3.10.1 Antarmuka Pengguna

Pada bagian ini penulis menjelaskan kebutuhan atau gambaran pelayanan yang harus dipenuhi pada sistem yang akan dibangun. Kebutuhan antarmuka pengguna (*user interface*) untuk mengoperasikan sistem yang akan dibangun terdapat pada Tabel 13:

Table 13. Kebutuhan User Interface

No.	Nama	Keterangan
1	<i>Monitor</i>	<i>Monitor</i> digunakan untuk menampilkan <i>output</i> dari proses yang dilakukan

No.	Nama	Keterangan
2	<i>Keyboard</i>	Antarmuka yang digunakan untuk memasukkan data ke dalam sistem.
3	<i>Monitor</i>	Antarmuka yang memungkinkan pengguna untuk melihat perubahan yang terjadi pada sistem serta mempermudah pengguna untuk berinteraksi dengan sistem yang akan dibangun.

3.10.2 Lingkungan Pengembangan

Pada bagian ini penulis menjelaskan *development environment* atau lingkungan pengembangan berisi spesifikasi teknis dari perangkat lunak yang diperlukan dalam pembangunan sistem. Daftar perangkat yang akan digunakan oleh *developer* dalam pembangunan adalah sebagai berikut:

1. *Client*
 - a. *Operating System*: Windows 10 Pro
 - b. *DBMS*: MySQL
 - c. *Tools*: XAMPP, Google Chrome, Mozilla Firefox, SQLYog
2. *Hardware*

Beberapa spesifikasi *hardware* yang digunakan antara lain:

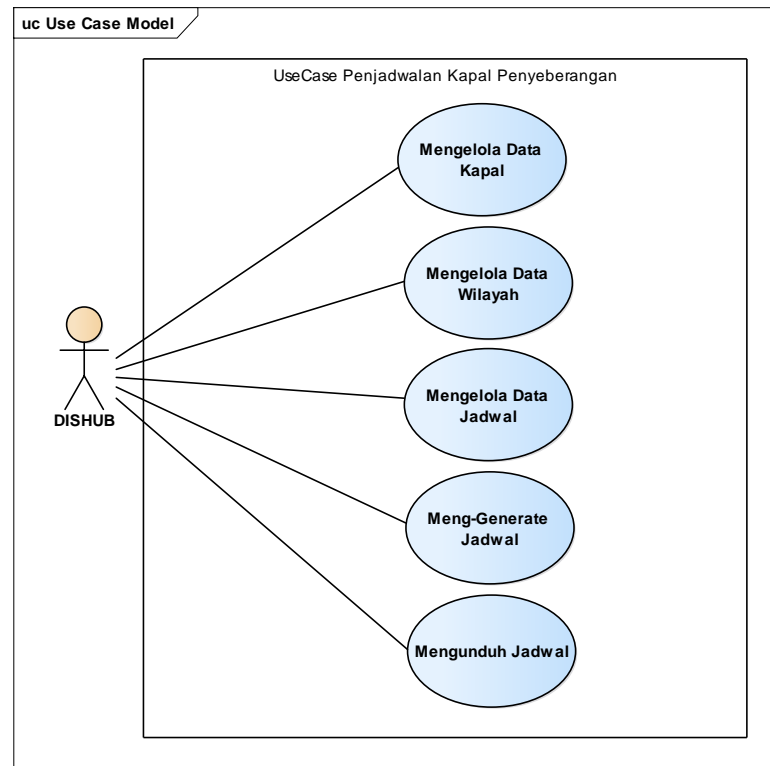
 - a. *Processor*: Intel Core I5
 - b. *Hard disk*: 500GB
 - c. *Memory*: 8GB

3.11 Deskripsi Fungsional

Deskripsi fungsional dari sistem penjadwalan yang akan dibangun dapat digambarkan dalam bentuk *use case diagram* dan *use case scenario*.

3.11.1 Use Case Diagram Penjadwalan Kapal Penyeberangan

Pada bagian ini dijelaskan fungsi-fungsi dari sistem penjadwalan yang digambarkan dalam bentuk *use case diagram* seperti yang ditunjukkan pada Gambar 19:



Gambar 19. Use Case Diagram

Penjelasan fungsi-fungsi pada *use case diagram* penjadwalan kapal penyeberangan pada Gambar 19 adalah sebagai berikut:

1. Mengelola Kapal

DISHUB sebagai *user* pada sistem penjadwalan kapal penyeberangan yang akan dibangun dapat melakukan hal-hal sebagai berikut:

- a. Menambahkan Data, DISHUB dapat menambahkan data seperti nama kapal, perusahaan, muatan, nama nahkoda, inisial nahkoda, kategori pelabuhan tujuan.
- b. Mengedit Data, DISHUB dapat mengedit data nama kapal, perusahaan, muatan, nama nahkoda, inisial nahkoda, dan kategori pelabuhan tujuan.
- c. Menghapus Data, DISHUB dapat menghapus data seperti nama kapal, perusahaan, muatan, nama nahkoda, inisial nahkoda, dan kategori pelabuhan tujuan.
- d. Melihat Data, DISHUB dapat menampilkan data seperti nama kapal, perusahaan, muatan, nama nahkoda, inisial nahkoda, dan kategori pelabuhan tujuan.

2. Mengelola wilayah

DISHUB sebagai *user* pada sistem penjadwalan kapal penyeberangan yang akan dibangun dapat melakukan hal-hal sebagai berikut:

- a. Menambahkan data, DISHUB dapat menambahkan data seperti jarak, pelabuhan awal dan pelabuhan tujuan.
- b. Meng-*update data*, DISHUB dapat mengu-*update data* yang diperlukan.
- c. Menghapus data, DISHUB dapat menghapus data yang diperlukan
- d. Melihat Data, DISHUB dapat menampilkan data seperti nama pelabuhan, jarak dan katehgori pelabuhan tujuan

3. Mengelola Jadwal

DISHUB sebagai *user* pada sistem penjadwalan kapal penyeberangan yang akan dibangun dapat melakukan hal-hal sebagai berikut:

- a. Menambahkan data, DISHUB dapat menambahkan data sesi keberapa, keberangkatan keberapa, dan nama nahkoda.
- b. Menghapus data, DISHUB dapat menghapus data yang diperlukan
- c. Melihat Data, DISHUB dapat menampilkan data seperti nama pelabuhan, jarak dan katehgori pelabuhan tujuan

4. Meng-*genarate* Jadwal

DISHUB sebagai *user* pada sistem penjadwalan kapal penyeberangan yang akan dibangun dapat melakukan *generate* jadwal. *Generate* jadwal terdiri dari dua tahap, pertama yaitu menjalankan algoritme *Backtracking* dengan metode CSP. Pada bagian ini, digunakan algoritme *Backtracking* untuk menghasilkan jadwal yang tidak bentrok. Tahap kedua dilakukan *generate Steepest Ascent Hill Climbing* untuk memperoleh wilayah berdasarkan jarak terdekat sesuai jadwal per harinya.

5. Mengunduh Jadwal

DISHUB sebagai *user* pada sistem penjadwalan kapal penyeberangan yang akan dibangun dapat melakukan unduh atau *download* jadwal yang sudah di-

generate. Mengunduh jadwal yang telah di-*generate* yaitu dengan cara melakukan *convert* data ke *excel*.

3.11.2 Use Case Sceneario

Pada bagian ini dijelaskan prosedur atau langkah-langkah *user* petugas pelabuhan Ajibata menggunakan sistem penjadwalan yang digambarkan dalam bentuk *use case scenario* seperti berikut:

3.11.2.1 Use Case Scenario Mengelola Data

Use case scenario untuk mengelola data dapat dilihat pada table - tabel *use case scenario* berikut:

1. Use Case Scenario Mengelola Data Kapal

Detail dari *usecase scenario* mengelola data kapal dapat dilihat pada Table 14 dibawah ini.

Table 14. Use Case Scenario Mengelola Data Kapal

<i>Use Cace ID</i>	UCS_001	
<i>Use Case Name</i>	Mengelola data kapal	
<i>Brief Description</i>	<i>Use case</i> ini menjelaskan <i>user</i> dapat menambah, mengupdate dan menghapus data kapal, dengan cara memasukkan data nama kapal. Perusahaan, muatan, nama nahkoda, kategori pelabuhan	
<i>Primary Actor</i>	DISHUB Ajibata	
<i>Secondary Actor</i>	-	
<i>Pre-condition</i>	DISHUB sudah mengakses sistem	
<i>Post-condition</i>	DISHUB berhasil melakukan mengelola (menambah, mengupdate, atau menghapus) data kapal	
<i>Primary Flow of Events</i>	<i>User Action</i>	<i>System Response</i>
	1. DISHUB memilih <i>button</i> Kelola Kapal	
		2. Aplikasi menampilkan halaman Kelola Kapal
	3. DISHUB mengisi <i>field</i> yang tersedia	
	4. DISHUB memilih tombol Tambah	
		5. Sistem menyimpan dan menampilkan data pada tabel yang tersedia

<i>Alternate Flow of Events</i>	-	-
<i>Error Flow of Events</i>	<i>User Action</i>	<i>System Response</i>
	3A. form tidak diisi dengan benar	
		5B. Instruksi tidak dilanjutkan
<i>Post Condition</i>	Aktor berhasil mengelola data kapal pada sistem	

2. Use Case Scenario Mengelola Data Wilayah

Detail dari *usecase scenario* mengelola data wilayah dapat dilihat pada Table 15 dibawah ini.

Table 15. Use Case Scenario Mengelola Data wilayah

<i>Use Case ID</i>	UCS_002	
<i>Use Case Name</i>	Mengelola data wilayah	
<i>Brief Description</i>	<i>Use case</i> ini menjelaskan <i>user</i> dapat menambah, mengupdate dan menghapus data wilayah, dengan cara memasukkan jarak antar pelabuhan.	
<i>Primary Actor</i>	DISHUB Ajibata	
<i>Secondary Actor</i>	-	
<i>Pre-condition</i>	DISHUB sudah mengakses sistem	
<i>Post-condition</i>	DISHUB berhasil melakukan mengelola (menambah, mengupdate, atau menghapus) data wilayah	
<i>Primary Flow of Events</i>	<i>User Action</i>	<i>Primary Flow of Events</i>
	1. DISHUB memilih <i>button</i> Kelola Wilayah	
		2. Aplikasi menampilkan halaman Kelola Wilayah
	3. DISHUB mengisi <i>field</i> yang tersedia	
	4. DISHUB memilih tombol Tambah	
		5. Sistem menyimpan dan menampilkan data pada tabel yang tersedia
<i>Alternate Flow of Events</i>	-	-
<i>Error Flow of Events</i>	<i>User Action</i>	<i>System Response</i>
	3A. form tidak diisi dengan benar	

		5B. Instruksi tidak dilanjutkan
<i>Post Condition</i>	Aktor berhasil mengelola data wilayah pada sistem	

3. Use Case Scenario Mengelola jadwal

Detail dari *usecase scenario* mengelola jadwal dapat dilihat pada Table 16 dibawah ini.

Table 16. Use Case Scenario Mengelola Data Jadwal

<i>Use Case ID</i>	UCS_003	
<i>Use Case Name</i>	Membuat jadwal	
<i>Brief Description</i>	<i>Use case</i> ini menjelaskan <i>user</i> dapat membuat jadwal dengan cara mengisi data sesi, keberangkatan dan nama kapal, lalu dapat di tambah.	
<i>Primary Actor</i>	DISHUB Ajibata	
<i>Secondary Actor</i>	-	
<i>Pre-condition</i>	DISHUB sudah mengakses sistem	
<i>Post-condition</i>	DISHUB berhasil <i>men-generate</i> membuat jadwal	
<i>Primary Flow of Events</i>	<i>User Action</i>	<i>System Response</i>
	1. DISHUB memilih <i>button</i> kelola jadwal	
		2. Aplikasi menampilkan halaman kelola Jadwal
	3. DISHUB mengisi <i>field</i> yang tersedia	
	4. DISHUB memilih tombol Tambah	
		5. Sistem menyimpan dan menampilkan data pada tabel yang tersedia
<i>Alternate Flow of Events</i>	-	-
<i>Error Flow of Events</i>	<i>User Action</i>	<i>System Response</i>
	3A. form tidak diisi dengan benar	
		5B. Instruksi tidak dilanjutkan
<i>Post Condition</i>	Aktor berhasil mengelola data jadwal pada sistem	

4. Use Case Scenario Generate Jadwal

Detail dari *usecase scenario* mengelola data sesi dapat dilihat pada Table 17 dibawah ini.

Table 17. Use Case Scenario Mengelola Generate Jadwal

<i>Use Case ID</i>	UCS_004	
<i>Use Case Name</i>	Meng-generate jadwal	
<i>Brief Description</i>	<i>Use case</i> ini menjelaskan <i>user</i> dapat meng-generate jadwal dengan cara memasukkan maksimal keberangkatan dalam 1 sesi	
<i>Primary Actor</i>	DISHUB Ajibata	
<i>Secondary Actor</i>	-	
<i>Pre-condition</i>	DISHUB sudah mengakses sistem	
<i>Post-condition</i>	DISHUB berhasil melakukan generate jadwal	
<i>Primary Flow of Events</i>	<i>User Action</i>	<i>System Response</i>
	1. DISHUB mengisi jumlah maksimal keberangkatan kapal lalu di generate	
		2. Sistem menampilkan tabel berisi data hari, sesi, keberangkatan, kapal, nahkoda, pelabuhan, dan rute penyeberangan
	3. DISHUB memilih tombol simpan	
		4. Sistem menyimpan dan menampilkan data pada tabel yang tersedia
<i>Alternate Flow of Events</i>	-	-
<i>Error Flow of Events</i>	<i>User Action</i>	<i>System Response</i>
	1A. form tidak diisi dengan benar	
		4B. Instruksi tidak dilanjutkan
<i>Post Condition</i>	Aktor berhasil membuat atau generate jadwal berdasarkan data yang sudah ditambahkan ke sistem	

3.11.2.2 Use Case Scenario Mengunduh Jadwal

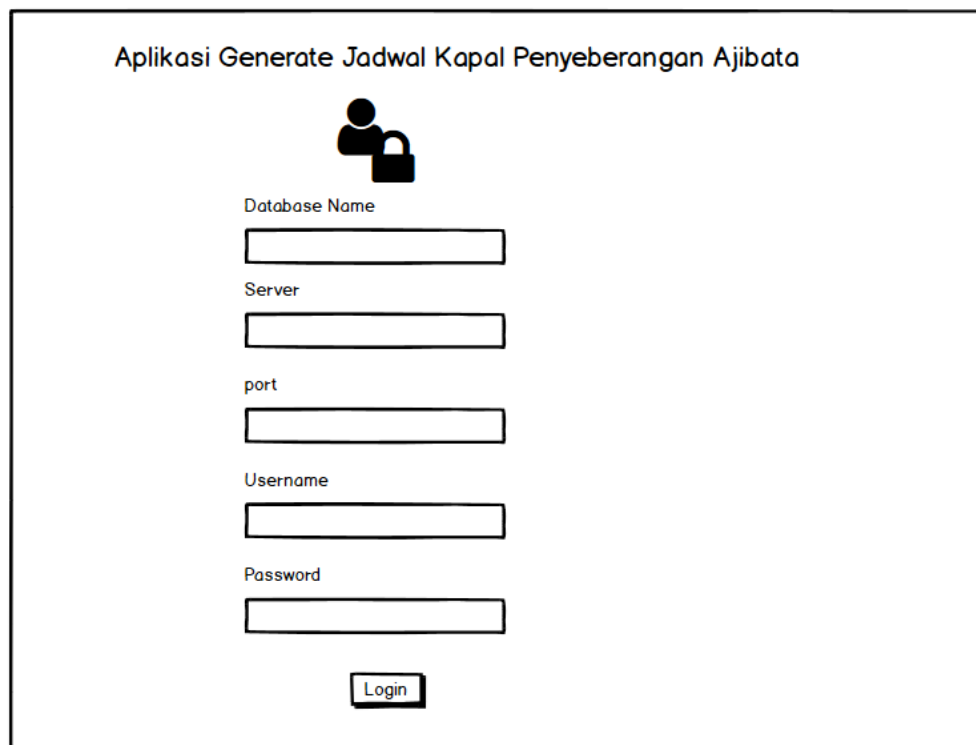
Use case scenario untuk membuat atau mengunduh jadwal data dapat dilihat pada Tabel 18 berikut:

Table 18. Use Case Scenario Membuat Jadwal


<i>Use Case ID</i>	UCS_005	
<i>Use Case Name</i>	Mengunduh jadwal	
<i>Brief Description</i>	<i>Use case</i> ini menjelaskan <i>user</i> dapat mengunduh hasil <i>generate</i> jadwal pada sistem	
<i>Primary Actor</i>	DISHUB Ajibata	
<i>Secondary Actor</i>	-	
<i>Pre-condition</i>	DISHUB sudah mengakses sistem	
<i>Post-condition</i>	DISHUB berhasil mengunduh jadwal	
<i>Primary Flow of Events</i>	<i>User Action</i>	<i>System Response</i>
	1. DISHUB memilih <i>button</i> Generate Jadwal	
		2. Aplikasi menampilkan halaman Generate Jadwal
	3. DISHUB memilih tombol Lihat Jadwal	
		4. Sistem menampilkan tabel berisi data jadwal yang sudah ditambahkan.
	5. DISHUB memilih tombol <i>Export ke Excel</i>	
		6. Sistem menampilkan data jadwal dalam bentuk <i>excel</i>
<i>Alternate Flow of Events</i>	-	-
<i>Error Flow of Events</i>	-	-
<i>Post Condition</i>	Aktor berhasil mengunduh hasil <i>generate</i> jadwal berdasarkan data yang sudah ditambahkan ke sistem	

3.12 Desain

Pada bab ini penulis menjelaskan desain aplikasi yang akan dibangun pada sistem penjadwalan saat melakukan *generate* jadwal.



Aplikasi Generate Jadwal Kapal Penyeberangan Ajibata



Database Name

Server

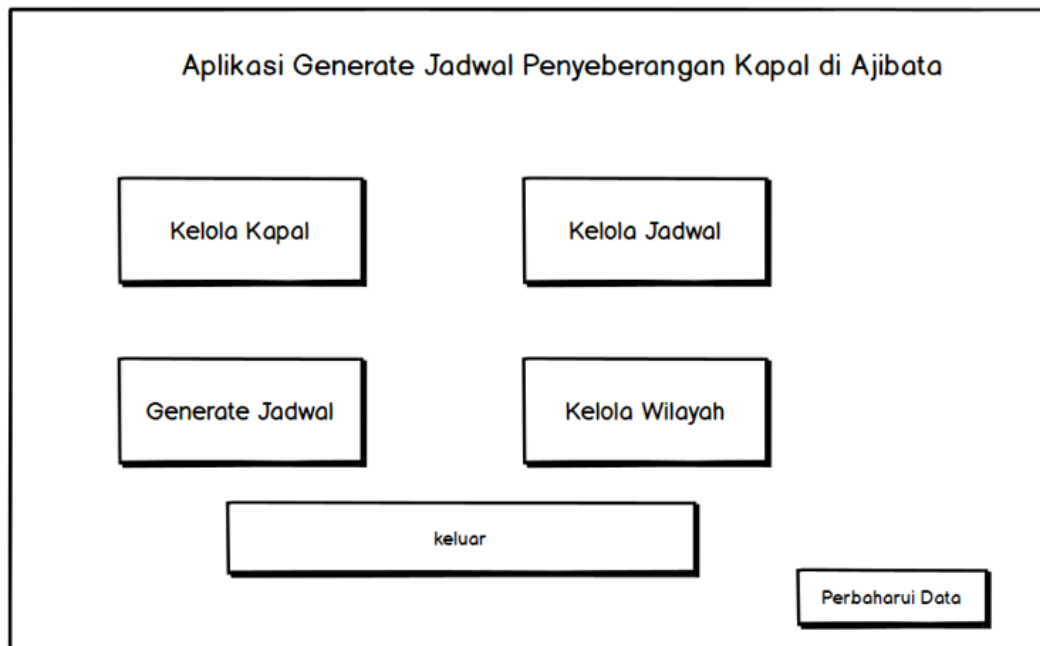
port

Username

Password

Gambar 20. Mockup Login

Pada Gambar 20 akan terlihat tampilan *Login*, ini diperuntukkan kepada Dinas Perhubungan Ajibata. Dimana dinas perhubungan dapat menginput *database name*, *server*, *port*, *username* yang telah diketahui sebelumnya tetapi *password* dapat dikosongkan karena hanya admin dinas perhubungan yang dapat *login* kedalam aplikasi.



Gambar 21. Mockup Home

Pada Gambar 21 akan terlihat tampilan *Home*. Dimana terdapat *button-button* untuk kelola kapal, kelola jadwal, generate jadwal, kelola wilayah–, dan keluar.

The mockup shows a 'Kelola Kapal' screen. At the top right is a 'kembali' button. The title is 'Aplikasi Jadwal Penyeberangan Kapal di Ajibata'. Below the title, there is a section for 'Data Kapal' with input fields for 'perusahaan', 'muatan', 'Nama Nahkoda', and 'Inisial Nahkoda'. There is also a 'Kategori' dropdown menu. Below these fields are three buttons: 'Tambah', 'Update', and 'Hapus'. To the right of the input fields is a table with the following columns: 'Nama Kapal', 'perusahaan', 'mtn', 'ktgr', 'N.Nahkoda', and 'I.Nahkoda'. Above the table is a note: '** Pilih salah satu data pada tabel untuk update'. The table has 10 empty rows for data entry.

Nama Kapal	perusahaan	mtn	ktgr	N.Nahkoda	I.Nahkoda

Gambar 22. Mockup Kelola Kapal

Pada Gambar 22 akan terlihat tampilan *form* kelola kapal. Dimana data yang harus diisi meliputi, nama kapal, nama perusahaan, nama nahkoda, inisial nahkoda, dan dapat memilih kategori pelabuhan. *Form* untuk kelola kapal juga dapat menambah, meng-*update*, dan menghapus data kapal.

Aplikasi Jadwal Penyeberangan Kapal di Ajibata kembali

Data Jadwal

Sesi

Keberangkatan

Kapal

Sesi	Keberangkatan	Kapal

Gambar 23. Mockup Kelola Jadwal

Pada Gambar 23 akan terlihat tampilan *form* kelola jadwal. Dimana data yang harus diisi meliputi, data sesi, keberangkatan, nama kapal. *Form* kelola jadwal juga dapat menambahkan data jadwal serta menghapus relasi yang tidak dibutuhkan.

Aplikasi Jadwal Penyeberangan Kapal di Ajibata

[kembali](#)

Data Wilayah

Nama	Jarak	Kategori

Gambar 24. Mockup Kelola Wilayah

Pada Gambar 24 akan terlihat tampilan *form* kelola wilayah. Dimana data yang harus diisi meliputi, nama wilayah, pelabuhan awal, pelabuhan yang ditempuh. *Form* untuk kelola wilayah juga dapat menambah, meng-*update*, dan menghapus data kapal.

Aplikasi Jadwal Penyeberangan Kapal di Ajibata

[kembali](#)

Jumlah Maksimal Keberangkatan dalam 1 sesi

Sesi	Keberangkatan	Kapal	Nahkoda	Pelabuhan	Rute

[Simpan ke Excel](#)

Gambar 25. Mockup Generate Jadwal

Pada Gambar 25 akan terlihat tampilan *form* Generate Jadwal. Dimana data yang harus diisi meliputi, jumlah sesi lalu di generate. Hasil generate nya dapat disimpan ke dalam bentuk excel.

BAB IV

IMPLEMENTASI dan TESTING

Pada bab ini akan dijelaskan tentang kebutuhan implementasi, pengujian, dan hasil pengujian.

4.1 Kebutuhan Implementasi

Pada subbab ini akan dijelaskan mengenai apa-apa saja yang menjadi kebutuhan perangkat keras dan perangkat lunak dalam proses pembangunan sistem penjadwalan kapal penyeberangan. Berikut merupakan spesifikasi perangkat keras dan perangkat lunak.

4.1.1 Spesifikasi Perangkat Keras

Table 19. Spesifikasi Perangkat Keras

No.	Hardware	Spesifikasi	
1	Lenovo	Processor	Intel Core i5
		Memory	8 GB
		Operating System	Windows 10

4.1.2 Spesifikasi Perangkat Lunak

Table 20. Spesifikasi Perangkat Lunak

No.	Software	Spesifikasi
1	Development Tools	Netbeans IDE
2	Programming Language	Java
3	Database	MY SQL
4	Web Server	XAMPP
5	Operating System	Windows 10

4.2 Hasil Aplikasi

Pada subbab ini akan dijelaskan mengenai sistem penjadwalan kapal penyeberangan yang telah dibangun. Sistem penjadwalan kapal penyeberangan memiliki beberapa halaman yang digunakan, diantaranya menu utama, kelola kapal, kelola jadwal, kelola wilayah, *generate* jadwal, perbaharui data dan keluar.

4.2.1 Halaman Login

Saat sistem penjadwalan tersebut dibuka, maka akan muncul halaman Login. Yang dapat Login ke sistem adalah Dinas Perhubungan Ajibata. Dimana akan memasukkan *database*, *name*, *server*, *port*, dan *username*.



Aplikasi Generate Jadwal Kapal Penyeberangan Ajibata

Database Name

Server

Port

Username

Password

Login

Gambar 26. Tampilan Menu Login

4.2.2 Menu Utama

Aplikasi *generate* jadwal kapal penyeberangan memiliki menu utama, dimana memperlihatkan adanya 5 menu, yaitu kelola kapal, kelola jadwal, generate jadwal, kelola wilayah, dan keluar. Berikut merupakan tampilan utama aplikasi *generate* jadwal penyeberangan kapal di Ajibata,



Gambar 27. Tampilan Menu Utama

4.2.3 Kelola Kapal

Menu kelola kapal dapat digunakan untuk mengelola data-data kapal yang dimiliki setiap pelabuhan. Adapun *attribute* dari data kapal yaitu nama kapal, nama perusahaan, muatan kapal, nama nahkoda, dan inisial nahkoda. *Button* kategori pada halaman kelola data kapal berfungsi untuk memilih kategori pelabuhan, *button* tambah pada halaman kelola data kapal berfungsi untuk menambahkan data yang sudah kita masukkan, *button update* berfungsi untuk memperbaharui data yang sudah ditambahkan sebelumnya. *Button* hapus berfungsi untuk menghapus data yang tidak dibutuhkan. *Button* “kembali” berfungsi untuk kembali ke halaman sebelumnya. Berikut merupakan tampilan halamal mengelola kapal

Nama Kapal	Perusahaan	Muatan	Kategori	Nama
DOS ROHA I	OPS.Tomok Tour	50	Tomok	Wak Tejo
Holden 01	PT.Aquafarm	50	Onanrunggu	Paijo
Holden 03	PT.Aquafarm	40	Onanrunggu	Sudirdjo
Holden 02	PT.Aquafarm	50	Onanrunggu	Painem
DOS ROHA II	OPS.Tomok Tour	50	Tomok	Wak Miner
Sarina	PT.Aquafarm	50	Tigaras	JOKO
Natjo 1	PT.Aquafarm	50	Tigaras	Sariyem
Adinda	PT.Aquafarm	50	Tigaras	Budi
Horas 01	PT.ASDP	50	Onanrunggu	Mukhlis
Horas 02	PT.ASDP	50	Onanrunggu	Muhsin
Soluna 01	PT.ASDP	50	Tomok	Supali
Glanter	PT.Aquafarm	50	Tigaras	Abidin
Soluna 02	PT.ASDP	50	Tomok	Suminem

Gambar 28. Tampilan Menu Kelola Kapal

4.2.4 Kelola Jadwal

Menu kelola jadwal dapat digunakan untuk mengelola data jadwal yang ada di setiap pelabuhan. Adapun *attribute* dari data jadwal yaitu sesi dan keberangkatan. *Button* nahkoda pada halaman kelola jadwal berfungsi untuk memilih nahkoda kapal. *Button* tambah berfungsi untuk menambahkan data yang baru. *Button* hapus semua relasi berfungsi untuk menghapus data yang tidak dibutuhkan.

Aplikasi Generate Jadwal Penyeberangan Kapal di Ajibata

[Kembali](#)

Data Jadwal

Sesi:

Keberangkatan:

Nahkoda:

[Tambah](#)

[Hapus Semua Relasi](#)

Sesi	Keberangkatan	Kapal
1	1	DOS ROHA I
1	1	Holden 01
2	1	Holden 03
3	1	Holden 02
2	1	DOS ROHA II
1	1	Sarina
2	1	Natjo 1
3	1	Adinda
4	1	Horas 01
5	1	Horas 02
3	1	Soluna 01
4	1	Glanter

Total Data

[Simpan ke Database](#) [GENERATE JADWAL](#)

Gambar 29. Tampilan Menu Kelola Jadwal

4.2.5 Generate Jadwal

Menu *generate* jadwal dapat digunakan untuk membuat jadwal pada penyeberangan kapal. Dimana data sesi, keberangkatan, data kapal, data nahkoda, data pelabuhan, dan rute yang ditambahkan akan muncul pada table di halaman *generate* jadwal. Yang mana data-data tersebut dapat disimpan dalam bentuk excel.

Sesi	Keberangkatan	Kapal	Nahkoda	Pelabuhan	
1	1	DOS ROHA I	Wak Tejo	Tomok	Ajibata->Tomok
1	1	Holden 01	Paijo	Onanrunggu	Ajibata->Onanr
2	1	Holden 03	Sudirdjo	Onanrunggu	Ajibata->Onanr
3	1	Holden 02	Painem	Onanrunggu	Ajibata->Onanr
2	1	DOS ROHA II	Wak Minem	Tomok	Ajibata->Tomok
1	1	Sarina	JOKO	Tigaras	Ajibata->Tigaras
2	1	Natjo 1	Sariyem	Tigaras	Ajibata->Tigaras
3	1	Adinda	Budi	Tigaras	Ajibata->Tigaras
4	1	Horas 01	Mukhlis	Onanrunggu	Ajibata->Onanr
5	1	Horas 02	Muhsin	Onanrunggu	Ajibata->Onanr

Gambar 30. Tampilan Menu Generate Jadwal

Proses generate jadwal diatas dilakukan dengan menggunakan metode *Backtracking CSP* dalam proses pencarian solusi. Setelah proses generate jadwal tersebut berhasil, maka proses selanjutnya yaitu melakukan generate jadwal dengan menggunakan metode *Steepest Hill Climbing*.

4.2.6 Kelola Wilayah

Menu kelola wilayah dapat digunakan untuk mengelola data wilayah pelabuhan yang dilewati setiap kapal pada saat melakukan penyeberangan. Menggunakan algoritme *Steepest Ascent Hill Climbing* yang digunakan untuk menentukan jarak terpendek dari antar pelabuhan. Hal ini bertujuan untuk meminimalisasi penggunaan waktu pada saat menyeberang menggunakan kapal.

Aplikasi Generate Jadwal Penyeberangan Kapal di Ajibata

[Kembali](#)

Data Wilayah

Pelabuhan

Pelabuhan

** Pilih salah satu data pada tabel untuk update

Nama	Jarak	Kategori
Ajibata	23.71	Simanindo
Ajibata	8.43	Tomok
Ajibata	25.29	Nainggolan
Ajibata	23.76	Onanrunggu
Ajibata	24.65	Sibandang
Ajibata	37.24	Balige
Simanindo	6.9	Tigaras
Tomok	22.56	Tigaras
Simanindo	17.01	Tomok
Nainggolan	25.29	Tomok
Onanrunggu	26.45	Tomok
Onanrunggu	16.12	Nainggolan
Onanrunggu	9.08	Sibandang
Onanrunggu	16.21	Balige
Sibandang	17.22	Balige

Gambar 31. Tampilan Menu Kelola Wilayah

4.2.7 Kode Program Pemodelan *Backtracking Constraint Satisfaction Problem (CSP)*

Pada bab ini akan menjelaskan kode pemograman *backtracking* yang berisi deklarasi dari setiap variable. Variable menyimpan nilai atau domain dalam bentuk *array*. Variable yang dimaksud yaitu, Nama kapal, sesi, dan keberangkatan. Berikut merupakan gambar dari kode program dari setiap variable penjadwalan penyeberangan kapal:

```

public void generateJadwal(Collection<Kapal> kapals, int maksKeberangkatan) {
    for (Kapal kapal : kapals) {
        int sesi = 1;
        int keberangkatan = 1;
        while (!addRelasi(sesi, keberangkatan, kapal)) {
            if (keberangkatan < maksKeberangkatan) {
                keberangkatan++;
            }
            else {
                sesi++;
                keberangkatan = 1;
            }
        }
    }
}

```

Gambar 32. Kode program Backtracking deklarasi variabel

4.2.8 Algoritme *Steepest Hill Climbing*

Algoritma *Steepest Hill Climbing* digunakan untuk mencari jarak terpendek antar pelabuhan yang harus ditempuh pada jadwal yang beruntun pada jam tertentu. Hal ini bertujuan untuk meminimalisasi penggunaan waktu pada sesi kapal. Pada halaman data kapal tersedia tabel data kapal yang telah memenuhi sesi, keberangkatan, nama kapal, nama nahkoda, pelabuhan dan rute dari pencarian algoritme *Backtracking*. Untuk proses selanjutnya yaitu dengan melakukan generate *Steepest Hill Climbing*. Berikut merupakan potongan kode *Steepest Hill Climbing*

```

//get real route
double lowest = Double.MAX_VALUE;
Pelabuhan lowPelabuhan = null;
for (Pelabuhan pelabuhan : akhir.jarak.keySet()) {
    if (pelabuhan != awal && pelabuhan != akhir) {
        if (pelabuhan.jarakKe(akhir) + distance.get(pelabuhan) < lowest && !steepest.get(pelabuhan).contains(akhir)) {
            lowPelabuhan = pelabuhan;
            lowest = pelabuhan.jarakKe(akhir) + distance.get(pelabuhan);
        }
    }
}
rute.addAll(steepest.get(akhir));
rute.add(akhir);
LinkedList<Pelabuhan> inverse = new LinkedList<Pelabuhan>();
if (lowPelabuhan != null) {
    for (Pelabuhan pelabuhan : steepest.get(lowPelabuhan)) {
        inverse.addFirst(pelabuhan);
    }
}
rute.add(lowPelabuhan);
rute.addAll(inverse);
return rute;
}

```

Gambar 33. Kode program *Steepest Hill Climbing*

potongan *code* diatas berfungsi untuk menyimpan data jarak antar pelabuhan, penelusuran dari pelabuhan awal ke pelabuhan tujuan, untuk menemukan jarak terdekat dengan pelabuhan lainnya.

4.3 Testing

Pada sub bab ini akan dibahas mengenai pengujian yang dilakukan terhadap algoritme *Backtracking* dan *Steepest Hill Climbing*, yang berguna untuk memastikan bahwa tidak ada *constraint* yang dilanggar setelah jadwal selesai di *generate*.

4.3.1 Metode Test

Pengujian aplikasi menggunakan metode *Blackbox* testing. Metode *Blackbox* testing adalah pengujian yang dilakukan hanya mengamati eksekusi tanpa melihat isi program secara langsung dan juga pengujian ini dilakukan setelah aplikasi sudah selesai dibangun. Pengujian *Blackbox* testing berfungsi untuk memastikan bahwa aplikasi yang dapat dilihat oleh *user* dapat berjalan dengan baik tanpa menemukan *error*.

4.3.2 Tujuan Pengujian

1. Pengujian dilakukan untuk memastikan aplikasi yang telah dibangun berjalan dengan baik.
2. Memastikan bahwa aplikasi berhasil menghasilkan jadwal yang tidak melanggar *constraints* yang ada.

4.3.3. Test Plan

Subbab *Test plan* merupakan subbab yang menjelaskan bagaimana *Test* dilakukan dan perencanaan *Test* berdasarkan scenario pengujian dengan tujuan menjelaskan alur pengujian, hingga hasil akhir yang seharusnya diperoleh

4.3.3.1. Test Plan Pengujian Mengelola Data Kapal

Data kapal merupakan data dari. *Test plan* mengelola data kapal, bertujuan untuk merancang perencanaan *Test* yang bertujuan untuk memastikan proses pengelolaan data kapal, telah berjalan sesuai dengan yang diharapkan.

Table 21. Test Plan Pengujian Mengelola Data Kapal

Nama Kasus Uji	Mengelola data Kapal		
Tujuan	Menguji aplikasi apakah data dapat ditambahkan, diubah atau dihapus		
Deskripsi	Fungsi ini akan digunakan untuk mengelola data sumber daya kapal yang digunakan untuk melakukan <i>generate</i> jadwal.		
Kondisi Awal	User telah masuk pada halaman utama aplikasi		
Skenario Uji			
1. User memilih menu kelola kapal 2. User menambahkan data 3. User mengubah data 4. User menghapus data			
Kriteria Evaluasi Hasil			
User berhasil melakukan penambahan, perubahan dan penghapusan data.			
Kasus dan Hasil Uji (Data Normal)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
User menambahkan nama dan jumlah	Daftar data kapal akan bertambah dengan data yang telah diinput	Sesuai dengan yang diharapkan	Diterima
User mengubah nama dan jumlah	Daftar data kapal yang dipilih untuk diubah dapat berubah.	Sesuai dengan yang diharapkan	Diterima
User menghapus nama dan jumlah	Daftar data kapal yang dipilih untuk dihapus, berhasil dihapus	Sesuai dengan yang diharapkan	Diterima
Kasus dan Hasil Uji Coba (Data Tidak Normal)			
User menambahkan data tanpa mengisi nama dan jumlah	Tetap berada pada halaman kelola data kapal dan memberikan notifikasi bahwa data harus diisi	Sesuai yang diharapkan	Diterima
User mengubah data dengan cara mengosongkan salah satu inputan	Tetap berada pada halaman kelola data kapal dan memberikan notifikasi bahwa data harus diisi	Sesuai dengan yang diharapkan	Diterima
Catatan			

4.3.3.2. Test Plan Pengujian Mengelola Data Wilayah

Data Wilayah merupakan data dari. *Test plan* mengelola data wilayah, bertujuan untuk merancang perencanaan *Test* yang bertujuan untuk memastikan proses pengelolaan data wilayah, telah berjalan sesuai dengan yang diharapkan.

Table 22. Test Plan Pengujian Mengelola Data Wilayah

Nama Kasus Uji	Mengelola data Wilayah		
Tujuan	Menguji aplikasi apakah data dapat ditambahkan, diubah atau dihapus		
Deskripsi	Fungsi ini akan digunakan untuk mengelola data sumber daya wilayah yang digunakan untuk melakukan <i>generate</i> jadwal.		
Kondisi Awal	User telah masuk pada halaman utama aplikasi		
Skenario Uji			
1. User memilih menu kelola wilayah 2. User menambahkan data 3. User mengubah data 4. User menghapus data			
Kriteria Evaluasi Hasil			
User berhasil melakukan penambahan, perubahan dan penghapusan data.			
Kasus dan Hasil Uji (Data Normal)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
User menambahkan jarak, awal pelabuhan, tujuan pelabuhan	Daftar data wilayah akan bertambah dengan data yang telah diinput	Sesuai dengan yang diharapkan	Diterima
User mengubah jarak, awal pelabuhan, tujuan pelabuhan	Daftar data wilayah yang dipilih untuk diubah dapat berubah.	Sesuai dengan yang diharapkan	Diterima
User menghapus jarak, awal pelabuhan, tujuan pelabuhan	Daftar data wilayah yang dipilih untuk dihapus, berhasil dihapus	Sesuai dengan yang diharapkan	Diterima
Kasus dan Hasil Uji Coba (Data Tidak Normal)			
User menambahkan data tanpa mengisi jarak, awal pelabuhan, tujuan pelabuhan	Tetap berada pada halaman kelola data wilayah dan memberikan notifikasi bahwa data harus diisi	Sesuai yang diharapkan	Diterima

User mengubah data dengan cara mengosongkan salah satu inputan	Tetap berada pada halaman kelola data wilayah dan memberikan notifikasi bahwa data harus diisi	Sesuai dengan yang diharapkan	Diterima
Catatan			

4.3.3.3. Test Plan Pengujian Mengelola Data Jadwal

Data Jadwal merupakan data dari. *Test plan* mengelola data jadwal, bertujuan untuk merancang perencanaan *Test* yang bertujuan untuk memastikan proses pengelolaan data jadwal, telah berjalan sesuai dengan yang diharapkan.

Table 23. Test Plan Pengujian Mengelola Data Jadwal

Nama Kasus Uji	Mengelola data Jadwal		
Tujuan	Menguji aplikasi apakah data dapat ditambahkan atau dihapus		
Deskripsi	Fungsi ini akan digunakan untuk mengelola data sumber daya jadwal yang digunakan untuk melakukan <i>generate</i> jadwal.		
Kondisi Awal	User telah masuk pada halaman utama aplikasi		
Skenario Uji			
1. User memilih menu kelola jadwal 2. User menambahkan data 3. User menghapus data			
Kriteria Evaluasi Hasil			
User berhasil melakukan penambahan, perubahan dan penghapusan data.			
Kasus dan Hasil Uji (Data Normal)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
User menambahkan sesi, keberangkatan, dan nahkoda	Daftar data jadwal akan bertambah dengan data yang telah diinput	Sesuai dengan yang diharapkan	Diterima
User menghapus sesi, keberangkatan, dan nahkoda	Daftar data jadwal yang dipilih untuk dihapus, berhasil dihapus	Sesuai dengan yang diharapkan	Diterima
Kasus dan Hasil Uji Coba (Data Tidak Normal)			
User menambahkan data tanpa mengisi sesi, keberangkatan, dan nahkoda	Tetap berada pada halaman kelola data jadwal dan memberikan notifikasi bahwa data harus diisi	Sesuai yang diharapkan	Diterima
Catatan			

4.3.3.4 Test Plan Pengujian Generate Jadwal

Test plan pengujian *generate* jadwal bertujuan untuk membuat perancangan pengujian terkait

Table 24. Test Plan Pengujian Generate Jadwal

Nama Kasus Uji	<i>Generate</i> Jadwal
Tujuan	Menguji aplikasi apakah data untuk <i>generate</i> jadwal dapat ditambahkan.
Deskripsi	Fungsi ini akan digunakan untuk mengelola data sumber daya yang telah dimuat pada satu daftar jadwal yang terdiri atas data kapal, data wilayah, dan data jadwal yang digunakan untuk melakukan <i>generate</i> jadwal.
Kondisi Awal	<i>User</i> telah masuk pada halaman utama aplikasi
Skenario Uji	
1. <i>User</i> memilih menu kelola <i>generate</i> jadwal 2. <i>User</i> menambahkan maksimal keberangkatan pada 1 sesi 3. <i>User</i> mengenerate jadwal 4. <i>User</i> melakukan <i>export to excel</i>	
Kriteria Evaluasi Hasil	

User berhasil melakukan penambahan data relasi antar domain.

Kasus dan Hasil Uji (Data Normal)

Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
<i>User</i> melakukan <i>generate</i> jadwal	Menghasilkan data jadwal berdasarkan data jadwal yang sudah diinput dan maksimal keberangkatan dalam 1 sesi.	Sesuai dengan yang diharapkan	Diterima
<i>User</i> melakukan <i>export to excel</i>	Menghasil data jadwal dalam bentuk excel	Sesuai dengan yang diharapkan	Diterima
Catatan			

Berikut merupakan pengujian yang dilakukan untuk setiap *constraint* yang ada:

Table 25. Pengujian setiap constraint

No	<i>Constraint</i>	Keterangan
1	Setiap kapal yang berbeda tidak boleh berada pada pelabuhan yang sama dan waktu yang sama.	Diterima
2	Nahkoda tidak boleh mengemudi pada dua kapal atau lebih.	Diterima
3	Setiap kapal yang berbeda tidak boleh berada pada keberangkatan yang sama dan rute yang sama.	Diterima
4	Pada setiap sesi tidak boleh terdapat lebih kapal yang berangkat dari yang sudah ditentukan.	Diterima

BAB V

HASIL DAN PEMBAHASAN

Pada sub bab ini akan dijelaskan mengenai pembahasan dari hasil proses pembuatan jadwal penyeberangan kapal di Ajibata, yang dilakukan menggunakan Algoritme *Backtracking CSP* dan *Steepest Ascent Hill Climbing*.

1.1 Hasil Jadwal

Pada subbab ini dijelaskan mengenai hasil jadwal yang dihasilkan menggunakan metode *Backtracking CSP* dalam proses pencarian solusi untuk membentuk sebuah jadwal, dan algoritme *Steepest Hill Climbing* digunakan untuk mencari jarak terpendek.

1.2 Hasil Penjadwalan Kapal

Dengan menggunakan algoritme *backtracking*, telah dibentuk *constraints* yang berguna untuk membatasi tugas dan fungsi dari masing-masing sumber daya yang ada. Hasil jadwal dibagi atas Nama kapal, perusahaan, dan nama nahkoda. Berikut merupakan *output* jadwal dari aplikasi penjadwalan penyeberangan kapal di Ajibata.

Sesi	Keberangkatan	Kapal
1	1	MURNI
1	2	ROGANDA
1	1	Grace 3
1	1	AUSTIN 17
1	3	DOS ROHA I
1	2	HORAS
1	3	Doruli 03
1	4	Doruli 02
1	2	KANRO
2	1	Yosuaris 01

Gambar 34. Hasil penambahan jadwal kapal

Hasil penambahan jadwal diatas, merupakan hasil akhir data yang telah dimiliki oleh DISHUB Ajibata. Jadwal yang diperoleh merupakan hasil penambahan data dari relasi satu kapal hanya dikemudi oleh satu nahkoda saja. Proses yang dilalui untuk menciptakan jadwal penyeberangan kapal Ajibata yaitu dengan melakukan set domain, membangun relasi antar domain, dan menjalankan *constraint* yang ada. Algoritme *Backtracking* yang berfungsi untuk menemukan solusi terbaik dengan mempertimbangkan *constraint – constraint* yang ada.

1.3 Hasil Generate Jadwal Menggunakan Algoritme *Steepest Hill Climbing*

Setelah jadwal selesai di-generate tanpa rute menggunakan *Backtracking*. Pencarian rute dilakukan dengan menggunakan algoritme *Steepest Hill Climbing*. Berikut merupakan halaman aplikasi untuk melakukan generate *Steepest Hill Climbing*,

Sesi	Keberangkatan	Kapal	Nahkoda	Pelabuhan	
1	1	MURNI	Jojo Situm...	Tomok	Ajibata->Tomok
1	2	ROGANDA	Torang G...	Tomok	Ajibata->Tomok
1	1	Grace 3	Dedi Situn...	Tigaras	Ajibata->Tigaras
1	1	AUSTIN 17	Charles Pa...	Onanrunggu	Ajibata->Onanru
1	3	DOS ROHA I	Hasudung...	Tomok	Ajibata->Tomok
1	2	HORAS	Soon Para...	Tigaras	Ajibata->Tigaras
1	3	Doruli 03	Joni Huta...	Tigaras	Ajibata->Tigaras
1	4	Doruli 02	Andika Sa...	Tigaras	Ajibata->Tigaras
1	2	KANRO	Dandi Situ...	Onanrunggu	Ajibata->Onanru
2	1	Yosuaris 01	Andi Gult...	Tigaras	Ajibata->Tigaras

Gambar 35. Generate Steepest Hill Climbing

Setelah jadwal telah berhasil untuk di-*generate*, maka dicari rute menggunakan algoritme *Steepest Hill Climbing* untuk memperoleh jarak terdekat antara pelabuhan. Hasil generate jadwal diatas, merupakan hasil akhir dari penjadwalan penyeberangan kapal di pelabuhan Ajibata. Berikut ini merupakan hasil eksport generate hasil jadwal oleh aplikasi ke dalam file excel.

1	Sesi	Keberangkatan	Kapal	Nahkoda	Pelabuhan	Rute
2	1	1	MURNI	Jojo Situmorang	Tomok	Ajibata->Tomok->Simanindo->Ajibata
3	1	2	ROGANDA	Torang Gultom	Tomok	Ajibata->Tomok->Simanindo->Ajibata
4	1	1	Grace 3	Dedi Situngkir	Tigaras	Ajibata->Tigaras->Simanindo->Ajibata
5	1	1	AUSTIN 17	Charles Parapat	Onanrunggu	Ajibata->Onanrunggu->Sibandang->Ajibata
6	1	3	DOS ROHA I	Hasudungan Sihotan	Tomok	Ajibata->Tomok->Simanindo->Ajibata
7	1	2	HORAS	Soon Parapat	Tigaras	Ajibata->Tigaras->Simanindo->Ajibata
8	1	3	Doruli 03	Joni Hutagaol	Tigaras	Ajibata->Tigaras->Simanindo->Ajibata
9	1	4	Doruli 02	Andika Samosir	Tigaras	Ajibata->Tigaras->Simanindo->Ajibata
10	1	2	KANRO	Dandi Situmeang	Onanrunggu	Ajibata->Onanrunggu->Sibandang->Ajibata
11	2	1	Yosuaris 01	Andi Gultom	Tigaras	Ajibata->Tigaras->Simanindo->Ajibata
12	1	4	PULO HORAS	Linggom Sitinjak	Tomok	Ajibata->Tomok->Simanindo->Ajibata
13	2	2	Yosuaris 02	Bornok Simbolon	Tigaras	Ajibata->Tigaras->Simanindo->Ajibata
			TELAGA	Budiman Rumahorb	Onanrunggu	

Gambar 36. Hasil Export Generate Jadwal

BAB VI

KESIMPULAN dan SARAN

6.1 Kesimpulan

Dari hasil pengerjaan Tugas Akhir ini diperoleh bahwa

1. Algoritme *Backtracking* dengan *Constraint Satisfaction Problem* dapat dijadikan sebagai metode yang tepat, untuk menyelesaikan masalah penjadwalan karena menghasilkan solusi terbaik berdasarkan batasan dan aturan yang harus diikuti.
2. Algoritme *Steepest Hill Climbing* dapat digunakan untuk pencarian rute terpendek antar pelabuhan.
3. Aplikasi yang telah dibangun, mampu menghasilkan jadwal penyeberangan kapal di pelabuhan Ajibata yang memperhatikan constraint.

6.2 Saran

Saran yang diperhatikan dalam pengembangan Tugas Akhir selanjutnya adalah:

1. Berdasarkan hasil pengujian, data yang terlalu besar mengakibatkan kebutuhan perangkat keras dengan spesifikasi yang lebih tinggi. Untuk permasalahan data yang besar, dibutuhkan metode untuk meningkatkan kinerja komputer.
2. Kombinasi data dari keseluruhan sumber daya disimpan dalam database untuk diproses. Sehingga saat aplikasi dijalankan dengan kombinasi data yang banyak, jadwal tetap dapat di-generate.

DAFTAR PUSTAKA

- [1] R. Febriyana dan W. F. Mahmudy, “Penjadwalan Kapal Penyeberangan Menggunakan Algoritma Genetika,” *J. Teknol. Inf. dan Ilmu Komput.*, vol. 3, no. 1, hal. 43, 2016.
- [2] J. Gerald, K. Suhada, dan D. T. Liputra, “Usulan Algoritma Penjadwalan Pengiriman Produk Di PT Ultra Jaya Milk Industry & Trading Company Tbk,” *J. Integr. Syst.*, vol. 2, no. 1, hal. 1–20, 2019.
- [3] S. Kasus *et al.*, “Perbandingan Algoritma Hill Climbing Dan Algoritma Ant Colony Dalam Penentuan Rute Optimum Comparison of Hill Climbing Algorithm and Ant Colony Algorithm in Determining Optimum Route,” vol. 11, hal. 139–150, 2017.
- [4] Novriyanto dan M. Zaid.S, “Penerapan Algoritma Backtracking Berbasis Blind Search untuk Menentukan Penjadwalan Mengajar,” *Semin. Nas. Apl. Teknol. Inf. 2013*, hal. 13–18, 2013.
- [5] R. Lumbantoruan, Y. N. Simatupang, M. N. Siahaan, M. H. Pardede, dan J. Pakpahan, “Penjadwalan Kuliah dengan Algoritma Backtracking (Penjadwalan Kuliah dengan Algoritma Backtracking,” no. March, 2017.
- [6] D. Fitriati dan N. M. Nessrayasa, “Implementasi Algoritma Hill Climbing Pada Penentuan Jarak Terpendek Kota Wisata Di Indonesia,” *J. Ris. Inform.*, vol. 1, no. 3, hal. 127–132, 2019.
- [7] Y. Adharani, E. Susilowati, dan E. Purwanto, “Penerapan Metode Simple Hill Climbing Search Untuk Pencarian Lokasi Terdekat Sekolah Menengah Atas Muhammdiyah,” *Sist. Informasi, Teknol. Inform. dan Komput.*, vol. 7, no. 2, hal. 15, 2017.
- [8] I. G. A. Widyadana dan R. Wibisono, “Penentuan Jadwal dan Rute Perjalanan Kapal dengan Batasan Waktu dan Jumlah Kunjungan,” *J. Tek. Ind.*, vol. 18, no. 2, hal. 123–128, 2016.
- [9] L. A. Zemma, Herfina, dan A. Qur, “Penerapan Metode Steepest Ascent Hill Climbing Pada Model Pencarian Rute Terdekat Fasilitas Pelayanan Darurat Di Kota Bogor Berbasis Android,” 2015.
- [10] F. Makarim, “Analisis Penggunaan Algoritma Backtracking dalam Penjadwalan Kuliah.” 2017.

- [11] A. M. Sodik, A. Mustakim, dan P. Wuryaningrum, “Analisis Perencanaan Pola Operasi Armada Kapal Penyeberangan: Studi Kasus Pelabuhan Ujung-Kamal,” *J. Tek. ITS*, vol. 8, no. 1, hal. 100–104, 2019.
- [12] E. Elvina dan L. Hakim, “Modifikasi Algoritma Steepest-Ascent Hill Climbing Dan Backtracking Untuk Pencarian Lintasan Kritis Proyek,” *CogITO Smart J.*, vol. 4, no. 2, hal. 268, 2019.
- [13] A. Saifudin dan U. Pamulang, “Penjadwalan Perkuliahan Menggunakan Algoritma Hill Climbing,” no. February 2017, 2018.
- [14] M. R. Firdaus, S. I. Halim, A. Hill, C. Search, dan L. Tedekat, “Penerapan Metode Hill Climbing Search untuk Pencarian Lokasi Terdekat pada Aplikasi Toko Virtual Berbasis Android,” *Business*, no. Pencarian Lokasi Terdekat, hal. 88–97, 2014.
- [15] D. Tobasa, “Letak Geografis Kecamatan Ajibata,” 2016. [Daring]. Tersedia pada: https://tobasamosirkab.bps.go.id/backend/pdf_publikasi/kecamatan-ajibata-dalam-angka-2016.pdf. [Diakses: 18-Sep-2019].
- [16] J. T. Rahayu, “Penjadwalan kapal di pelabuhan Ajibata,” 2019. [Daring]. Tersedia pada: <https://www.antaranews.com/berita/804422/layani-kawasan-danau-toba-asdp-pastikan-aspek-keselamatan-kapal-ihan-batak>. [Diakses: 20-Sep-2019].
- [17] W. Hendrastiti *et al.*, “Penjadwalan tenaga kerja proyek kereta barang PT. XYZ menggunakan algoritma genetika,” *Tek. Pomits*, vol. 1, no. 1, hal. 1–4, 2012.
- [18] F. Utama, A. H. Kridalaksana, dan I. F. Astuti, “Implementasi Backtracking Algorithm Untuk Penyelesaian Permainan Su Doku Pola 9x9,” *Inform. Mulawarman J. Ilm. Ilmu Komput.*, vol. 11, no. 1, hal. 29, 2016.
- [19] A. Pratama, “Analisis Penerapan Algoritma Backtracking Pada Pencarian Jalan Keluar di Dalam Labirin Prinsip Pencarian Solusi dengan Metode Runut-Balik,” no. 16, 2007.
- [20] Y. Rifqo, Muhammad Husni Apridiansyah, “Implementasi Algoritma Backtracking Untuk Pencarian Judul Buku,” 2017.
- [21] N. I. Lesmana, “Penjadwalan Produksi Untuk Meminimalkan Waktu

- Produksi Dengan Menggunakan Metode Branch And Bound,” *J. Tek. Ind.*, vol. 17, no. 1, hal. 42, 2017.
- [22] H. Anam, F. S. Hanafi, A. F. Adifia, A. F. Ababil, dan S. Bukhori, “Penerapan Metode Steepest Ascent Hill Climb pada Permainan Puzzle,” *INFORMAL Informatics J.*, vol. 3, no. 2, hal. 36, 2018.
- [23] B. Choudhury *et al.*, “Game slider puzzle,” *Pros. Semin. Nas. Multidisiplin Ilmu*, vol. 2, no. 1, hal. 65–73, 2015.
- [24] A. Mauluddin, H. P. Utomo, dan S. A. Dewi, “Implementasi Algoritma Steepest Ascent Hill Climbing Pada Permainan Slide Puzzle Berbasis Android,” vol. 2016, no. Sentika, hal. 18–19, 2016.
- [25] H. Waskita, H. D. Purnomo, dan H. Hendry, “Penjadwalan Kelas Praktikum Menggunakan Algoritma Steepest Ascent Hill Climbing,” *J. Teknol. Informasi-AITI*, vol. 13, no. 2, hal. 153–168, 2016.

LAMPIRAN

Wawancara dengan Kepala Dinas Perhubungan Ajibata.

Nama ~~Kepala~~ : Sawaldi S. Babukan ,

Hari/Tanggal wawancara : Sabtu, 07 Desember 2019

Lokasi wawancara : Pelabuhan Ajibata

1. Adakah sistem penjadwalan yang terdapat di pelabuhan Ajibata?
2. Bagaimana system penjadwalannya?
3. Apakah semua kapal di Ajibata dikelola oleh Dishub? Jika tidak, mengapa?
4. Apakah ada peraturan tertentu yang menetapkan bahwa Dishub yang mengelola semua kapal di Ajibata ini? Jika tidak, apakah peraturan itu akan dibuat?
5. Data-data apa saja kah yang dikelola di Pelabuhan Ajibata ini? Sebutkan.
6. Ada berapakah rute perjalanan yang berawal dari pelabuhan Ajibata?
7. Berdasarkan apakah pemilihan rute tersebut?
8. Bagaimana lintasannya? Mengapa lintasannya seperti itu?
9. Apakah jadwal yang sekarang berlaku untuk hari besar? Jika tidak, bagaimana jadwal di hari besar?
10. Berapa jarak antar wilayah?

Diketahui oleh,

Kepala Dinas Perhubungan Ajibata /Anggota

(Sawaldi S. Babukan)

Sawaldi S. Babukan

Gambar 37. hasil wawancara dari DISHUB

12

tomok -> Nainggolan -> $\frac{24,51}{16,668} = 87,50$
 $= 1 \text{ jam } 27 \text{ menit}$

Nainggolan -> Ajibata -> $\frac{25,29}{16,668} = 91 \Rightarrow 1 \text{ jam } 31 \text{ menit}$

Onanunggu -> Mura -> $\frac{10,08}{16,668} = \frac{36,28}{16,668} = 36$
 $= 36 \text{ menit}$

Mura -> Ajibata -> $\frac{26,65}{16,668} = 1 \text{ jam } 35 \text{ menit}$

Ajibata-tigeras -> Ajibata-tigeras-jinanindo-Ajibata.
 $22,60 + 6,90 + 23,71 = 53,21 \text{ km.} \Rightarrow 3 \text{ jam } 13 \text{ menit.}$

Ajibata-tomole -> Ajibata-tomole-tigeras-Ajibata.
 $8,43 + 22,56 + 22,60 = 53,59. \Rightarrow 3 \text{ jam } 21 \text{ menit.}$

Ajibata-Onanunggu -> Ajibata-Onanunggu-Jibadang-Ajibata.
 $23,76 + 9,08 + 21,65 = 54,49 \Rightarrow 3 \text{ jam } 44 \text{ menit.}$

Gambar 38. hasil perhitungan jarak