TUGAS AKHIR – IF184802

DESAIN DAN ANALISA ALGORITMA KOMPUTASI INTEGRASI NUMERIK : STUDI KASUS UVA ONLINE JUGDE ENVIRONMENT PROTECTION

MAGDALENA ANDINIWARIH ISMANU
05111640000112

Dosen Pembimbing I :
Rully Soelaiman, S.Kom., M.Kom.

Dosen Pembimbing II :
Yudhi Purwananto, S.Kom, M.Kom.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya
2020

TUGAS AKHIR – IF184802

DESAIN DAN ANALISA ALGORITMA KOMPUTASI INTEGRASI NUMERIK : STUDI KASUS URI ONLINE JUGDE ENVIRONMENT PROTECTION

MAGDALENA ANDINIWARIH ISMANU
05111640000112

Dosen Pembimbing I :
Rully Soelaiman, S.Kom., M.Kom.

Dosen Pembimbing II :
Yudhi Purwananto, S.Kom, M.Kom.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya
2020

TUGAS AKHIR – IF184802

DESAIN DAN ANALISA ALGORITMA KOMPUTASI INTEGRASI NUMERIK : STUDI KASUS UVA ONLINE JUGDE ENVIRONMENT PROTECTION

MAGDALENA ANDINIWARIH ISMANU
05111640000112

Dosen Pembimbing I :
Rully Soelaiman,S.Kom., M.Kom.

Dosen Pembimbing II :
Yudhi Purwananto, S.Kom, M.Kom.

DEPARTEMEN TEKNIK INFORMATIKA
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember
Surabaya
2020

[*Halaman ini sengaja dikosongkan*]

UNDERGRADUATE THESIS- IF184802

DESAIN AND ANALYSIS COMPUTATION INTEGRATION NUMERIC ALGORITHM : STUDI CASE UVA ONLINE JUDGE ENVIRONMENT PROTECTION

MAGDALENA ANDINIWARIH ISMANU
05111640000112

Supervisor I
Rully Soelaiman, S.Kom., M.Kom.

Supervisor II
Yudhi Purwananto, S.Kom, M.Kom.

DEPARTMENT OF INFORMATICS
Faculty of Intelligent Electrical and Information Technology
Institut Teknologi Sepuluh Nopember
Surabaya
2020

[*Halaman ini sengaja dikosongkan*]

LEMBAR PENGESAHAN

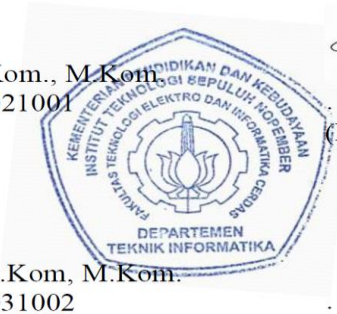
DESAIN DAN ANALISA ALGORITMA KOMPUTASI INTEGRASI NUMERIK : STUDI KASUS UVA ONLINE JUGDE ENVIRONMENT PROTECTION

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Algoritma Pemrograman
Program Studi S-1 Teknik Informatika
Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember

Oleh:
Magdalena Andiniwarih Ismanu
NRP: 051116 40000 112

Disetujui oleh Dosen Pembimbing Tugas Akhir:



Rully Soelaiman, S.Kom., M.Kom.
NIP. 197002131994021001 .....
(Pembimbing 1)

Yudhi Purwananto, S.Kom, M.Kom.
NIP. 197007141997031002 .....
(Pembimbing 2)

[*Halaman ini sengaja dikosongkan*]

DESAIN DAN ANALISA ALGORITMA KOMPUTASI INTEGRASI NUMERIK : STUDI KASUS UVA ONLINE JUGDE ENVIRONMENT PROTECTION

Nama Mahasiswa : Magdalena Andiniwarih Ismanu
NRP : 051116 40000 112
Departemen : Teknik Informatika Fakultas Teknologi Elektro dan Informatika Cerdas - ITS
DosenPembimbing 1 : Rully Soelaiman, S.Kom., M.Kom.
DosenPembimbing 2 : Yudhi Purwananto, S.Kom, M.Kom.

Abstrak

Permasalahan Environment Protection adalah permasalahan untuk mencari luas grafik. Permasalahan asli dari Permasalahan Environment Protection adalah mencari kedalaman pada pengeboran minyak untuk memaksimalkan keuntungan.

Pada tugas akhir ini, akan dirancang penyelesaian permasalahan Environment Protection dalam mencari kedalaman yang harus digali agar luas daerah sesuai dengan luas daerah yang diinginkan. Permasalahan ini dapat diselesaikan dengan menggunakan integrasi numerik dan pencarian biner.

Hasil dari tugas akhir ini telah berhasil menyelesaikan permasalahan diatas dengan cukup efisien, dengan rata-rata waktu penyelesaian 0,075 detik.

[*Halaman ini sengaja dikosongkan*]

DESAIN DAN ANALISA ALGORITMA KOMPUTASI INTEGRASI NUMERIK : STUDI KASUS UVa ONLINE JUGDE ENVIRONMENT PROTECTION

Student Name	: Magdalena Andiniwarih Ismanu
NRP	: 051116 40000 112
Department	: Informatics Engineering Department Faculty of Intelligent and Informatics Technology - ITS
First Supervisor	: Rully Soelaiman, S.Kom., M.Kom.
Second Supervisor	: Yudhi Purwananto, S.Kom, M.Kom.

Abstract

The problem of Environment Protection is the problem of finding the area of a graph. The original problem with the Environment Protection Problem is finding depth in oil drilling to maximize profits.

In this thesis, an environmental protection problem will be designed to find the depth that must be explored so that the area is in accordance with the desired area. This problem can be solved by using numerical integration and binary search.

The results of this thesis have successfully resolved the above problems quite efficiently, with an average completion time of 0.075 seconds.

[*Halaman ini sengaja dikosongkan*]

KATA PENGANTAR

Puji syukur penulis ucapkan kepada Tuhan Yang Maha Esa atas pimpinan, penyertaan, dan karunia-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul:

DESAIN DAN ANALISA ALGORITMA KOMPUTASI INTEGRASI NUMERIK : STUDI KASUS UVa ONLINE JUGDE ENVIRONMENT PROTECTION

Pengerjaan Tugas Akhir ini dilakukan untuk memenuhi salah satu syarat meraih gelar Sarjana di Departemen Teknik Informatika Fakultas Teknologi Informasi dan Komunikasi Institut Teknologi Sepuluh Nopember.

Dengan selesainya Tugas Akhir ini diharapkan apa yang telah dikerjakan penulis dapat memberikan manfaat bagi perkembangan ilmu pengetahuan terutama di bidang teknologi informasi serta bagi diri penulis sendiri selaku peneliti.

Penulis mengucapkan terima kasih kepada semua pihak yang telah memberikan dukungan baik secara langsung maupun tidak langsung selama penulis mengerjakan Tugas Akhir maupun selama menempuh masa studi antara lain:

1. Bapak Rully Soelaiman, S.Kom., M.Kom. selaku Dosen Pembimbing yang telah membimbing saya selama masa kuliah maupun selama penyelesaian Tugas Akhir ini, Dosen yang paling perhatian kepada saya dalam memberi ilmu, nasihat, dan motivasi selama berada menempuh kuliah di Departemen Informatika ITS.
2. Bapak Yudhi Purwananto, S.Kom, M.Kom. selaku dosen pembimbing yang telah memberikan ilmu, dan masukan kepada penulis.
3. Bapak, Ibu, dan keluarga penulis yang selalu memberikan dukungan, perhatian, dan kasih sayang bagi penulis yang menjadi semangat selama perkuliahan maupun pengerjaan Tugas Akhir.

4. Teman-teman angkatan 2016 Departemen Teknik Informatika ITS yang telah menemani perjuangan penulis selama empat tahun masa perkuliahan.
5. Yoshima yang telah mengkoreksi Tugas Akhir dari penulis.
6. Serta pihak-pihak lain yang tidak dapat disebutkan disini yang telah banyak membantu penulis dalam penyusunan Tugas Akhir ini.

Penulis mohon maaf apabila masih ada kekurangan pada Tugas Akhir ini. Penulis juga mengharapkan kritik dan saran yang membangun untuk pembelajaran dan perbaikan di kemudian hari. Semoga melalui Tugas Akhir ini penulis dapat memberikan kontribusi dan manfaat yang sebaik-baiknya.

Surabaya, 2020

Magdalena Andiniwarih Ismanu

DAFTAR ISI

DAFTAR GAMBAR

[*Halaman ini sengaja dikosongkan*]

DAFTAR TABEL

DAFTAR PSEUDOCODE

DAFTAR KODE SUMBER

[*Halaman ini sengaja dikosongkan*]

BAB I
PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar tugas akhir yang meliputi latar belakang, tujuan, rumusan masalah, batasan permasalahan, metodologi pembuatan tugas akhir, dan sistematika penulisan buku tugas akhir ini.

1.1 Latar Belakang

Banyak dari kondisi alam di Indonesia yang memiliki sumber daya alam dapat dimanfaatkan untuk kesejahteraan rakyat. Perlunya teknologi informasi untuk mempermudah pemanfaatan sumber daya alam dan menanggulangi pencemaran lingkungan yang terjadi akibatnya. Oleh karena itu perlu adanya sistem informasi yang presisi dan optimal yang dapat mengatasi masalah tersebut.

Topik tugas akhir ini mengacu pada pemanfaatan metode numerical integration untuk mencari luasan grafik pada soal UVA Online Judge kode 1297 *Environment Protection*. Modal dari permasalahan ini adalah diberikannya dua grafik *f(x)* yang merupakan daerah pengeboran minyak dan juga diberikan lebar dan kedalaman dari daerah. Pertanyaan yang mendasari permasalahan ini adalah pada kedalaman berapa harus melakukan pengeboran minyak jika daerah yang terdapat minyak sesuai dengan yang diinginkan user. Pendekatan yang akan digunakan adalah mencari luas daerah yang dibatasi oleh *yi(x) = pi(x)/qi(x)*, dan menentukan kedalaman dari daerah yang akan digali.

Hasil yang diharapkan dari tugas akhir ini adalah dapat menjadi gambaran tentang sistem yang digunakan untuk pengeboran minyak dengan hasil yang akurat dengan menggunakan metode *Numerical Integration* dan diharapkan dapat berkontribusi pada perkembangan ilmu dan teknologi.

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini adalah sebagai berikut:

1. Bagaimana menganalisis serta menentukan desain dan algoritma yang akurat dan optimal pada soal UVA Online Judge 1297 *Environment Protection*?
2. Bagaimana mengimplementasikan desain dan algoritma yang akurat dan optimal pada soal UVA Online Judge 1297 *Environment Protection*?
3. Bagaimana hasil dari kinerja algoritma yang digunakan untuk menyelesaikan permasalahan pada soal UVA Online Judge 1297 *Environment Protection*?

1.3 Batasan Masalah

Permasalahan yang dibahas pada Tugas Akhir ini memiliki beberapa batasan, yaitu sebagai berikut:

Pustaka yang digunakan untuk membantu pengimplementasian algoritma merupakan C++ *Standard Template Library* (STL). Pustaka-pustaka tersebut antara lain: *cstdio* dan *cstring*.

Adapun batasan dalam UVa Online Judge untuk Tugas Akhir ini adalah sebagai berikut :

1. $\boldsymbol{W}$ adalah lebar dari area pengeboran dengan batasan ($1 \leq \boldsymbol{W} \leq 8$)
2. $\boldsymbol{D}$ adalah kedalaman ($1 \leq \boldsymbol{D} \leq 10$)
3. $\boldsymbol{A}$ adalah luas yang diinginkan($1 \leq \boldsymbol{A} \leq \boldsymbol{W} \; x \; \boldsymbol{D}$)
4. $\boldsymbol{K}$ adalah fungsi grafik *y1(x)* dan *y2(x)* ($0 \leq \boldsymbol{K} \leq 8$).
5. *Dataset* yang digunakan adalah *dataset* pada problem UVA Online Judge 1297*Environment Protection*

1.4 Tujuan

Tujuan dari pembuatan Tugas Akhir ini adalah sebagai berikut:

1. Melakukan analisis dan desain algoritma yang tepat untuk permasalahan UVA Online Judge 1297 *Environment Protection*.
2. Melakukan implementasi algoritma yang tepat untuk permasalahan UVA Online Judge 1297 *Environment Protection*.
3. Melakukan evaluasi kinerja algoritma untuk permasalahan UVA Online Judge 1297 *Environment Protection*.
4. Melakukan algoritma yang efisien untuk permasalahan UVA Online Judge 1297 *Environment Protection*.

1.5 Manfaat Tugas Akhir

Tugas Akhir ini diharapkan dapat meningkatkan pemahaman tentang *Numerical Integration* untuk penentuan luas grafik yang akurat dan optimal sehingga dapat memberikan kontribusi terhadap perkembangan teknologi dan informasi.

1.6 Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini yaitu:

1.6.1 Penyusunan Proposal TugasAkhir

Tahap awal untuk memulai pengerjaan Tugas Akhir adalah penyusunan proposal Tugas Akhir. Pada proposal ini, penulis mengajukan gagasan untuk menyelesaikan permasalahan pada studi kasus UVa *Environment Protection* dengan teori pengkodean dan repetisi kode biner.

1.6.2 Studi Literatur

Pada tahap ini dilakukan pencarian informasi dan studi literatur yang relevan untuk dijadikan referensi dalam melakukan pengerjaan Tugas Akhir. Informasi didapatkan dari materi-materi yang berhubungan dengan struktur data. Informasi tersebut didapatkan dari buku, internet, dan materi kuliah yang berhubungan dengan metode yang akan digunakan.

1.6.3 Implementasi Algoritma

Tahapan ini merupakan tahapan untuk membangun algoritma yang akan digunakan. Pada tahap ini dilakukan implementasi dari rancangan struktur data yang akan dimodelkan sesuai dengan permasalahan. Implementasi ini dilakukan dengan menggunakan bahasa pemrograman C++.

1.6.4 Pengujian dan Evaluasi

Pada tahap ini dilakukan uji coba kebenaran dan kinerja solusi yang telah diimplementasikan dengan melakukan pengiriman sumber kode sistem ke situs penilaian UVa Online Judge pada permasalahan yang terkait unutk mendapatkan umpan balik. Pengujian dilakukan dengan membandingkan kompleksitas hasil uji coba dengan kompleksitas hasil analisis.

1.6.5 Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi algoritma yang telah dibuat.

1.7 Sistematika Penulisan

Buku Tugas Akhir ini merupakan laporan secara lengkap mengenai Tugas Akhir yang telah dikerjakan baik dari sisi teori, rancangan, maupun implementasi sehingga memudahkan bagi pembaca dan juga pihak yang ingin mengembangkan lebih lanjut.

Sistematika penulisan buku Tugas Akhir secara garis besar dapat dilihat seperti dibawah ini.

Bab I Pendahuluan

Bab ini berisi penjelasanlatar belakang, rumusan masalah, batasan masalah dan tujuan pembuatan Tugas Akhir. Selain itu, metodologi pengerjaan dan sistematika penulisan laporan Tugas Akhir juga dijelaskan di dalamnya.

Bab II Dasar Teori

Bab ini berisi penjelasan secara detil mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

Bab III Desain dan Analisis

Bab ini berisi penjelasan tentang rancangan dari sistem yang akan dibangun.

Bab IV Implementasi

Bab ini berisi penjelasan implementasi dari rancangan yang telah dibuat pada bab III. Implementasi disajikan dalam bentuk *pseudocode* disertai dengan penjelasannya.

Bab V Pengujian dan Evaluasi

Bab ini berisi penjelasan mengenai data hasil percobaan dan pembahasan mengenai hasil percobaan yang telah dilakukan.

Bab VI Kesimpulan dan Saran

Bab ini merupakan bab terakhir yang menjelaskan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

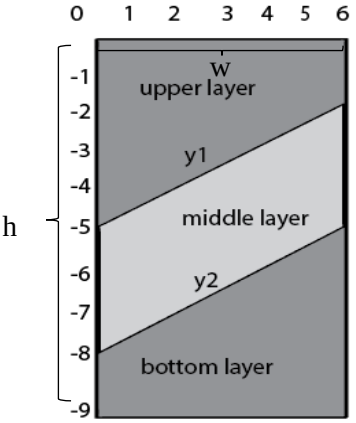
[*Halaman ini sengaja dikosongkan*]

BAB II
DASAR TEORI

Bab ini berisi pembahasan tentang dasar teori yang digunakan sebagai teori pendukung tugas akhir ini. Terlebih dahulu akan dijelaskan mengenai kasus pencarian ulan, kemudian dilanjutkan penjelasan permasalahan UVa Environment Protection yaitu teori pengkodean dan kode biner.

2.1 Deskripsi Umum Permasalahan

Permasalahan yang diangkat pada Tugas Akhir ini adalah permasalahan pencarian kedalaman penggalian agar luas daerah sesuai dengan yang diinginkan. Permasalahan ini akan merepresentasikan fungsi grafik batas atas dan grafik batas bawah dari daerah pengeboran minyak. Dari grafik tersebut akan dicari daerah perpotongan antar dua grafik. Dan dicari kedalaman dalam pengeboran agar luas yang dicari sesuai dengan yang diinginkan. Sebagai contoh, persoalan sederhana dengan simulasi penyelesaian untuk menentukan kedalam dari penggalian:

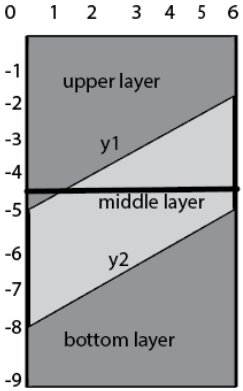


Gambar 2.1 Permasalahan Environment Protection

Gambar 2.1 adalah sebidang tanah yang memiliki lebar *w* 6 , kedalaman maksimal yang bisa digali adalah 9. Pengusaha

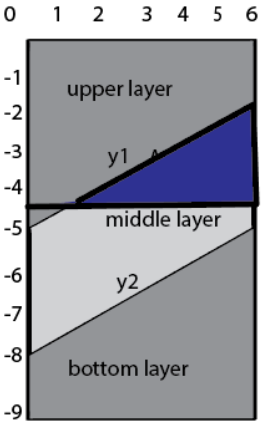
ingin agar daerah *middle layer* memiliki luas 4 dengan batas grafik y1 dan y2 yang repesentasikan pada Gambar 2.1

Metode pertama penyelesaian adalah penetuan kedalaman dengan *Binary search.*Pencarian terus berlanjut sampai luas daerah sesuai yang diinginkan.



Gambar 2.2 Penyelesaian masalah dengan Environment Protection

Metode yang kedua adalah *Simpson 1/3* adalah algoritma untuk mencari luas daerah di bawah garis. Pada permasalahan ini luas yang di cari adalah grafik dibawah y1 dengan batas D.



Gambar 2.3 Penyelesaian masalah dengan Simpson 1/3

Setelah melakukan pencarian luas daerah, yang dilakukan selanjutnya adalah membandingkan luas tersebut dengan luas yang diinginkan oleh pengusaha.

Tabel 2.1 Ilustrasi iterasi *Binary Search*

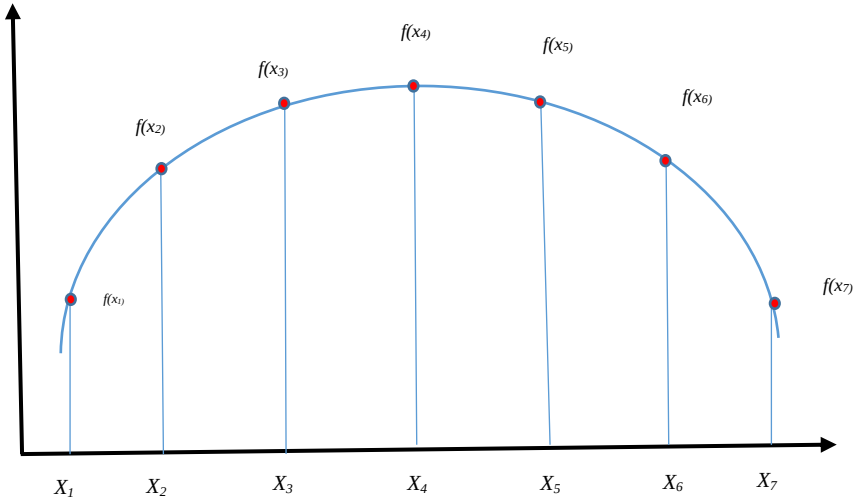
Iterasi	Kedalaman
1	-4.50
2	-2.25
3	-3.37
4	-3.93
5	-4.21
6	-4.07
7	-4.00

Iterasi berhenti setelah ditemukan luas sama dengan yang dicari dan ditemukan kedalamannya adalah -4.

2.2 Definisi Umum

Kedalaman dari tanah didefinisikan sebagai batas bawah dari grafik agar luas daerah perpotongan sesuai dengan yang diinginkan. Thomas Simpson mengusulkan algoritma *simpson 1/3* untuk menentuka luasan grafik.

Dengan menentukan batas atas dan batas bawah grafik yang akan dihitung dan membaginya menjadi beberapa interval untuk menimimalkan jumlah *error*.



Gambar 2.4 Ilustrasi algoritma Simpson 1/3

2.3 *Jumlah Partisi(Number of Interval)*

Pada grafik yang tidak beraturan perlu adanya beberapa partisi untuk meminimalkan *Approximate Error* ketika penghitungan. Nilai dari perhitungan luas setiap partisi grafik akan dijumlahkan. Sehingga solusi untuk masalah menemukan kedalaman tanah yaitu menghitung setiap kemungkinan yang terjadi agar luas grafik sesuai yang diinginkan.

2.4 *Approximate dan Relative Approximate Errors*

Seringkali nilai sebenarnya tidak diketahui, terutama dalam komputasi numerik. Dalam hal ini kita harus menghitung kesalahan hanya menggunakan nilai perkiraan. Ketika metode iteratif digunakan, mendapatkan nilai perkiraan pada akhir setiap iterasi.

Approximate error (E_a) = present approximation – previous approximation (2.1)

didefinisikan sebagai perbedaan antara nilai perkiraan saat ini dan perkiraan sebelumnya (misal Perubahan antara iterasi).

Relative approximate error (ϵ_a) = approximate error / present approximation (2.2)

2.5 Simpson 1/3

Perkiraan integral yang lebih akurat adalah menggunakan polinomial tingkat tinggi untuk menghubungkan antar titik. Misalnya, jika ada titik lain di antara *f(a)* dan *f(b)*, ketiga titik dapat dihubungkan dengan parabola. Rumus yang dihasilkan dari mengambil integral di bawah polinomial ini disebut aturan Simpson.

Berikut adalah rumus untuk penghitungan *Simpson 1/3*

$$I = (b - a)\frac{f(x0)+4f(x1)+f(x2)}{6} \qquad (2.3)$$

dimana *a* adalah *lower limit* atau batas bawah dari partisi grafik, *b = upper limit* adalah batas atas dari partisi grafik, dan *x1* = titik di tengah antara *a*(*lower limit)* dan *b*(*upper limit)* .

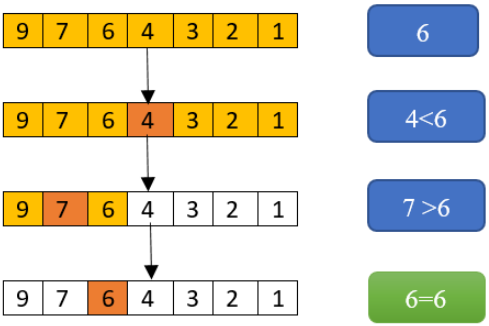
1.	**Function** *simpson_formula(l, r, p[], q[], y*)
2.	**Data:** *l* left point of graph, *right* right point of graph,
3.	*p[]* first function, *q[]* second function, *y* depth
4.	**Result:** *answer*
5.	*answer = function (l, p, q, y) + 4 * function ((l +r) / 2, p ,*
6.	*q, y) + function (r, p, q , y)) * (r – l) / 6*
7.	***return*** *answer*
8.	***end***

Gambar 2.5 Pseudocode Simpson 1/3

2.6 *Binary Search*

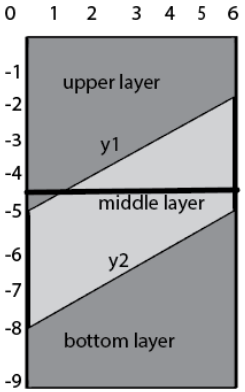
Binary Search adalah algoritma pencarian yang menemukan posisi nilai target dalam *array* yang diurutkan. *Binary Search* membandingkan nilai target dengan elemen

tengah *array*. Jika mereka tidak sama, setengah dari target dihilangkan dan pencarian berlanjut pada setengah sisanya, lagi mengambil elemen tengah untuk membandingkan dengan nilai target, dan mengulanginya sampai nilai target ditemukan. Jika pencarian berakhir dengan separuh sisanya kosong, target tidak dalam *array*.



Gambar 2.6 Ilustrasi penyelesaian algoritma *Binary Search*

Pencarian biner lebih cepat daripada pencarian linier kecuali untuk *array* kecil. Namun, *array* harus disortir terlebih dahulu untuk dapat menerapkan pencarian biner. Contoh dari penerapan *binary serach* pada permasalahan.



Gambar 2.7 Ilustrasi penerapan *Binary Search*

Setelah menentukan kedalaman awal atau titik tengah dari kedalaman maksimal yaitu -4.5. Iterasi terus berlangsung sampai luas yang diperoleh sesuai dengan luas yang diminta.

2.7 Permasalahan Environment Protection pada UVa Online Judge

Environment Protection merupakan suatu permasalahan yang terdapat pada situs penilaian *UVa Online Judge* dengan deskripsi soal dari sumber asli menggunakan bahasa Inggris

Penambangan Arsenik & Sianida (ACM) adalah perusahaan yang memutuskan untuk mulai mengembangkan tambang di tanah dekat kota asal. Sebagai anggota komite pengaturan warga negara untuk operasi ACM, tugas Anda adalah mengendalikan seberapa banyak perusahaan dapat menambang dari tanah tersebut, sehingga Anda dapat mempertahankan pekerjaan dan manfaat lainnya tanpa mengorbankan lingkungan dan kesehatan penduduk setempat.

ACM memiliki rencana untuk menambang beberapa bidang tanah persegi panjang. Sepetak tanah memiliki lebar *W*, dapat digali hingga kedalaman maksimum *D*, dan memiliki permukaan dianggap berada di kedalaman 0. Mineral dalam tambalan disusun dalam tiga lapisan, yang dapat bervariasi dalam kedalamannya sepanjang lebar tambalan, tetapi selalu memiliki profil yang sama sepanjang panjangnya. Inilah sebabnya mengapa ACM hanya tertarik pada profil sepanjang lebar setiap tambalan, dan telah melakukan pekerjaan eksplorasi untuk menentukan bentuknya secara tepat.

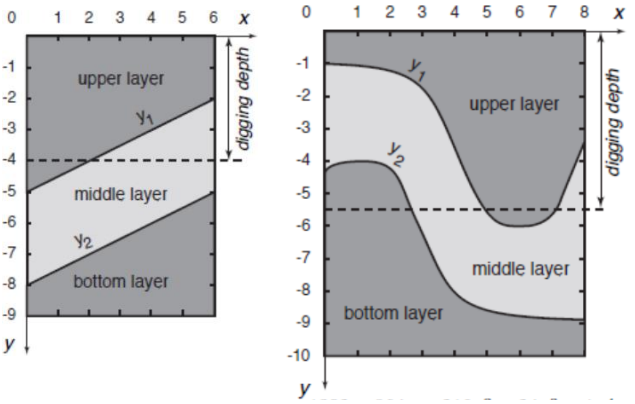
Akibatnya, mereka menemukan bahwa dua antarmuka antara tiga lapisan mineral dapat diwakili oleh dua fungsi $y_1(x)$ dan $y_2(x)$, di mana yang pertama menggambarkan batas antara lapisan atas dan lapisan tengah, dan kedua menggambarkan batas antara lapisan tengah dan lapisan bawah. Fungsi-fungsi ini selalu seperti itu.

$$-D < y_2(x) < y_1(x) < 0 \text{ for } 0 \leq x \leq W , \qquad (2.4)$$

Sehingga batas-batas lapisan tidak pernah saling menyentuh. Selain itu, setiap fungsi memiliki bentuk $y_i\,(x) = p_i\,(x) / q_i\,(x)$, di mana

$$p_i\,(x) = \sum_{k=0}^{K} P_{i,k} x^k \quad \ldots (2.5) \qquad q_i\,(x) = \sum_{k=0}^{K} Q_{i,k} x^k \qquad (2.6)$$

Untuk i = 1; 2 dan bilangan bulat K. Gambar di bawah ini menunjukkan profil dari dua bidang tanah dalam cara ACM mewakili mereka. Tambalan di sebelah kiri memiliki lebar W = 6 dan kedalaman D = 9, sedangkan tambalan di sebelah kanan memiliki W = 8 dan D = 10. Batas-batas lapisan setiap tambalan dijelaskan oleh fungsi-fungsi yang ditentukan di bawahnya.



$$y_1\,(x) = \frac{10+1x}{2+0x} \qquad y_1\,(x) = \frac{-1392+864x-216x^2+24x^3-1x^4}{1312-864x+216x^2-24x^3+1x^4}$$

$$y_2\,(x) = \frac{-16+1x}{2+0x} \qquad y_2\,(x) = \frac{-73+36x-54x^2+36x^3-9x^4}{17-4x+6x^2-4x^3+1x^4}$$

Gambar 2.8 Soal Environment Protection

ACM akan menggali segala sesuatu di sepetak tanah hingga kedalaman penggalian tertentu *d*, dan kemudian menjual semua mineral yang diperoleh untuk mendapatkan keuntungan. Namun, mineral di lapisan atas dan bawah tidak berharga, sehingga keuntungan dari keseluruhan operasi datang secara eksklusif dari mineral-mineral di lapisan tengah. Faktanya, laba sebanding dengan area A dari lapisan tengah dalam profil yang paling dalam d. Diberikan deskripsi sebidang tanah dan bilangan bulat A, Anda ingin mengetahui kedalaman penggalian d Anda harus memungkinkan ACM untuk menggali tambalan sehingga mereka mendapatkan area mineral dari lapisan tengah dalam profil tepat A. Pada gambar di atas, Anda dapat melihat jawaban untuk dua kasus uji dalam input sampel. Untuk tambalan di sebelah kiri, untuk mendapatkan area A = 14 kedalaman penggalian harus d = 4,00000, sedangkan untuk tambalan di sebelah kanan area A = 14 membutuhkan kedalaman penggalian d = 5,51389.

Input

Setiap *test case* dijelaskan menggunakan 5 baris. Baris pertama berisi empat bilangan bulat *W, D, A* dan *K*, di mana *W* adalah lebar bidang tanah yang ingin ditambang ACM ($1 \leq W \leq 8$), *D* adalah kedalaman ($1 \leq D \leq 10$), *A* adalah area lapisan tengah dalam profil yang harus diperoleh ACM ($1 \leq A \leq W \times D$), dan K memungkinkan definisi antarmuka $y_1(x)$ dan $y_2(x)$ seperti yang dijelaskan di atas ($0 \leq K \leq 8$). Masing-masing baris lainnya berisi bilangan bulat $K + 1$ antara -108 dan 108, inklusif. Baris kedua berisi koefisien $p_1(x)$ dari *P1*,0 ke *P1, K*. Baris ketiga berisi koefisien $q_1(x)$dari *Q1*,0 hingga *Q1, K*. Baris keempat berisi koefisien $p_2(x)$dari *P2*,0 ke *P2, K*. Baris kelima berisi koefisien $q_1(x)$ dari *Q2*,0 hingga *Q2, K*. Dalam setiap uji kasus *A* benar-benar kurang dari total luas lapisan tengah dalam profil, dan terdapat nilai tunggal *d* sedemikian sehingga kedalaman penggalian *d* menghasilkan area mineral dari lapisan tengah dalam profil tepat *A*. Selain itu, , $q_1(x)\ 6 = 0$, $q_2(x)\ 6 = 0$ dan $-D < y_2(x) < y_1(x) < 0$, untuk $0 \leq x \leq W$.

Keluaran

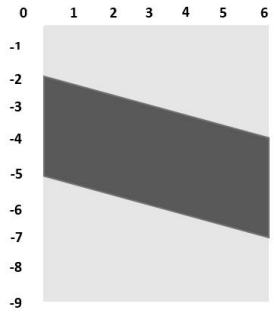
Untuk setiap test case menghasilkan garis dengan bilangan rasional yang mewakili kedalaman *d* bahwa ACM harus diizinkan untuk menggali sepetak tanah sehingga mereka mendapatkan area mineral dari lapisan tengah dalam profil tepat *A*. Hasilnya harus output sebagai bilangan rasional dengan tepat lima digit setelah titik desimal, dibulatkan jika perlu.

Sample Input	Sample Output
6 9 4 1 -10 1 2 0 -16 1 2 0 8 10 14 4 -1392 864 -216 24 -1 1312 -864 216 -24 1 -73 36 -54 36 -9 17 -4 6 -4 1	4.00000 5.51389

Gambar 2.9 Test Case Environment Protection

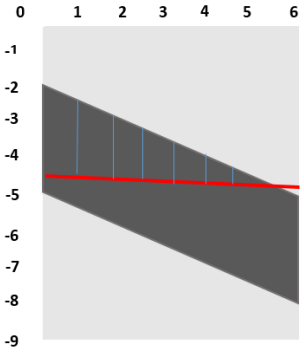
2.8 *Penyelesaian permasalahan Environment Protection dengan Binary Search*

Diberikan contoh permasalahan soal Environment Protection yaitu dua grafik yang diilustrasikan seusai Gambar 2.7



Gambar 2.10 Contoh Permasalahan Environment Protection

Langkah –langkah yang dilakukan untuk memperoleh kedalaman sesuai dengan luas yang diminta adalah



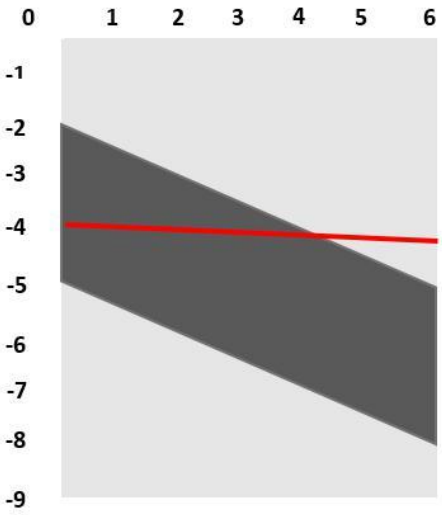
Gambar 2.11 Langkah penyelesaian permasalahan Environment Protection

Mencari titik tengah dengan membagi kedalaman menjadi 2 yaitu 4,5. Setelah itu mencari luas grafik yang dibatasi oleh garis dengan menggunakan algoritma Simpson 1/3 yang akan dibahas pada subbab 2.10. Jika luas kurang dari luas yang diinginkan maka kedalaman agar bergeser kebawah, jika luas lebih dari luas yang diinginkan.

Tabel 2.2 Iterasi *Binary Search*

Iterasi	Kedalaman
1	-4.50
2	-2.25
3	-3.37
4	-3.93
5	-4.21
6	-4.07
7	-4.00

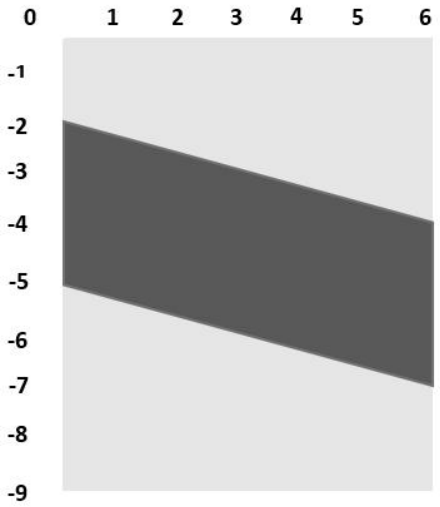
Maka akan dihasilkan kedalaman seperti gambar dibawah.



Gambar 2.12 Hasil Pencarian Biner

2.9 *Penyelesaian permasalahan Environment Protection Simpson 1/3*

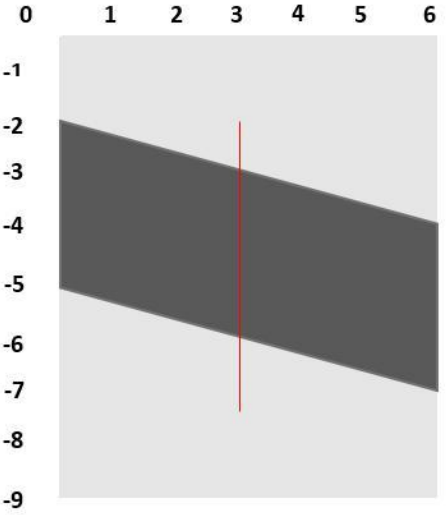
Diberikan contoh permasalahan soal Environment Protection yaitu dua grafik yang diilustrasikan sesuai gambar



$$y_1(x) = \frac{10 + 1x}{2 + 0x} \qquad y_2(x) = \frac{-16 + 1x}{2 + 0x}$$

Gambar 2.13 Ilustrasi Permasalahan Environment Protection

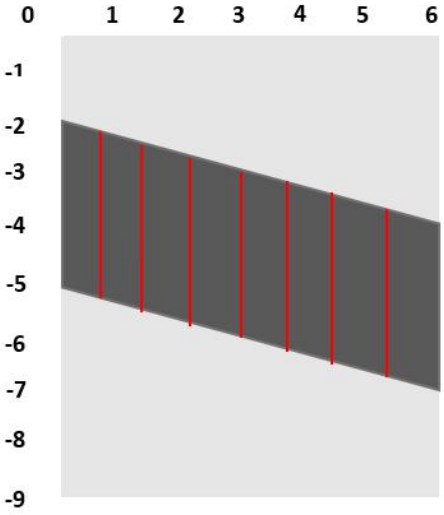
Langkah – langkah pencarian kedalaman dengan algoritma Simpson 1/3 dalam permasalahan adalah sebagai berikut.



Gambar 2.14 Langkah penyelesian algoritma Simpson 1/3

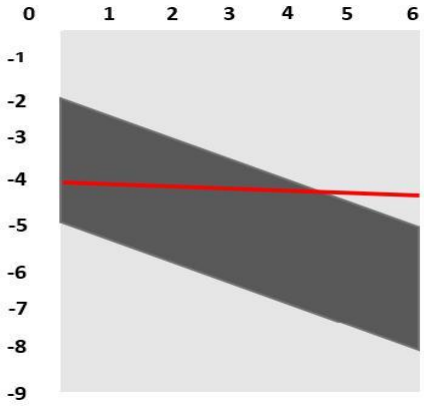
Menunjukkan proses partisi pada grafik yaitu dengan membaginya menjadi dua bagian yaitu 0 untuk titik bawah dan 3 untuk titik atas grafik partisi 1. 3 untuk batas bawah dan 6 untuk batas atas grafik partisi 2.

Melakukan partisi grafik *2n* sampai luas yang diinginkan tercapai.



Gambar 2.15 Partisi Grafik

Partisi terus berlanjut sampai menemukan luas yang diinginkan



Gambar 2.16 Hasil dari algoritma Simpson 1/3

[*Halaman ini sengaja dikosongkan*]

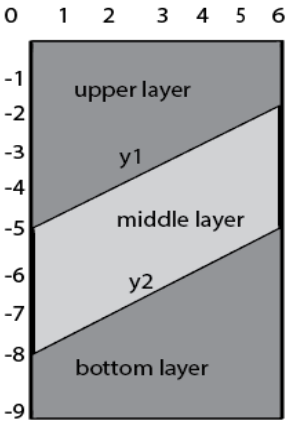
BAB III
DESAIN

Pada bagian ini akan dijelaskan analisis struktur data yang digunakan untuk menyelesaikan permasalahan pada Tugas Akhir ini.

3.1 Desain dan Analisis Penyelesaian Masalah

Pada permasalahan pencarian kedalam pada permasalahan *Environment Protection*, perusahaan diminta untuk memasukan fungsi dari grafik atas dan grafik yang bawah. Pendekatan yang dilakukan untuk menyelesaikan permasalahan tersebut, yaitu menggunakan *Simpson 1/3* dan *Binary Search*.

Diberikan contoh permasalahan soal Environment Protection yaitu dua grafik yang diilustrasikan seusai gambar

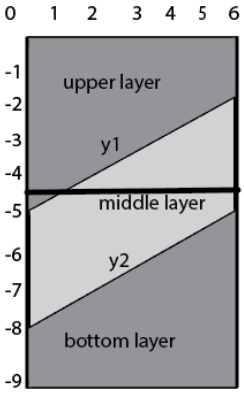


Gambar 3.1 Persoalan Environment Protection

$$y_1(x) = \frac{10 + 1x}{2 + 0x} \qquad y_2(x) = \frac{-16 + 1x}{2 + 0x}$$

Langkah – langkah untuk mencari kedalaman adalah sebagai berikut:

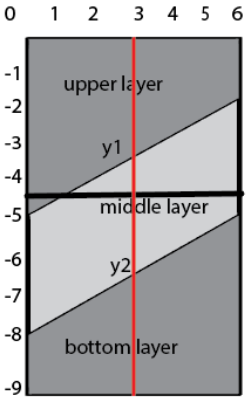
Untuk mencari kedalaman dari grafik agar luas yang dihasilkan sesuai diinginkan maka dilakukan pencarian biner (*binary search*). Berdasarkan Gambar 2.13, *high* adalah 0 atau batas atas dan *low* adalah -9 atau batas bawah. Pada pencarian biner, akan dicari titik tengah atau *mid*. Pada permasalahan tersebut, *mid* dapat dicari dengan rumus *mid = (high + low)*. Dengan demikian, mid adalah -4,5



Gambar 3.2 Daerah yang dibatasi D

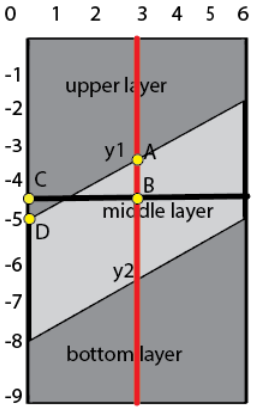
Setelah menentukan kedalam yang dilakuka dengan pencarian biner, selanjutnya mencari luas daerah yang dibatasi oleh *D* menggunakan algroritma *Simpson 1/3*.

Untuk memperkecil nilai error, perlu adanya partisi saat perhitungan *Simpson 1/3*.Pada kasus ini membagi daerah menjadi 2^n, dimana n adalah jumlah pembagian daerah yang dilakukan secara rekursif sampai nilai dari luas grafik < 1 e^{-10}.



Gambar 3.3 Partisi untuk mencari luas grafik

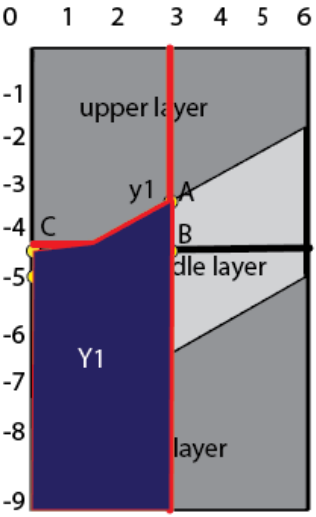
Tahap selanjutnya adalah penetuan titik yang akan digunakan untuk batas kiri dan batas kanan dari daerah yang akan dihitung luasnya.



Gambar 3.4 Titik potong pada grafik

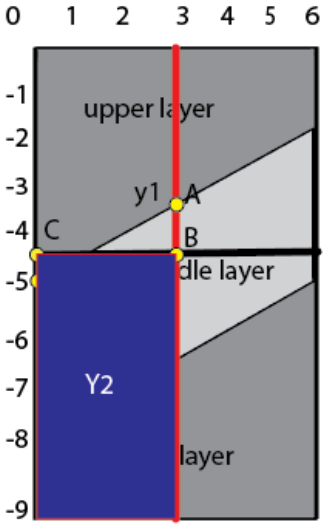
Pada Gambar 2.16 titik yang akan dipilih untuk dihitung luasnya dengan *Simpson 1/3* adalah titik yang memiliki nilai paling

besar. Pada *x* =3, terdapat dua titik, yaitu titik *A* -3,2 dan titik *B* -4,5. Maka yang akan dipilih adalah titik *A*. Begitu juga dengan titik *C* dan titik *D*. Maka daerah yang dicari luasnya adalah sebagai berikut.



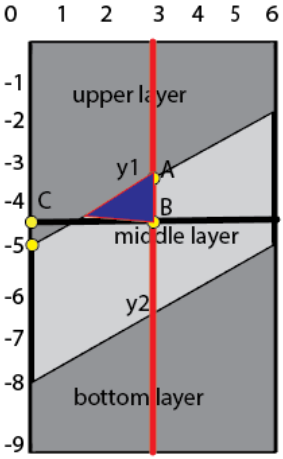
Gambar 3.5 Daerah y1

Gambar 2.17 adalah gambar daerah dibawah garis *y1* yang dibatasi oleh *D*. Untuk mencari daerah yang diarsir perlu digunakan algoritma *Simpson 1/3* dengan persamaan 2.3.
Dengan *b* = 3, *a* = 0, *f(x0)* = -4,5, *f(x1)* = -4.5, *f(x2)* = -3.2
Untuk mencari luas daerah diatas *D* maka perlu dicari luas daerah dibawah *D*



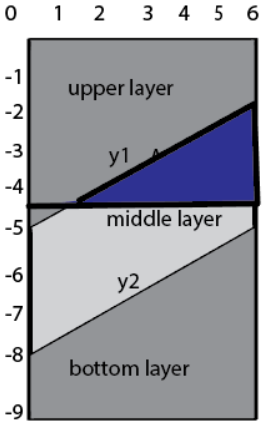
Gambar 3.6 Daerah dibawah D

Gambar 3.6 adalah luas daerah dibawah *D*. Setelah menentukan luas kedua daerah maka dapat ditentukan luas daerah *middle layer* yang dibatasi dengan *D*.



Gambar 3.7 Luas daerah diatas garis *D*

Setelah mengetahui luas daerah diatas grafik, maka dilakukan penggabungan semua partisi.



Gambar 3.8 Luas daerah diatas grafik setelah penggabungan

Setelah menemukan daerah yang dicari, maka perlu membandingan dengan daerah yang diinginkan. Jika luas kurang dari luas yang diinginkan maka kedalaman agar bergeser kebawah, jika luas lebih dari luas yang diinginkan, maka ke atas.

Tabel 3.1 Tabel iterasi Binary Search

Iterasi	Kedalaman
1	-4.50
2	-2.25
3	-3.37
4	-3.93
5	-4.21
6	-4.07
7	-4.00

3.2 Deskripsi Umum Sistem

Pada subbab ini akan dijelaskan tentang gambaran secara umum dari sistem yang akan dibangun. Pertama-tama sistem akan menerima masukan empat bilangan bulat *W*, *D*, *A* dan *K*, di mana *W* adalah lebar bidang tanah yang ingin ditambang ACM (*D* adalah kedalaman , *A* adalah area lapisan tengah dalam profil yang harus diperoleh ACM, dan *K* merepresentasikn derajat pangkat $y_1(x)$ dan $y_2(x)$ Kemudian sistem akan menyimpan fungsi grafik *y1(x)*, *y2(x)*,*q1(x)* dan *q(x)* dalam bentuk array.

Sebelum melanjutkan ke proses lainnya, akan dilakukan algoritma pencarian biner menggunakan pencarian biner (*binary search*) kedalam agar luas yang diperoleh sesuai dengan A.Setelah melakukan pencarian biner hitung luas daerah yang telah dibatasi oleh *d* menggunakan fungsi *simpson* dan *simpson_formula*, sebelum itu, lakukan pencarian titik *y* dengan menggunakan fungsi function yang akan dijelaskan pada subab selanjutnya.

1.	**Function** *main*()
4.	Input w ,h, area, k
5.	**for** *i = 0* **to** 4 **do**
6.	**for** *j = k* **to** 0 **do**
7.	*Input a[i][j]*
9.	**end**
11.	**end**
12.	*Solve*(-h, 0)
	Print *solve*
	end

Pseudocode 3.1 Fungsi Main

3.2.1 Fungsi Pencarian Biner

Untuk mencari kedalaman penggalian, maka diperlukan makan diperlukan algoritma untuk mencari kedalaman. Karena jumlah derajat pangkat pada permasalahan cukup tinggi $0 \leq K \leq 8$. Oleh karena itu, perlu adanya algoritma efisien yaitu *binary search.*

Input dari fungsi ini adalah batas bawah atau minus dari ketinggiannya *low* dan batas atas *low*. Pertama-tama siapkan bilangan bulat *low* untuk *-h* dan *high* untuk pertama adalah 0. Lalu tambahkan kedua bilangan dan bagi dua yang akan menjadi mid atau batas tengah. Lalu implementasikan fungsi *simpson* dengan menggunakan batas *mid*. Jika now lebih dari area yang dinginkan maka ulangi perulangan dengan mengganti *ans = mid* dan *low = mid*.

1.	**Function** *solve* (*low, high*)
2.	**Data:** *low* the lowest point, *high* the highest point
3.	**Result:** *answer*
4.	mid, ans = -1
5.	**while** *t* **do**
6.	*Mid = (low +high)/2*

7.	*Now = simpson (0, w, a[0], a[1], mid)*
8.	*- simpson (0, w, a[2], a[3], mid)*
9.	**If (**now > area) **do**
10.	ans = mid, low = mid
11.	**end**
12.	**else if** *now<=area* **do**
13.	high = mid
14.	**end**
15.	**end**
16.	**return** ans
17.	**end**

Pseudocode 3.2 Fungsi Solve(Binary Search)

3.2.2 Fungsi Menghitung Mempartisi Grafik

Untuk menghitung menghitung luas grafik, hal pertama yang akan dilakukan adalah mempartisi grafik menjadi beberapa bagian untuk meminimalkan *error true* yang terjadi. Pertama, mencari nilai tengah atau *m* kemudian menghitung pada fungsi *simpson_formula* akan dibahas pada subbab 3.3.3. Fungsi ini bersifat rekursif dan akan berhenti saat hasil partisinya kurang dari $1e^{-10}$.

1.	**Function** *simpson* (l, *r, p[], q[], y*)
2.	**Data:** *l* left point of graph, *right* right point of graph, *p[]* first function, *q[]* second function, *y* depth
3.	**Result:** *ans*
4.	m = (*l +r)/2*
5.	*ans = simpson_formula (l, r , p, q, y)*
6.	**if**(*ans – simpson_formula (l, m, p, q, y*)
7.	*– simpson_formula (m, r, p, q) < eps)* **do**
8.	**return** ans **end**
9.	**return** simpson (l, m, p, q, y) + simpson(m, r, p, q, y)
21.	**end**

Pseudocode 3.3 Fungsi Menghitung Partisi Grafik

3.2.3 Fungsi Simpson 1/3

Pada fungsi ini adalah implementasi dari *Simpson 1/3*. Masukan yang dibutuhkan adalah *l* batas kiri , *r* batas kanan, fungsi bagian atas *p[]*, fungsi bagian bawah *q[]*, dan kedalaman *y*. Pada fungsi berguna untuk menghitung luas grafik.

1.	**Function** *simpson_formula(l, r, p[], q[], y*)
2.	**Data:** *l* left point of graph, *right* right point of graph,
3.	*p[]* first function, *q[]* second function, *y* depth
4.	**Result:** *answer*
5.	*answer = function (l, p, q, y) + 4 * function ((l +r) / 2, p ,*
6.	*q, y) + function (r, p, q , y)) * (r – l) / 6*
7.	***return*** *answer*
8.	***end***

Pseudocode 3.4 Fungsi Mencari Luas grafik

3.2.4 Fungsi Mencari titik y

Untuk mencari luasan dengan *simpson 1/3*,perlu adanya fungsi untuk mencari titik yang akan digunakan dalam *simpson 1/3*. Pada fungsi ini memerlukan masukan berupa titik x, fungsi atas *p[]*, dan fungsi bagian bawah *q[]*, dan kedalaman d. Pertama, mendeklarasi jika up adalah 0 dan down adalah 0.Melakukan perhitungan perulangan untuk menghitungan nilai y pada grafik bagian atas dan grafik bagian bawah. Fungsi ini akan mengembalikan nilai yang paling maksimal dari kedalaman atau nilai y dari grafik dengan membagi *up* dan *down*.

1.	**Function** *function(x, , p[], q[], y*)
2.	**Data:** *x* point of graph, *p[]* first function,
3.	*q[]* second function, *y* depth
4.	**Result:** *y*
5.	*up: 0, down: 0*
6.	**for** *i = 0* **to** k **do**
7.	*up = up * x + p[i]*
8.	*down = down * x + q[i]*
	end
	y = max(y, up/down)
	return *y*

Pseudocode 3.5 Mencari titik y

BAB IV
IMPLEMENTASI

Pada bab ini akan dijelaskan tentang implementasi yang dilakukan berdasarkan algoritma yang telah dirancang pada bab sebelumnya.

4.1 Contoh Input dan Output

Berikut adalah contoh input dan output yang ada pada UVa Online Judge.

Tabel 4.1 Contoh Input dan Output pada UVa Online Judge

Contoh Input	Contoh Output
6 9 4 1 -10 1 2 0 -16 1 2 0	4.00000
8 10 14 4 -1392 864 -216 24 -1 1312 -864 216 -24 1 -73 36 -54 36 -9 17 -4 6 -4 1	5.51389

4.2 Lingkungan Implementasi

Lingkungan implementasi menggunakan sebuah komputer dengan spesifikasi perangkat lunak dan perangkat keras seperti tercantum pada Table 4.2.

Tabel 4.2 Spesifikasi Lingkungan Implementasi

No	Jenis Perangkat	Spesifikasi
1.	Perangkat Keras	• *Processor* Intel Core i7-8550U CPU @1.80GHz • *Memory* 8192 RAM
2.	Perangkat Lunak	• Sistem operasi Windows 10 Pro • *Intergrated Development Environment* Dev-C++ 5.3.0.3

4.3 Pendefinisian *Preprocessor Directivers*

Pada bagian ini, akan dijelaskan beberapa pendefinisian dari *preprocessor directives* yang akan digunakan dalam program ini. Penggunaan preprocessor directive ditujukan untuk mempermudah dan mempersingkat implementasi. Implementasi dari preprocessor directive ditunjukkan pada Kode Sumber 4.1.

```
#include <iostream>
#include <cstdio>
#include <cmath>
using namespace std;
const double eps = 1e-10;
double w , h , area , a[4][10];
```

Kode Sumber 4.2 Implementasi Preprocessor Directive

4.4 Fungsi Main

Fungsi main adalah implementasi algoritma yang dirancang pada Pseudocode 4.1. Implementasi fungsi main dapat dilihat pada Kode Sumber 4.2.

```
1.  int main() {
2.   while (scanf ("%lf %lf %lf %d" ,&w,&h,&area,
                    &k) != EOF) {
3.         for (int i = 0 ; i < 4 ; i ++)
4.             for (int j = k ; j >= 0 ; j --) {
5.                 scanf ("%lf" , &a[i][j]);
6.             }
7.         printf ("%.5f\n" , -solve (-h , 0));
8.     }
9.     return 0; }
```

Kode Sumber 4.3 Implementasi Fungsi Main

4.5 Fungsi Pencarian Biner

Fungsi pembangkitan query adalah implementasi algoritma yang dirancang pada Pseudocode 4.2. Implementasi fungsi pencarian biner dapat dilihat pada Kode Sumber 4.3.

```
1. double solve (double low , double high) {
2.     double mid , ans = -1;
3.     int t = 50;
4.     while (t --) {
5.         mid = (low + high) / 2.0;
6.         double now = simpson (0 , w , a[0],
            a[1] , mid) - simpson (0 , w ,
            a[2], a[3] , mid);
7.         if (now > area) ans = mid ,
           low = mid;
8.         else high = mid;
9.     }
10.     return ans;
11. }
```

Kode Sumber 4.4 Implementasi Fungsi Pencarian Biner

4.6 Fungsi Menghitung Partisi Grafik

Fungsi menghitung partisi grafik adalah implementasi algoritma yang dirancang pada Pseudocode 3.3. Implementasi fungsi pembangkitan query dapat dilihat pada Kode Sumber 4.4.

```
1. double simpson (double l , double r ,
          double *p , double *q , double y) {
2.      double m = (l + r) / 2.0;
3.      double ans = simpson_formula (l , r , p ,
                     q , y);
4.      if (fabs (ans - simpson_formula (l , m ,
            p , q , y) - simpson_formula (m , r ,
             p , q , y)) < eps) return ans;
5.      return simpson (l , m , p , q , y) +
               simpson (m , r , p , q , y);
6. }
```

Kode Sumber 4.5 Fungsi Menghitung Partisi Grafik

4.7 Implementasi Algoritma Pencarian Luas Grafik dengan Simpson 1/3

Fungsi luas grafik adalah implementasi algoritma yang dirancang pada Pseudocode 4.4 .Implementasi fungsi mencari luas grafik dapat dilihat pada Kode Sumber 4.5.

```
1. double simpson_formula (double l ,
           double r , double *p , double *q,
           double y) {

2.   return (function (l , p , q , y) + 4 *
              function ((l + r) / 2.0 , p ,
              q , y) + function (r , p , q,
              y)) * (r - l) / 6.0;
3. }
```

Kode Sumber 4.6 Implementasi Fungsi Simpson 1/3

4.8 Fungsi Mencari titik y

Fungsi main adalah implementasi algoritma yang dirancang pada Pseducode 3.5. Implementasi fungsi main dapat dilihat pada Kode Sumber 4.6.

```
1. double function (double x , double *p ,
2.                  double*q , double y) {
3.
4.      double up = 0 , down = 0;
5.      for (int i = 0 ; i <= k ; i ++) {
6.          up = up * x + p[i];
7.          down = down * x + q[i];
8.      }
9.      return max (y , up / down);
10.  }
```

Kode Sumber 4.7 Implementasi Fungsi Main

BAB V
UJI COBA DAN EVALUASI

Pada bab ini akan dijelaskan tentang uji coba dan evaluasi dari implementasi sistem yang telah dilakukan pada Bab 4.

5.1 Lingkungan Uji Coba

Lingkungan uji coba menggunakan sebuah komputer dengan spesifikasi perangkat lunak dan perangkat keras seperti tercantum pada Tabel 5.1.

Tabel 5.1 Spesifikasi Lingkungan Uji Coba

No	Jenis Perangkat	Spesifikasi
1.	Perangkat Keras	• *Processor* Intel Core i7-8550U CPU @1.80GHz • *Memory* 8192 RAM
2.	Perangkat Lunak	• Sistem operasi Windows 10 Pro • *Intergrated Development Environment* Dev-C++ 5.3.0.3

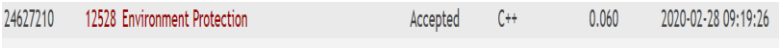
5.2 Skenario Uji Coba

Subbab ini akan menjelaskan hasil pengujian program untuk menyelesaikan permasalahan Environment Protection. Metode pengujian yang dilakukan adalah sebagai berikut.

1. Uji Coba Kebenaran. Pengujian ini menggunakan Online Judge untuk menguji kebenaran dan kinerja program.
2. Uji Coba Kinerja. Pengujian ini dulakukan untuk mengukur kinerja program dengan cara *submit* kode program ke situs *UVa Online Judge* sebanyak 4 kali.
3. Uji Coba Lokal. Pengujian ini menggunakan beberapa test case pada Uva Online Judge.

5.3 Uji Coba Kebenaran

Pada subbab ini akan dibahas mengenai uji coba kebenaran yang dilakukan dengan mengirim kode sumber terakit ke dalam situs penilaian daring UVa Online Judge. Bukti hasil pengujian dapat dilihat pada Gambar 5.1.



Gambar 5.1 Hasil Uji Coba Kebenaran Situs Penilaian UVa Online Judge

Dari hasil uji coba yang telah dilakukan, kode sumber program mendapat umpan balik *Accepted*. Waktu yang dibutuhkan program adalah 0.06 detik. Hasil uji coba diatas membuktikan bahwa implementasi yang dilakukan telah berhasil menyelesaikan permasalahan Environmnet Protection dengan batasan-batasan yang telah ditetapkan.

5.4 Uji Coba Kinerja

Pada subbab ini akan ditampilkan hasil uji coba kinerja yang dilakukan dengan mengirim kode sumber terakit ke dalam situs penilaian daring UVa Online Judge dapat dilihat pada Gambar 5.2. Rata-rata hasil pengumpulan kode berkas dengan table pencarian adalah 0.07 detik. Hasil Uji Coba Waktu dapat dilihat pada Gambar 5.3.

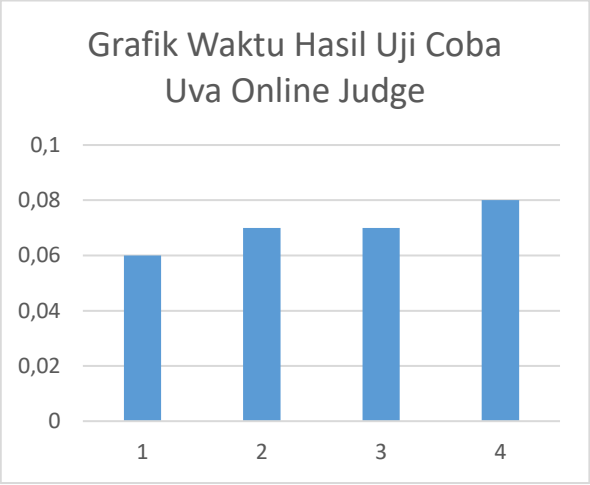
Gambar 5.2 Hasil Uji Coba Kebenaran Situs Penilaian UVa Online Judge

5.5 Uji Coba Lokal

Input	Output	Hasil
6 10 15 8 -3131 2500 -750 100 - 2510 2000 -600 80 -4 626 -500 150 -20 627 - 500 150 -20 1 -11053 35540 -50658 41420 -21142 6880 -1392 160 -8 1394 -4460 6342 -5180 2643 -860 174 -20 1	7.32349	7.32349
7 10 18 8 -412 540 -270 60 -335 432 -216 48 -4 82 -108 54 -12 83 -108 54 -12 1 -8116000 11891412 - 7617078 2785572 -636027 92840 -8460 440 -10 811922 -1189364 761766 - 278564 63603 -9284 846 - 44 1	7.37047	7.37047
6 10 14 8 -42487 97860 -98946 57204 -20648 4760 -684 56 -2 21074 -48748 49398 - 28588 10323 -2380 342 - 28 1 -185 756 -1638 2220 - 1936 1080 -372 72 -6	5.41577	5.41577

34 -132 278 -372 323 -180 62 -12 1		
6 10 13 8 -730 2596 -5082 6052 -4435 2000 -540 80 -5 164 -544 1032 -1216 888 -400 108 -16 1 -2312 2304 -864 144 -2578 2560 -960 160 -10 257 -256 96 -16 258 -256 96 -16 1	5.37291	5.37291
6 10 12 8 -86865 223840 -249528 153688 -56709 12800 -1728 128 -4 22049 -56192 62448 -38432 14178 -3200 432 -32 1 -10640 31032 -54348 55272 -32470 11016 -2124 216 -9 1252 -3504 6056 -6144 3608 -1224 236 -24 1	5.26767	5.26767
6 10 13 8 -84271 222112 -249096 153640 -56707 12800 -1728 128 -4 22049 -56192 62448 -38432 14178 -3200 432 -32 1 -1798 1792 -672 112 -2062 2048 -768 128 -8 257 -256 96 -16 258 -256 96 -16 1	4.70912	4.70912

8 10 14 8 -576 1744 -3156 3664 - 2666 1200 -324 48 -3 164 -544 1032 -1216 888 -400 108 -16 1 -11675 7776 -1944 216 - 9090 6048 -1512 168 -7 1297 -864 216 -24 1298 - 864 216 -24 1	5.27290	5.27290



Gambar 5.2 Grafik Waktu Uji Coba UVa Online Judge

BAB VI
KESIMPULAN DAN SARAN

Pada bab ini akan dijelaskan kesimpulan dari hasil uji coba yang telah dilakukan serta saran-saran tentang pengembangan yang dapat dilakukan terhadap Tugas Akhir ini di masa yang akan datang.

6.1 Kesimpulan

Berdasarkan penjabaran di bab-bab sebelumnya, dapat disimpulkan beberapa poin terkait penyelesaian permasalahan Environment Protection.

1. Permasalahan Environment Protection dapat diselesaikan dengan algoritma Simpon 1/3 dan Binary Search.
2. Dari hasil uji coba dengan cara mengirimkan kode program ke situs UVa, didapatkan rata-rata waktu dengan algoritma Simpson 1/3 adalah 0.06 detik.
3. Solusi menggunakan algoritma Simpson 1/3 dan Binary Search dapat diterima pada pengujian online UVa Online Judge.
4. Solusi menggunakan algoritma Simpson 1/3 dan Binary Search merupakan solusi yang efektif.

6.2 Saran

Saran untuk penelitian terkait adalah bagaimana penggunaan algoritma lain selain dengan Simpson 1/3 agar hasil yang dihasilkan lebih maksimal.

[*Halaman ini sengaja dikosongkan*]

DAFTAR PUSTAKA

[1] UVa Online Judge (Environment Protection) https://onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=441&page=show_problem&problem=3973

[2] Chapra. C. Steven, *Numerical Methods for Engineers Seventh Edition. 2*015, ISBN: 978–0–07–339792–4

[*Halaman ini sengaja dikosongkan*]

BIODATA PENULIS

Penulis bernama Magdalena Andiniwarih Ismanu, putri pertama dari dua bersaudara yang lahir pada tanggal 9 September 1998 di Jombang. Penulis telah mengenyam pendidikan di Sekolah Dasar Negeri Candimulyo 2004 hingga 2010, Sekolah Menengah Pertama 1 Jombang pada tahun 2010 hingga 2013, dan Sekolah Menengah Atas 2 Jombang pada tahun 2013 hingga 2016. Pada masa penulisan, penulis sedang menempuh masa studi S1 di Institut Teknologi Sepuluh Nopember, Surabaya di Departemen Informatika.

Selama masa studi, penulis memiliki ketertarikan yang dalam mengenai rancang bangun aplikasi sistem informasi. Keinginan penulis dalam mengajar juga mendorong penulis menjadi asisten dosen pada mata kuliah Komputasi Numerik, dan Probabilitas Statistik.

Di luar kesibukan akademik, penulis juga berkontribusi dalam berbagai kepanitiaan dan organisasi dalam skala kampus. Kepanitian yang penulis ikuti adalah Schematics (2017 dan 2018), ITS EXPO 2017 dan FTIF Festival (2017). Organisasi yang penulis ikuti adalah Persatuan Mahasiswa Kristen ITS dan Himpunan Mahasiswa Teknik Computer (HMTC) ITS.