



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - IF184802

IMPLEMENTASI REDUKSI POLYGON DALAM ME- NYELESAIKAN PERMASALAHAN RELATIVE CON- VEX HULL DENGAN STUDI KASUS SPHERE ONLINE JUDGE 5637 LL AND ERBAO

MICHAEL JULIAN ALBERTUS
NRP 05111640000097

Dosen Pembimbing 1
Rully Soelaiman, S.Kom., M.Kom.

Dosen Pembimbing 2
Yudhi Purwananto, S.Kom., M.Kom.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya, 2019

[Halaman ini sengaja dikosongkan]



TUGAS AKHIR - IF184802

**IMPLEMENTASI REDUKSI POLYGON DALAM ME-
NYELESAIKAN PERMASALAHAN RELATIVE CON-
VEX HULL DENGAN STUDI KASUS SPHERE ONLINE
JUDGE 5637 LL AND ERBAO**

MICHAEL JULIAN ALBERTUS
NRP 05111640000097

Dosen Pembimbing 1
Rully Soelaiman, S.Kom., M.Kom.

Dosen Pembimbing 2
Yudhi Purwananto, S.Kom., M.Kom.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya, 2019

[Halaman ini sengaja dikosongkan]



UNDERGRADUATE THESES - IF184802

**IMPLEMENTATION OF POLYGON REDUCTION FOR
SOLVING RELATIVE CONVEX HULL PROBLEM WITH
CASE STUDY SPHERE ONLINE JUDGE 5637 LL AND
ERBAO**

MICHAEL JULIAN ALBERTUS
NRP 05111640000097

Supervisor 1
Rully Soelaiman, S.Kom., M.Kom.

Supervisor 2
Yudhi Purwananto, S.Kom., M.Kom.

INFORMATICS DEPARTMENT
Faculty of Information Technology and Communication
Institut Teknologi Sepuluh Nopember
Surabaya, 2019

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

IMPLEMENTASI REDUKSI POLYGON DALAM MENYELESAIKAN PERMASALAHAN RELATIVE CONVEX HULL DENGAN STUDI KASUS SPHERE ONLINE JUDGE 5637 LL AND ERBAO

TUGAS AKHIR

Diajukan Guna Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Algoritma Pemrograman
Program Studi S-1 Departemen Informatika
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

Michael Julian Albertus
NRP. 05111640000097

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)

Surabaya
9 November 2019

[Halaman ini sengaja dikosongkan]

ABSTRAK

IMPLEMENTASI REDUKSI POLYGON DALAM MENYELESAIKAN PERMASALAHAN RELATIVE CONVEX HULL DENGAN STUDI KASUS SPHERE ONLINE JUDGE 5637 LL AND ERBAO

Nama : Michael Julian Albertus
NRP : 05111640000097
Departemen : Departemen Informatika,
Fakultas Teknologi Informasi dan
Komunikasi, ITS
Pembimbing I : Rully Soelaiman, S.Kom., M.Kom.
Pembimbing II : Yudhi Purwananto, S.Kom.,
M.Kom.

Abstrak

Computational geometry adalah cabang dari ilmu komputer yang dikhususkan untuk mempelajari algoritma yang dapat dinyatakan dalam suatu geometri. Salah satu algoritma yang sering dipakai pada computational geometry adalah algoritma convex hull. Convex hull adalah sebuah set polygon dari titik pada bidang euclidean atau ruang euclidean, atau dapat disebut himpunan cembung terkecil yang berisi titik. Convex hull dapat divisualisasikan sebagai bentuk yang tertutup oleh karet gelang yang membentang di sekitar titik - titik tersebut.

Relative convex hull merupakan penurunan dari convex hull. Relative convex hull merupakan convex hull yang mempunyai cavity (cekungan ke dalam) yang diakibatkan atau relatif terhadap sesuatu yang membatasi convex hull tersebut.

Topik Tugas Akhir ini mengulas algoritma reduksi poligon untuk menyelesaikan permasalahan relative convex hull. Melalui pengujian dan studi kasus didapatkan bahwa algoritma reduksi poligon dapat menyelesaikan permasalahan relative convex hull dengan efisien.

Kata Kunci: geometri; convex hull; melkman algorithm; relative poligon;

ABSTRACT

IMPLEMENTATION OF POLYGON REDUCTION FOR SOLVING RELATIVE CONVEX HULL PROBLEM WITH CASE STUDY SPHERE ONLINE JUDGE 5637 LL AND ERBAO

Name : Michael Julian Albertus
Student ID : 05111640000097
Department : Informatics Department,
Faculty of Information Technology
and Communication, ITS
Supervisor I : Rully Soelaiman, S.Kom., M.Kom.
Supervisor II : Yudhi Purwananto, S.Kom.,
M.Kom.

Abstract

computational geometry is one of the computer science branches that mainly focus in studying geometrical algorithm. One of the algorithm that mostly used is convex hull. Convex hull is a polygon from multiple point inside of euclidean plane. In short, minimum convex polygon that covers set of points. convex hull can be visualized by a rubber band that covers the set of points.

Relative convex hull derived from convex hull. Relative convex hull is convex hull that have one or more cavity that relative to some thing that limit the convex polygon.

In this Thesis will review polygon reduction algorithm to solve relative convex hull problem. According to sub-sequence testing and case study, it appears that polygon reduction algorithm

can solve relative convex hull problem efficiently.

Keywords: geometry; convex hull; melkman algorithm; relative polygon;

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa. Atas rahmat dan kasih sayangNya, penulis dapat menyelesaikan tugas akhir dan laporan akhir dalam bentuk buku ini.

Pengerjaan buku ini penulis tujukan untuk mengeksplorasi lebih mendalam topik-topik yang tidak diwadahi oleh kampus, namun banyak menarik perhatian penulis. Selain itu besar harapan penulis bahwa pengerjaan tugas akhir sekaligus pengerjaan buku ini dapat menjadi batu loncatan penulis dalam menimba ilmu yang bermanfaat.

Penulis ingin menyampaikan rasa terima kasih kepada banyak pihak yang telah membimbing, menemani dan membantu penulis selama masa pengerjaan tugas akhir maupun masa studi.

1. Ayah, Ibu, dan keluarga penulis yang selalu memberikan dukungan, perhatian, dan kasih sayang bagi penulis yang menjadi semangat selama perkuliahan maupun pengerjaan Tugas Akhir.
2. Bapak Rully Soelaiman S.Kom.,M.Kom., selaku pembimbing I penulis. Ucapan terima kasih juga penulis sampaikan atas segala perhatian, didikan, pengajaran, dan nasihat yang telah diberikan oleh beliau selama masa studi penulis.
3. Bapak Yudhi Purwananto, S.Kom., M.Kom., selaku dosen pembimbing II yang telah memberikan petunjuk dan saran-saran yang berharga dalam penulisan Tugas Akhir ini.
4. Semua pihak Departemen Informatika Institut Teknologi Sepuluh Nopember yang telah mendukung penulisan Tugas Akhir ini.
5. Serta teman-teman lainnya yang tidak bisa saya sebutkan satu-persatu yang banyak membantu dalam penulisan Tugas Akhir ini hingga selesai.

Penulis menyadari bahwa buku ini jauh dari kata sempurna. Maka dari itu, penulis memohon maaf apabila terdapat salah kata maupun makna pada buku ini. Akhir kata, penulis mempersembahkan buku ini sebagai wujud nyata kontribusi penulis dalam ilmu pengetahuan.

Surabaya, 9 November 2019

Michael Julian Albertus

DAFTAR ISI

LEMBAR PENGESAHAN	vii
ABSTRAK	ix
ABSTRACT	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xix
DAFTAR TABEL	xxi
DAFTAR PSEUDOCODE	xxiii
DAFTAR KODE SUMBER	xxv
DAFTAR NOTASI	xxvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan	3
1.5 Manfaat	4
1.6 Metodologi	4
1.7 Sistematika Penulisan	5
BAB II DASAR TEORI	7
2.1 Deskripsi Permasalahan	7
2.2 Convex Polygon	8
2.3 Relative Convex Polygon	8
2.4 Strategi Penyelesaian Permasalahan	9
2.4.1 Pemrosesan Titik Pembentuk Polygon yang Membentuk Convex	10

2.4.2	Convex Hull dari Titik yang Berada di Dalam Polygon	11
2.5	Convex Hull	11
2.5.1	Relative Convex Hull	12
2.5.2	Algoritma Convex Hull	13
2.6	Point Inside Polygon	18
BAB III	DESAIN	19
3.1	Desain Umum Sistem	19
3.2	Desain Fungsi Main	19
3.3	Desain Class Point	19
3.4	Desain Class Vec	21
3.5	Desain Class Line	22
3.6	Desain Class Segment	23
3.7	Desain Class Polygon	24
3.8	Fungsi BetweenD	27
3.9	Fungsi EDist	28
3.10	Fungsi Cross	29
3.11	Fungsi Orientation	29
3.12	Fungsi OnSegment	30
3.13	Fungsi ConvexHull	30
3.14	Fungsi InSimplePolygon	31
3.15	Fungsi GetBetween	33
3.16	Fungsi Solve	36
BAB IV	IMPLEMENTASI	39
4.1	Lingkungan implementasi	39
4.2	Implementasi Program Utama	39
4.2.1	Header yang diperlukan	39
4.2.2	Preprocessor	40
4.2.3	Variabel Global	41

4.2.4	Implementasi Fungsi Main	41
4.2.5	Implementasi Class Point	42
4.2.6	Implementasi Class Vec	43
4.2.7	Implementasi Class Line	44
4.2.8	Implementasi Class Segment	46
4.2.9	Implementasi Class Polygon	46
4.2.10	Implementasi Fungsi BetweenD	47
4.2.11	Implementasi Fungsi EDist	48
4.2.12	Implementasi Fungsi Cross	48
4.2.13	Implementasi Fungsi Orientation	48
4.2.14	Implementasi Fungsi OnSegment	49
4.2.15	Implementasi Fungsi ConvexHull	49
4.2.16	Implementasi Fungsi InSimplePolygon	50
4.2.17	Implementasi Fungsi GetBetween	50
4.2.18	Implementasi Fungsi Solve	52
BAB V	UJI COBA DAN EVALUASI	55
5.1	Lingkungan Uji Coba	55
5.2	Skenario Uji Coba	56
5.3	Uji Coba Kebenaran	56
5.4	Uji Coba Kinerja Lokal	57
5.5	Evaluasi Kebenaran Uji Coba Lokal	57
5.6	Uji Coba Kinerja Luar	65
BAB VI	KESIMPULAN	67
6.1	Kesimpulan	67
6.2	Saran	67
	DAFTAR PUSTAKA	69
	LAMPIRAN A: DATA UJI	71
	LAMPIRAN B: Hasil Uji Coba Reduksi Polygon pada Si-	
	tus SPOJ Sebanyak 10 Kali	101

BIODATA PENULIS 103

DAFTAR GAMBAR

Gambar 1.1	Ilustrasi Convex Hull	1
Gambar 1.2	Ilustrasi Relative Convex Hull	2
Gambar 2.1	Ilustrasi Contoh Kasus Tanpa Solusi	7
Gambar 2.2	Ilustrasi Contoh Kasus	8
Gambar 2.3	Ilustrasi Properti Convex Polygon 1	9
Gambar 2.4	Ilustrasi Properti Convex Polygon 2	9
Gambar 2.5	Ilustrasi Relative Convex Polygon	9
Gambar 2.6	Ilustrasi Convex Cull	12
Gambar 2.7	Ilustrasi Relative Convex Hull	12
Gambar 2.8	Ilustrasi Algoritma Melkman	14
Gambar 2.9	Ilustrasi Algoritma Monotone Chain	16
Gambar 2.10	Ilustrasi Algoritma Point Inside Polygon . . .	18
Gambar 5.1	Hasil Uji Coba Kebenaran Situs Penilaian Sphere Online Judge	56
Gambar 5.2	Grafik Mean Running Time Kasus Uji	58
Gambar 5.3	Ilustrasi Kondisi Awal	58
Gambar 5.4	Ilustrasi Iterasi 1	59
Gambar 5.5	Ilustrasi Iterasi 2	60
Gambar 5.6	Ilustrasi Iterasi 3	61
Gambar 5.7	Ilustrasi Iterasi 4	62
Gambar 5.8	Ilustrasi Iterasi 5	62
Gambar 5.9	Ilustrasi Iterasi 6	63
Gambar 5.10	Ilustrasi Iterasi 7	64
Gambar 5.11	Ilustrasi Iterasi 8	64

Gambar 5.12 Grafik Waktu Uji Coba 10 Kali pada Situs SPOJ 65

Gambar 5.13 Grafik Memori Uji Coba 10 Kali pada Situs
SPOJ 66

Gambar C.1 Hasil Pengumpulan Kode Program Utama
Dengan Algoritma reduksi polygon 101

DAFTAR TABEL

Tabel 2.1	Tabel Perbandingan Algoritma Convex Hull . .	13
Tabel 3.1	Nama dan Fungsi Variabel dalam Class POINT .	21
Tabel 3.2	Nama dan Fungsi Variabel dalam Class VEC . .	21
Tabel 3.3	Nama dan Fungsi Variabel dalam class LINE . .	22
Tabel 3.4	Nama dan Fungsi Variabel dalam Class SEGMENT	23
Tabel 3.5	Nama dan Fungsi Variabel dalam Class POLYGON	24
Tabel 3.6	Masukan, Proses, dan Keluaran dari Fungsi NEXT Class POLYGON	25
Tabel 3.7	Masukan, Proses, dan Keluaran dari Fungsi PREV Class POLYGON	26
Tabel 3.8	Masukan, Proses, dan Keluaran dari Fungsi PERIMETER Class POLYGON	27
Tabel 3.9	Masukan, Proses, dan Keluaran dari Fungsi BETWEEN	28
Tabel 3.10	Masukan, Proses, dan Keluaran dari Fungsi EDIST	28
Tabel 3.11	Masukan, Proses, dan Keluaran dari Fungsi CROSS	29
Tabel 3.12	Masukan, Proses, dan Keluaran dari Fungsi ORIENTATION	30
Tabel 3.13	Masukan, Proses, dan Keluaran dari Fungsi ONSEGMENT	31
Tabel 3.14	Masukan, Proses, dan Keluaran dari Fungsi CONVEXHULL	33
Tabel 3.15	Masukan, Proses, dan Keluaran dari Fungsi INSIMPLEPOLYGON	33

Tabel 3.16	Masukan, Proses, dan Keluaran dari Fungsi GETBETWEEN	36
Tabel 3.17	Masukan, Proses, dan Keluaran dari Fungsi SOLVE	38
Tabel 5.1	Tabel Data Uji Coba Kebenaran Lokal dengan Sampel Data	59

DAFTAR PSEUDOCODE

Pseudocode 2.1	Melkman Convex Hull	15
Pseudocode 2.2	Monotone Chain Algorithm	17
Pseudocode 3.1	Fungsi MAIN	20
Pseudocode 3.2	Class POINT	20
Pseudocode 3.3	Class VEC	22
Pseudocode 3.4	Class LINE	23
Pseudocode 3.5	Class SEGMENT	24
Pseudocode 3.6	Class POLYGON	25
Pseudocode 3.7	Fungsi NEXT pada class POLYGON . . .	26
Pseudocode 3.8	Fungsi PREV pada class POLYGON . . .	26
Pseudocode 3.9	Fungsi PERIMETER pada class POLYGON	27
Pseudocode 3.10	Fungsi BETWEEN	27
Pseudocode 3.11	Fungsi EDIST	29
Pseudocode 3.12	Fungsi CROSS	29
Pseudocode 3.13	Fungsi ORIENTATION	30
Pseudocode 3.14	Fungsi ONSEGMENT	31
Pseudocode 3.15	Fungsi CONVEXHULL	32
Pseudocode 3.16	Fungsi INSIMPLEPOLYGON	34
Pseudocode 3.17	Fungsi GETBETWEEN	35
Pseudocode 3.18	Fungsi SOLVE	37

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 4.1	<i>Header</i> yang Diperlukan	40
Kode Sumber 4.2	<i>Preprocessor</i> yang Diperlukan	41
Kode Sumber 4.3	Variabel Global yang Didefinisikan Untuk Program	41
Kode Sumber 4.4	Fungsi Main	42
Kode Sumber 4.5	Struct Point	43
Kode Sumber 4.6	Struct Vec	44
Kode Sumber 4.7	Struct Line	45
Kode Sumber 4.8	Struct Segment	46
Kode Sumber 4.9	Struct Polygon	47
Kode Sumber 4.10	Fungsi BetweenD	47
Kode Sumber 4.11	Fungsi EDist	48
Kode Sumber 4.12	Fungsi Cross	48
Kode Sumber 4.13	Fungsi Orientation	48
Kode Sumber 4.14	Fungsi OnSegment	49
Kode Sumber 4.15	Fungsi ConvexHull	49
Kode Sumber 4.16	Fungsi InSimplePolygon	50
Kode Sumber 4.17	Fungsi GetBetween	51
Kode Sumber 4.18	Fungsi Solve	52

[Halaman ini sengaja dikosongkan]

DAFTAR NOTASI

\sum	Notasi yang digunakan untuk menjumlahkan sejumlah bilangan terurut dengan aturan tertentu.
\forall	Notasi yang mewakili setiap element pada himpunan
$\langle \dots \rangle$	Notasi untuk menyatakan himpunan
$ \dots $	Notasi untuk menyatakan banyak anggota himpunan

[Halaman ini sengaja dikosongkan]

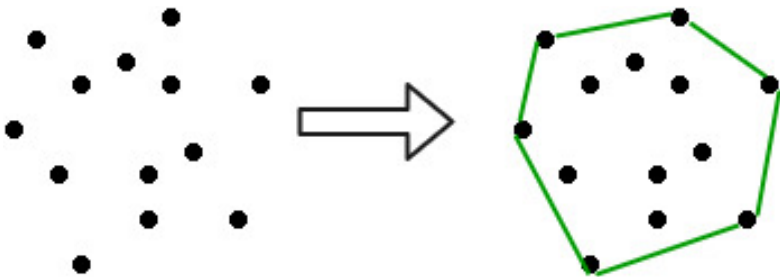
BAB I

PENDAHULUAN

Pada bab ini, akan dijelaskan mengenai latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi pengerjaan, dan sistematika penulisan Tugas Akhir.

1.1 Latar Belakang

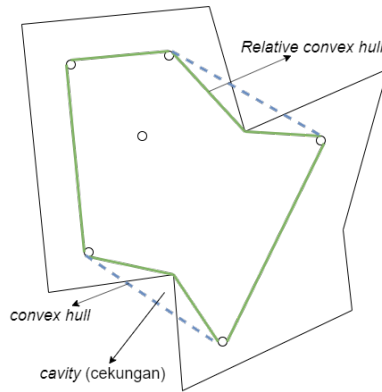
Computational geometry adalah cabang dari ilmu komputer yang dikhususkan untuk mempelajari algoritma yang dapat dinyatakan dalam suatu geometri. Salah satu algoritma yang sering dipakai pada *computational geometry* adalah algoritma *convex hull*. *Convex hull* adalah sebuah set polygon dari titik pada bidang *euclidean* atau ruang *euclidean*, atau dapat disebut himpunan cembung terkecil yang berisi titik. Sebagai contoh, ketika suatu kumpulan titik merupakan bagian yang dibatasi dalam sebuah bidang, *convex hull* dapat divisualisasikan sebagai bentuk yang tertutup oleh karet gelang yang membentang di sekitar titik - titik tersebut. Berikut merupakan contoh dari *convex hull* :



Gambar 1.1 Ilustrasi Convex Hull

Relative convex hull merupakan penurunan dari *convex hull*. *Relative convex hull* merupakan *convex hull* yang mempunyai

cavity (cekungan ke dalam) yang diakibatkan atau relatif terhadap sesuatu yang membatasi *convex hull* tersebut. Ilustrasi *relative convex hull* dapat dilihat pada gambar 2.7.



Gambar 1.2 Ilustrasi Relative Convex Hull

Pada topik Tugas Akhir ini akan dijelaskan algoritma penyelesaian untuk mencari *relative convex hull* dari sekumpulan titik yang berada di dalam sebuah polygon sederhana dengan menggunakan reduksi polygon pada studi kasus pada Sphere Online Judge 5637 LL and ErBao.

1.2 Rumusan Masalah

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

1. Bagaimana mencari *relative convex hull* dari kumpulan titik di dalam sebuah polygon?
2. Bagaimana reduksi polygon menyelesaikan masalah *relative convex hull* dari kumpulan titik?

1.3 Batasan Masalah

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

1. Implementasi reduksi polygon sebagai penyelesaian permasalahan *relative convex hull* pada soal ISUN1.
2. Algoritma *relative convex hull* terbatas pada analisis intuitif yang logis.

Berikut merupakan batasan pada situs Sphere Online Judge:

1. Implementasi dilakukan menggunakan bahasa pemrograman C++.
2. Banyaknya sisi pada polygon pembatas (n) diantara 3 sampai 500.
3. Banyaknya pohon yang berada dalam taman (m) diantara 0 sampai 500.
4. Batas maksimum untuk tiap vertex memenuhi (x, y) dimana nilai $|x|, |y| \leq 10000$.
5. Banyak soal tidak diketahui karena program berhenti sampai EOF.
6. Batas waktu yang diberikan adalah 0.142 detik.
7. Batas memori yang diberikan adalah 1.536 MB.
8. Batas kode sumber yang diberikan adalah 50.000 B.

1.4 Tujuan

Tujuan Tugas Akhir ini adalah sebagai berikut :

1. Mengevaluasi kinerja reduksi polygon untuk menyelesaikan permasalahan komputasi *relative convex hull* pada LL and Er-Bao.

1.5 Manfaat

Tugas Akhir ini mampu memberikan pemahaman algoritma yang tepat untuk menyelesaikan permasalahan komputasi *relative convex hull* dengan efisien.

1.6 Metodologi

Metodologi pengerjaan yang digunakan pada Tugas Akhir ini memiliki beberapa tahapan. Tahapan-tahapan tersebut yaitu :

1. Penyusunan proposal
Pada tahapan ini penulis memberikan penjelasan mengenai apa yang penulis akan lakukan dan mengapa Tugas Akhir ini dilakukan. Penjelasan tersebut dituliskan dalam bentuk proposal Tugas Akhir.
2. Studi literatur
Pada tahapan ini penulis mengumpulkan referensi yang diperlukan guna mendukung pengerjaan Tugas Akhir. Referensi yang digunakan dapat berupa hasil penelitian yang sudah pernah dilakukan, buku, artikel internet, atau sumber lain yang bisa dipertanggungjawabkan.
3. Implementasi algoritma
Pada tahapan ini penulis mulai mengembangkan algoritma yang digunakan untuk menyelesaikan permasalahan komputasi *relative convex hull*.
4. Pengujian dan evaluasi
Pada tahapan ini penulis menguji performa algoritma yang digunakan. Hasil pengujian kemudian dievaluasi untuk kemudian dipertimbangkan apakah algoritma masih bisa ditingkatkan lagi atau tidak.
5. Penyusunan buku
Pada tahapan ini penulis menyusun hasil pengerjaan Tugas Akhir mengikuti format penulisan Tugas Akhir.

1.7 Sistematika Penulisan

Sistematika laporan Tugas Akhir yang akan digunakan adalah sebagai berikut :

1. **BAB I : PENDAHULUAN**

Bab ini berisi latar belakang, rumusan masalah, batasan masalah, tujuan, manfaat, metodologi dan sistematika penulisan Tugas Akhir.

2. **BAB II : DASAR TEORI**

Bab ini berisi dasar teori mengenai permasalahan dan algoritma penyelesaian yang digunakan dalam Tugas Akhir

3. **BAB III : DESAIN**

Bab ini berisi desain algoritma dan struktur data yang digunakan dalam penyelesaian permasalahan.

4. **BAB IV : IMPLEMENTASI**

Bab ini berisi implementasi berdasarkan desain algoritma yang telah dilakukan pada tahap desain.

5. **BAB V : UJI COBA DAN EVALUASI**

Bab ini berisi uji coba dan evaluasi dari hasil implementasi yang telah dilakukan pada tahap implementasi.

6. **BAB VI : PENUTUP**

Bab ini berisi kesimpulan dan saran yang didapat dari hasil uji coba yang telah dilakukan.

[Halaman ini sengaja dikosongkan]

BAB II

DASAR TEORI

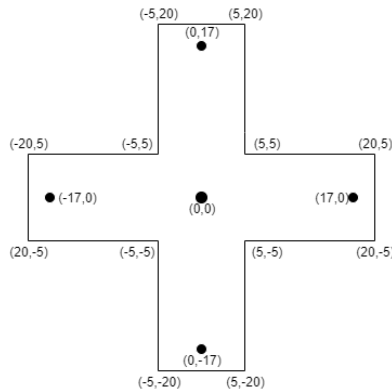
Pada bab ini, akan dijelaskan dasar teori yang digunakan sebagai landasan pengerjaan Tugas Akhir ini.

2.1 Deskripsi Permasalahan

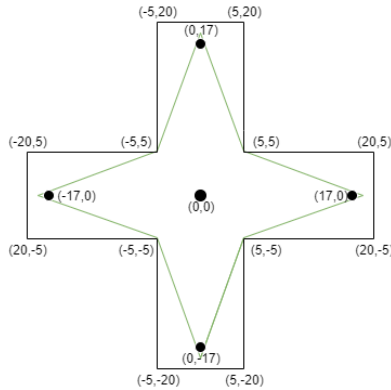
Permasalahan yang dibahas pada Tugas Akhir ini adalah perhitungan untuk mencari nilai x yang didefinisikan oleh persamaan (2.1).

$$x = \sum_{i=0}^{n-1} RCH_i \quad (2.1)$$

Sisi polygon dari RCH yang merupakan *relative convex hull* yang didapatkan dari sekumpulan titik yang dibatasi di dalam polygon sederhana[1] dinyatakan dalam RCH_i pada persamaan (2.1). Permasalahan pada tugas akhir ini adalah mencari *relative convex hull* dari sekumpulan titik yang dibatasi oleh polygon sederhana. Gambar 2.1 dan 2.2 merupakan contoh dari permasalahan ISUN1.



Gambar 2.1 Ilustrasi Contoh Kasus Tanpa Solusi



Gambar 2.2 Ilustrasi Contoh Kasus

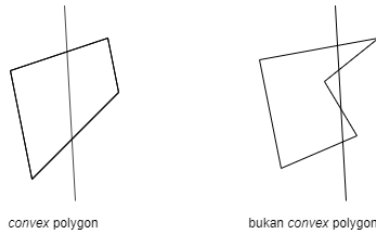
2.2 Convex Polygon

Convex polygon merupakan sebuah polygon sederhana yang memiliki sudut maksimal 180 derajat pada tiap edgenya. *Convex polygon* memiliki beberapa properti, yaitu:

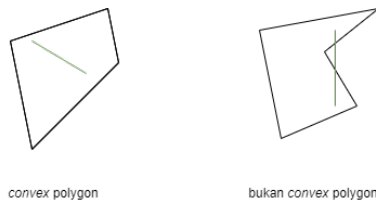
1. Sebuah garis lurus yang di gambar melewati sebuah *convex polygon* akan berpotongan maksimal 2 kali. Ilustrasi dapat dilihat pada gambar 2.3.
2. Jika dua titik sembarang diambil dan ditarik garis antara keduanya, tidak ada bagian dari garis yang berada di luar polygon. Ilustrasi dapat dilihat pada gambar 2.4.

2.3 Relative Convex Polygon

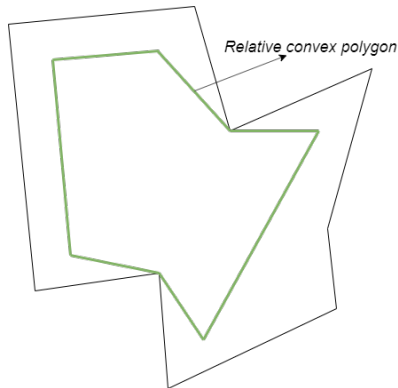
Relative convex polygon merupakan penurunan dari *convex polygon* tetapi ada beberapa sisi dari polygon tersebut berbentuk *convace* atau cekung ke dalam dikarenakan adanya batasan dari luar seperti polygon atau segmen garis lainnya. Ilustrasi *relative convex polygon* dapat dilihat pada gambar 2.5.



Gambar 2.3 Ilustrasi Properti Convex Polygon 1



Gambar 2.4 Ilustrasi Properti Convex Polygon 2



Gambar 2.5 Ilustrasi Relative Convex Polygon

2.4 Strategi Penyelesaian Permasalahan

Pada subbab ini akan dipaparkan mengenai strategi penyelesaian masalah klasik pada daring SPOJ dengan kode ISUN1 meng-

gunakan algoritma reduksi polygon. Secara singkat, strategi penyelesaian masalah dari ISUN1 menggunakan algoritma reduksi polygon menjadi 2 bagian besar, yaitu :

1. Pemrosesan titik pembentuk polygon yang membentuk *Convex*.
2. *Convex Hull* dari titik yang berada di dalam polygon.

Sebagai contoh, pada subbab ini akan digunakan beberapa variabel seperti, P sebagai polygon luar yang mempunyai n vertex, dimana $P = \langle p_1, p_2, \dots, p_n \rangle$ yang mempunyai titik sebanyak m ($S = \langle s_1, s_2, \dots, s_m \rangle$), dan $D(A)$ merupakan sebuah deque (*doubly-ended queue*) yang menampung vertex dari polygon P . Reduksi polygon didasari dari algoritma *Melkman convex hull* dengan sedikit modifikasi. Modifikasi yang dilakukan adalah ketika 3 buah titik pembentuk polygon yang konsekutif membuat *convex* maka titik tengah dari ketiga titik tersebut dibuang, dan jika *concave* maka titik tengahnya tetap disimpan. Pada saat sebuah titik dibuang, maka luas dari polygon akan tereduksi. Langkah-langkah reduksi dilakukan dengan mengulangi 2 langkah yang akan dijelaskan pada subbab 2.4.1 dan 2.4.2.

2.4.1 Pemrosesan Titik Pembentuk Polygon yang Membentuk Convex

Pemrosesan titik pembentuk polygon dapat dilakukan dengan cara melakukan *traversing* terhadap semua vertex pembentuk polygon. Untuk setiap vertex p_i yang di periksa, hitung orientasi(secara berlawanan arah jarum jam) titik p_i dengan p_{i-1} dan p_{i+1} . Jika orientasinya membentuk *convex*, maka titik p_i akan dibuang.

Sebelum membuang titik p_i , kita akan membuat sebuah segitiga ABC dimana $A = p_i$, $B = p_{i-1}$, dan $C = p_{i+1}$ karena *triangulation of polygon* (Teorema 1).

Teorema 1 (Triangulation of Polygon) *Semua polygon dapat di*

buat dari beberapa segitiga.

Kemudian cari $T(ABC)$ dimana $T(ABC)$ merupakan semua titik S yang berada di dalam segitiga ABC dengan menggunakan algoritma *Point Inside Polygon* (dapat dilihat pada subbab 2.6). Pencarian titik yang berada di dalam segitiga ABC berguna untuk mencari pengganti vertex p_i sebagai pembentuk polygon luarnya.

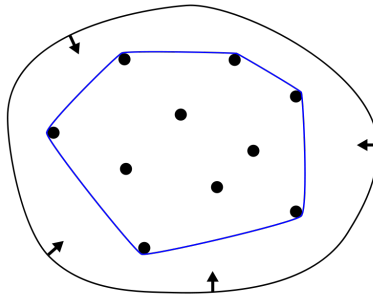
2.4.2 Convex Hull dari Titik yang Berada di Dalam Polygon

Melanjutkan dari subbab 2.4.1, ketika sudah mendapatkan $T(ABC)$, lakukan pencarian *Convex Hull* dari titik-titik tersebut menggunakan algoritma *monotone chain* (dapat dilihat pada subbab 2.5.2.2). Kemudian sisipkan semua titik yang membentuk *Convex Hull* di antara vertex p_{i-1}, p_{i+1} untuk me-rekonstruksi polygon luar yang sudah direduksi.

2.5 Convex Hull

Convex Hull dari sekumpulan titik S adalah sebuah set dari semua kombinasi *convex* dari titik-titik tersebut. Setiap titik s_i pada S diberikan sebuah koefisien a_i dimana a_i merupakan bilangan non negatif dan jika semua a_i dijumlahkan hasilnya satu. Dan koefisien ini digunakan untuk menghitung berat rata-rata untuk setiap titik. Untuk setiap koefisien yang dipilih akan dikombinasikan dan menghasilkan *convex hull*. Set *convex hull* ini dapat diekspresikan dengan formula (2.2) dan ilustrasi *convex hull* ada pada gambar 2.6.

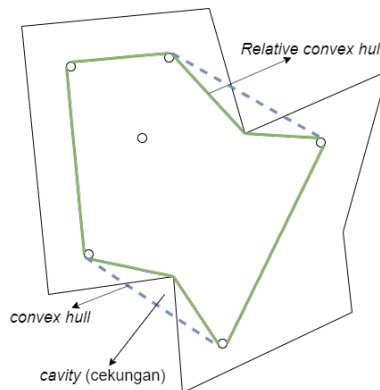
$$Conv(S) = \left\{ \sum_{i=1}^{|S|} a_i s_i \mid (\forall i : a_i \geq 0 \wedge \sum_{i=1}^{|S|} a_i = 1) \right\} \quad (2.2)$$



Gambar 2.6 Ilustrasi Convex Cull

2.5.1 Relative Convex Hull

Relative convex hull merupakan penurunan dari *convex hull*. *Relative convex hull* merupakan *convex hull* yang mempunyai *cavity* (cekungan ke dalam) yang diakibatkan atau relatif terhadap sesuatu yang membatasi *convex hull* tersebut. Ilustrasi *relative convex hull* dapat dilihat pada gambar 2.7.



Gambar 2.7 Ilustrasi Relative Convex Hull

Penentuan untuk mengetahui sebuah polygon merupakan *convex* atau *concave* dapat menggunakan orientasi. Apabila ori-

Tabel 2.1 Tabel Perbandingan Algoritma Convex Hull

Algoritma Convex Hull	Implementasi	Kompleksitas	Kode Sumber	Jenis Input
Jarvis's Algorithm	Mudah	$\mathcal{O}(n^2)$	Singkat	Kumpulan Titik
Graham's Scan	Sedikit Mudah	$\mathcal{O}(n \log(n))$	Singkat	Kumpulan Titik
Quick Hull	Kompleks	$\mathcal{O}(n \log(n))$	Panjang	Kumpulan Titik
Monotone Chain	Mudah	$\mathcal{O}(n \log(n))$	Singkat	Kumpulan Titik
Melkman's Algorithm	Mudah	$\mathcal{O}(n)$	Singkat	Polygon Sederhana atau Polyline

entasi dari tiga titik yang berurutan adalah positif berlawanan arah jarum jam maka tiga titik tersebut adalah *convex*. Sebaliknya apabila negatif maka tiga titik tersebut adalah *concave*. Untuk mencari orientasi antara tiga titik dapat digunakan persamaan 2.3.

$$\begin{aligned}\vec{u} &= (B_x - A_x)x + (B_y - A_y)y \\ \vec{v} &= (C_x - A_x)x + (C_y - A_y)y\end{aligned}\tag{2.3}$$

$$Orientasi = u_x * v_y - u_y * v_x$$

2.5.2 Algoritma Convex Hull

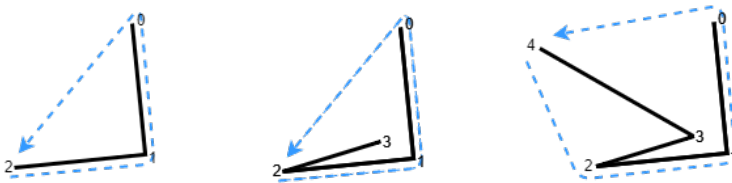
Ada beberapa algoritma yang dapat digunakan untuk mencari sebuah *convex hull*, untuk melihat perbandingan dari beberapa algoritma dapat dilihat pada tabel 2.1. Berdasarkan tabel 2.1, penulis memilih 2 algoritma yang akan digunakan pada buku ini.

2.5.2.1 Algoritma Melkman Convex Hull

Algoritma *Melkman convex hull* merupakan algoritma untuk menghitung rantai polygonal ataupun polygon sederhana dengan waktu linear $O(n)$ [2]. Asumsikan sebuah polygon sederhana P , dengan vertex p_i dan edge $p_i p_{i+1}$. Algoritma ini menggunakan deque, $D = \langle d_1, d_2, \dots, d_n \rangle$, untuk merepresentasikan *convex hull*, $Q_i = CH(P_i)$, dimana $P_i = (p_0, p_1, \dots, p_i)$. Deque mempunyai fungsi *push* dan *pop* dari atas/depan dan *insert* dan *remove* dari bawah/belakang. Secara spesifiknya yang dilakukan *push* v ke deque melakukan $(l \leftarrow l+1; d_l \leftarrow v)$, untuk *pop d_t dari deque melakukan $(t \leftarrow t-1)$, untuk *insert* v ke deque melakukan $(b \leftarrow b+1; d_b \leftarrow v)$, dan *remove* d_b dari deque melakukan $(b \leftarrow b-1)$.*

Algoritma ini menggunakan konvensi dimana vertexnya berurutan secara berlawanan arah jarum jam di sekitar *convex hull* Q .

Setiap d_t dan d_b mengacu kepada vertex yang sama pada rantai polygon C , dan vertex ini akan selalu menjadi vertex yang kita tambahkan terakhir pada *convex hull*. Pseudocode *Melkman convex hull* dapat dilihat pada pseudocode 2.1.



Gambar 2.8 Ilustrasi Algoritma Melkman

Algoritma *Melkman convex hull* sendiri tidak dapat dapat digunakan untuk mencari *relative convex hull*, tetapi beberapa ide dari *Melkman convex hull* nantinya akan digunakan untuk mendapatkan solusi dalam mencari *relative convex hull*. Ide yang diambil penulis dari *Melkman convex hull* adalah cara algoritma tersebut dalam melakukan *traversing* polygon atau segmen garis dan cara algori-

tma tersebut menyimpan atau membuang suatu titik yang dianggap tidak berguna dalam pembentukan polygon yang diinginkan. Ide yang pertama adalah *traversing* pada vertex dapat dilakukan dengan queue ataupun deque. Ide yang kedua adalah untuk masing-masing vertex yang di *traverse* dapat dicari tahu apakah titik atau vertex tersebut dapat dibuang atau tidak dengan melihat orientasinya dengan titik sebelum dan sesudahnya.

Pseudocode 2.1 Melkman Convex Hull

Input: P

Output: Q

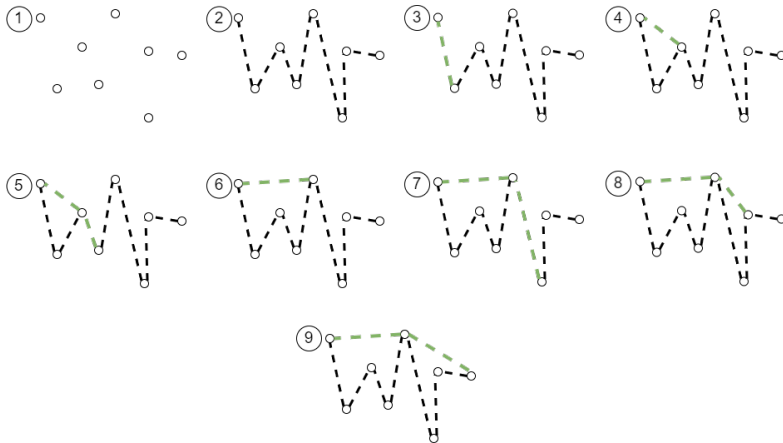
```

1: Inisialisasi:  $D$ 
2: if LEFT( $p_0, p_1, p_2$ ) then
3:    $D \leftarrow \langle p_2, p_0, p_1, p_2 \rangle$ 
4: else
5:    $D \leftarrow \langle p_2, p_1, p_0, p_2 \rangle$ 
6: end if
7:  $i = 3$ 
8: while  $i < n$  do
9:   while LEFT( $d_{t-1}, d_t, p_i$ ) dan LEFT( $d_b, d_{b+1}, p_i$ ) do
10:     $i \leftarrow i + 1$ 
11:   end while
12:   while !LEFT( $d_{t-1}, d_t, p_i$ ) do
13:    pop  $d_t$ 
14:   end while
15:   push  $p_i$ 
16:   while !LEFT( $p_i, d_b, d_{b+1}$ ) do
17:    remove  $d_b$ 
18:   end while
19:   insert  $p_i$ 
20:    $i \leftarrow i + 1$ 
21: end while

```

2.5.2.2 Algoritma Monotone Chain

Algoritma *monotone chain* merupakan proses pembentukan *convex hull* dari sekumpulan titik dengan kompleksitas $\mathcal{O}(n \log(n))$ [3]. Asumsikan sekumpulan titik S sejumlah n , $S = \langle s_1, s_2, \dots, s_n \rangle$ algoritma ini menggunakan list untuk membentuk sebuah rantai (*monotone chain*), dimana list $L(S)$ menampung semua titik yang ada di S yang terurut berdasarkan nilai koordinatnya terhadap sumbu x . Algoritma ini memeriksa setiap 3 vertex yang berurutan, jika 3 vertex tersebut membuat *convex* maka ketiga vertex tersebut disimpan, dan sebaliknya jika ketiga vertex tersebut membuat *concave* maka vertex ke 2 akan dibuang dari vertex penyusun *convex hull*. Lalu lakukan hal yang sama dengan membalikkan urutan pada L untuk mendapatkan *lower hull*. Pseudocode algoritma *Monotone Chain* dapat dilihat pada pseudocode 2.2.



Gambar 2.9 Ilustrasi Algoritma Monotone Chain

Algoritma *monotone chain* ini nantinya akan digunakan pada saat pencarian titik pengganti ketika suatu titik pembentuk polygon akan dibuang.

Pseudocode 2.2 Monotone Chain Algorithm

Input: S
Output: $CH(S)$

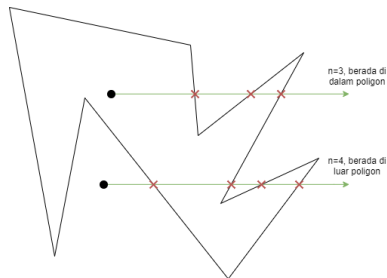
```

1: Inisialisasi:  $L$ 
2: Sort  $S$ 
3:  $L \leftarrow S$ 
4: Inisialisasi  $CH(S)$ 
5: for  $i = 0; i < 2; i++$  do
6:   for  $j = 0; j < Size(L); j++$  do
7:     while  $Size(CH) \geq 2$  and  $right(CH[Size(CH) -$ 
       $1], CH[Size(CH) - 2], S[j])$  do
8:       Delete  $CH$  last element
9:     end while
10:    push  $pt$  to  $CH$ 
11:   end for
12:   reverse  $L$ 
13: end for

```

2.6 Point Inside Polygon

Point inside polygon merupakan algoritma untuk menentukan apakah suatu polygon berada di dalam sebuah polygon atau tidak [4]. Ide utama dari algoritma ini adalah dengan cara menarik garis sejajar dengan sumbu x dimana garis tersebut berujung pada titik yang ingin dicari lokasinya kemudian hitung ada berapa edge dari polygon yang berpotongan dengan garis tersebut. Jika jumlah edge polygon yang berpotongan adalah ganjil, maka titik tersebut berada dalam polygon, dan sebaliknya, jika jumlahnya genap maka titik tersebut berada di luar polygon.



Gambar 2.10 Ilustrasi Algoritma Point Inside Polygon

BAB III

DESAIN

Pada bab ini akan dijelaskan desain algoritma yang akan digunakan untuk menyelesaikan permasalahan.

3.1 Desain Umum Sistem

Pada subbab ini akan dijelaskan mengenai gambaran secara umum dari algoritma yang dirancang. Sistem diawali dengan menerima masukan 2 buah bilangan bulat N yang merupakan banyaknya vertex pembentuk polygon luar dan M yang merupakan banyaknya titik yang ada di dalam polygon tersebut. N baris berikutnya berisikan 2 buah bilangan bulat x_i, y_i yang merupakan koordinat dari vertex pembentuk polygon luar terurut berlawanan arah jarum jam. M baris berikutnya berisikan dua buah bilangan bulat $x1_i, y1_i$ yang merupakan koordinat dari titik yang ada di dalam polygon.

3.2 Desain Fungsi Main

Fungsi MAIN merupakan fungsi yang bertanggung jawab untuk menerima masukan yang sudah dijelaskan pada 3.1 untuk dilakukan proses selanjutnya. Pseudocode fungsi MAIN dapat dilihat pada pseudocode 3.1. Fungsi INPUT merupakan fungsi untuk menerima masukan, dan fungsi PRINT merupakan fungsi untuk menampilkan hasil.

3.3 Desain Class Point

Class POINT adalah class untuk menyimpan titik dalam diagram Kartesius. Pseudocode 3.2 merupakan pseudocode dari class POINT. Nantinya pada implementasi, class ini akan melakukan *override* terhadap operator perbandingan.

Pseudocode 3.1 Fungsi MAIN

```

1: while ( $N \leftarrow \text{INPUT}()$ ) and  $N \neq \text{EOF}$  do
2:    $M \leftarrow \text{INPUT}()$ 
3:    $\text{perimeter} \leftarrow \text{POLYGON}$ 
4:    $\text{trees} \leftarrow \text{Array POINT}$ 
5:   for  $i \leftarrow 1, N$  do
6:      $x_i, y_i \leftarrow \text{INPUT}()$ 
7:      $\text{perimeter}.P[i] \leftarrow \text{POINT}(x_i, y_i, \text{false})$ 
8:   end for
9:   if  $M = 1$  or  $M = 0$  then
10:     $\text{PRINT}(0)$ 
11:     $\text{CONTINUE}$ 
12:  end if
13:  for  $i \leftarrow 1, M$  do
14:     $x1_i, y1_i \leftarrow \text{INPUT}()$ 
15:     $\text{trees} \leftarrow \text{POINT}(x1_i, y1_i, \text{true})$ 
16:  end for
17:   $\text{ans} \leftarrow \text{SOLVE}(\text{perimeter}, \text{trees})$ 
18:   $\text{PRINT}(\text{ans})$ 
19: end while

```

Pseudocode 3.2 Class POINT

```

1:  $x, y \leftarrow \text{double}$ 
2:  $\text{fixed} \leftarrow \text{boolean}$ 
3: constructor  $\text{POINT}()$ 
4: constructor  $\text{POINT}(\_x, \_y, \_fixed)$ 

```

Tabel 3.1 Nama dan Fungsi Variabel dalam Class POINT

Nama Variabel	Fungsi Variabel
x	Menyimpan ordinat dari titik tersebut
y	Menyimpan absis dari titik tersebut
$fixed$	Untuk membedakan antara titik pembentuk polygon P dan titik yang ada di dalam kumpulan titik S

Class POINT tidak memiliki fungsi karena class ini memang hanya untuk menyimpan suatu titik yang akan digunakan nanti.

Fungsi *Constructor* dari class ini terdiri dari dua jenis. Fungsi *constructor* yang pertama adalah fungsi dengan tanpa parameter, pada *constructor* ini, semua variabel yang ada di dalam class POINT akan di inisialisasi dengan 0. Fungsi *constructor* kedua adalah fungsi dengan parameter $_x, _y, _fixed$, menyatakan nilai $x, y, fixed$ secara berurutan.

3.4 Desain Class Vec

Class VEC merupakan class yang menyimpan vector dari dua buah titik pada diagram kartesian. Pseudocode 3.3 merupakan pseudocode dari class VEC.

Tabel 3.2 Nama dan Fungsi Variabel dalam Class VEC

Nama Variabel	Fungsi Variabel
x	Menyimpan arah vektor absis
y	Menyimpan arah vektor ordinat

Class VEC tidak memiliki fungsi karena class ini hanya untuk menyimpan vector dari dua titik yang akan digunakan nanti.

Fungsi *Constructor* dari class ini terdiri dari 3 jenis. Fungsi *constructor* yang pertama adalah fungsi dengan tanpa paramete-

Pseudocode 3.3 Class VEC

```

1:  $x, y \leftarrow \text{double}$ 
2: constructor VEC()
3: constructor VEC( $_x, _y$ )
4: constructor VEC( $A, B$ )

```

ter, pada *constructor* ini, semua variabel yang ada di dalam class VEC akan di inisialisasi dengan 0. Fungsi *constructor* kedua adalah fungsi dengan parameter $_x, _y$, menyatakan nilai x, y secara berurutan. Fungsi *constructor* ketiga adalah fungsi dengan parameter A, B , menyatakan POINT dari titik A dan POINT dari titik B , dimana nantinya nilai x dan y akan didapatkan dari pengurangan koordinat dari POINT A dan POINT B .

3.5 Desain Class Line

Class LINE merupakan class yang bertanggung jawab untuk melakukan operasi-operasi pada garis dalam diagram kartesian. Pseudocode 3.4 merupakan pseudocode dari Class LINE.

Tabel 3.3 Nama dan Fungsi Variabel dalam class LINE

Nama Variabel	Fungsi Variabel
a	Menyimpan nilai a pada persamaan $ax + by + c = 0$
b	Menyimpan nilai b pada persamaan $ax + by + c = 0$
c	Menyimpan nilai c pada persamaan $ax + by + c = 0$

Class LINE tidak memiliki fungsi karena class ini hanya untuk menyimpan nilai dari fungsi $ax + by + c = 0$ yang akan digunakan nanti.

Pseudocode 3.4 Class LINE

```

1:  $a, b, c \leftarrow \text{double}$ 
2: constructor LINE()
3: constructor LINE( $\_a, \_b, \_c$ )
4: constructor LINE( $A, B$ )

```

Fungsi *Constructor* dari class ini terdiri dari 3 jenis. Fungsi *constructor* yang pertama adalah fungsi dengan tanpa parameter, pada *constructor* ini, semua variabel yang ada di dalam class LINE akan di inisialisasi dengan 0. Fungsi *constructor* kedua adalah fungsi dengan parameter $_a, _b, _c$, menyatakan nilai a, b, c secara berurutan. Fungsi *constructor* ketiga adalah fungsi dengan parameter A, B , menyatakan POINT dari titik A dan POINT dari titik B , dimana nantinya nilai a, b dan c akan didapatkan dengan mencari fungsi garis yang melewati POINT A dan POINT B .

3.6 Desain Class Segment

Class SEGMENT merupakan class yang bertanggung jawab untuk menyimpan dan melakukan operasi-operasi pada segmen garis dalam diagram kartesian. Pseudocode 3.5 merupakan pseudocode dari class SEGMENT.

Tabel 3.4 Nama dan Fungsi Variabel dalam Class SEGMENT

Nama Variabel	Fungsi Variabel
P	Menyimpan POINT yang merupakan ujung awal dari sebuah segmen garis
Q	Menyimpan POINT yang merupakan ujung akhir dari sebuah segmen garis
L	Menyimpan fungsi dari garis yang melalui dua titik tersebut

Class SEGMENT tidak memiliki fungsi karena class ini hanya

Pseudocode 3.5 Class SEGMENT

```
1:  $P, Q \leftarrow \text{POINT}$ 
2:  $L \leftarrow \text{LINE}$ 
3: constructor SEGMENT()
4: constructor SEGMENT( $\_P, \_Q$ )
```

untuk menyimpan data dari sebuah segmen garis yang akan digunakan nanti.

Fungsi *Constructor* dari class ini terdiri dari 2 jenis. Fungsi *constructor* yang pertama adalah fungsi dengan tanpa parameter, pada *constructor* ini, semua variabel yang ada di dalam class SEGMENT akan di inisialisasi dengan 0. Fungsi *constructor* kedua adalah fungsi dengan parameter $_P, _Q$, menyatakan POINT dari titik P dan POINT dari titik Q , yang merupakan titik POINT A dan POINT B secara berturut, dan LINE L didapat dengan menggunakan *constructor* LINE dengan parameter P dan Q .

3.7 Desain Class Polygon

Class POLYGON merupakan class yang bertanggung jawab untuk menyimpan dan melakukan operasi-operasi pada polygon pada diagram kartesian. Pseudocode 3.6 merupakan pseudocode dari class POLYGON.

Tabel 3.5 Nama dan Fungsi Variabel dalam Class POLYGON

Nama Variabel	Fungsi Variabel
P	Menyimpan array dari POINT yang membentuk polygon tersebut

Fungsi-fungsi yang terkandung dalam class ini adalah PREV, NEXT, PERIMETER. Tabel 3.5 menjelaskan variabel dan kegunaannya dalam class POLYGON.

Pseudocode 3.6 Class POLYGON

```

1:  $P \leftarrow$  Array POINT
2: constructor POLYGON()
3: constructor POLYGON( $\_P$ )
4: function PREV( $idx$ )
5: function NEXT( $idx$ )
6: function PERIMETER()

```

Fungsi *Constructor* dari class ini terdiri dari 2 jenis. Fungsi *constructor* yang pertama adalah fungsi dengan tanpa parameter, pada *constructor* ini, variabel P yang ada di dalam class POLYGON akan di inisialisasi. Fungsi *constructor* kedua adalah fungsi dengan parameter $_P$, menyatakan array POINT dari titik pembentuk polygon tersebut.

Fungsi *next* bertanggung jawab untuk mencari index selanjutnya dari titik yang membentuk polygon. Masukan, proses dan keluaran dari fungsi ini tercantum pada tabel 3.6. Pseudocode fungsi ini dapat dilihat pada pseudocode 3.7.

Tabel 3.6 Masukan, Proses, dan Keluaran dari Fungsi NEXT Class POLYGON

Masukan	Proses	Keluaran
Suatu bilangan bulat idx yang menyatakan index saat ini	mencari index selanjutnya	Suatu bilangan bulat yang menyatakan index selanjutnya

Fungsi *prev* bertanggung jawab untuk mencari index sebelumnya dari titik yang membentuk polygon. Masukan, proses dan keluaran dari fungsi ini tercantum pada tabel 3.7. Pseudocode fungsi ini dapat dilihat pada pseudocode 3.8.

Fungsi *perimeter* bertanggung jawab untuk mencari keliling

Pseudocode 3.7 Fungsi NEXT pada class POLYGON

Input: idx

```
1: if  $idx = \text{SIZE}(P) - 1$  then
2:   return 0
3: else
4:   return  $idx + 1$ 
5: end if
```

Tabel 3.7 Masukan, Proses, dan Keluaran dari Fungsi PREV Class POLYGON

Masukan	Proses	Keluaran
Suatu bilangan bulat idx yang menyatakan index saat ini	mencari index sebelumnya	Suatu bilangan bulat yang menyatakan index sebelumnya

dari sebuah polygon. Masukan, proses, dan keluaran dari fungsi ini tercantum pada tabel 3.8. Pseudocode fungsi ini dapat dilihat pada pseudocode 3.9.

Pseudocode 3.8 Fungsi PREV pada class POLYGON

Input: idx

```
1: if  $idx = 0$  then
2:   return  $\text{SIZE}(P) - 1$ 
3: else
4:   return  $idx - 1$ 
5: end if
```

Tabel 3.8 Masukan, Proses, dan Keluaran dari Fungsi PERIMETER
Class POLYGON

Masukan	Proses	Keluaran
-	Mencari keliling dengan mencari jarak eulerian dari semua titik pembentuk polygon	Suatu bilangan berkoma yang menyatakan keliling dari polygon tersebut

Pseudocode 3.9 Fungsi PERIMETER pada class POLYGON

```

1:  $ret \leftarrow 0$ 
2: for  $i \leftarrow 0$  to  $SIZE(P)-1$  do
3:    $ret \leftarrow ret + EDIST(P[i], P[NEXT(i)])$ 
4: end for
5: return  $ret$ 

```

3.8 Fungsi BetweenD

Fungsi BETWEEND bertanggung jawab untuk mencari tahu apakah suatu bilangan x berada diantara bilangan l dan bilangan r . Pseudocode dari fungsi BETWEEND dapat dilihat pada pseudocode 3.10. Masukan, proses, dan keluaran dari fungsi ini tercantum pada tabel 3.9.

Pseudocode 3.10 Fungsi BETWEEND

Input: x, l, r

```

1: if  $MIN(l, r) \leq x + EPS$  and  $x \leq MAX(l, r) + EPS$  then
2:   return TRUE
3: else
4:   return FALSE
5: end if

```

Tabel 3.9 Masukan, Proses, dan Keluaran dari Fungsi BETWEEN

Masukan	Proses	Keluaran
Tiga buah bilangan x, l, r , dimana bilangan x merupakan bilangan yang ingin diketahui apakah berada diantara titik l dan r	Mencari tahu apakah bilangan x berada diantara bilangan l dan r	Sebuah boolean yang menyatakan apakah x berada diantara l dan r

3.9 Fungsi EDist

Fungsi EDIST bertanggung jawab untuk mencari jarak antara dua titik POINT A dan POINT B dengan menggunakan rumus *Pythagoras*. Rumus *Pythagoras* dapat di lihat pada persamaan 3.1. Pseudocode fungsi EDIST dapat dilihat pada pseudocode 3.11. Masukan, proses, dan keluaran dari fungsi ini tercantum pada tabel 3.10.

$$C = \sqrt{A^2 + B^2}$$

(3.1)

Tabel 3.10 Masukan, Proses, dan Keluaran dari Fungsi EDIST

Masukan	Proses	Keluaran
Dua buah POINT A dan POINT B yang akan dicari jaraknya	Mencari jarak antara POINT A dan POINT B	Sebuah bilangan berkoma yang menyatakan jarak POINT A dan POINT B

Pseudocode 3.11 Fungsi EDIST

Input: A, B

 1: **return** $\text{SQRT}((A * A) + (B * B))$

3.10 Fungsi Cross

Fungsi CROSS bertanggung jawab untuk mencari nilai perkalian *cross* dari dua buah vector. Rumus Pythagoras dapat di lihat pada persamaan 3.2. Pseudocode fungsi CROSS dapat dilihat pada pseudocode 3.12. Masukan, proses, dan keluaran dari fungsi ini tercantum pada tabel 3.11.

$$C = (u_x * v_y) - (u_y * v_x) \quad (3.2)$$

Tabel 3.11 Masukan, Proses, dan Keluaran dari Fungsi CROSS

Masukan	Proses	Keluaran
Dua buah VEC U dan VEC V yang akan dicari hasil perkalian silangnya	Mencari nilai perkalian silang dari VEC U dan VEC V	Sebuah bilangan yang menyatakan hasil perkalian silang VEC U dan VEC V

Pseudocode 3.12 Fungsi CROSS

Input: U, V

 1: **return** $(U.x * V.y) - (U.y * V.x)$

3.11 Fungsi Orientation

Fungsi ORIENTATION bertanggung jawab untuk mencari orientasi dari tiga titik. Pseudocode fungsi ORIENTATION dapat dilihat

pada pseudocode 3.13. Masukan, proses, dan keluaran dari fungsi ini tercantum pada tabel 3.12.

Tabel 3.12 Masukan, Proses, dan Keluaran dari Fungsi
ORIENTATION

Masukan	Proses	Keluaran
Tiga buah POINT O , POINT P dan POINT Q yang akan dicari orientasinya	Mencari orientasi antara POINT O , POINT P dan POINT Q	Sebuah bilangan yang menyatakan orientasi antara POINT O , POINT P dan POINT Q

Pseudocode 3.13 Fungsi ORIENTATION

Input: O, P, Q

- 1: $OP \leftarrow \text{VEC}(O, P)$
 - 2: $OQ \leftarrow \text{VEC}(O, Q)$
 - 3: **return** $\text{CROSS}(OP, OQ)$
-

3.12 Fungsi OnSegment

Fungsi ONSEGMENT bertanggung jawab untuk mencari tahu apakah sebuah titik POINT P bersentuhan dengan sebuah segmen garis SEGMENT S atau tidak. Pseudocode fungsi ONSEGMENT dapat dilihat pada pseudocode 3.14. Masukan, proses, dan keluaran dari fungsi ini tercantum pada tabel 3.13.

3.13 Fungsi ConvexHull

Fungsi CONVEXHULL bertanggung jawab untuk mencari *convex hull* dari sekumpulan titik *pts*. Algoritma yang digunakan oleh fungsi ini adalah algoritma *Monotone Chain* yang dapat dilihat pada

Tabel 3.13 Masukan, Proses, dan Keluaran dari Fungsi
ONSEGMENT

Masukan	Proses	Keluaran
Dua buah POINT P dan SEGMENT S yang akan dicari tahu apakah POINT P berada di SEGMENT S	Mencari tahu apakah POINT P berada di SEGMENT S	Sebuah boolean yang menyatakan apakah POINT P berada di dalam SEGMENT S

Pseudocode 3.14 Fungsi ONSEGMENT

Input: P, S

```

1: if ORIENTATION( $S.P, S.Q, P$ ) then
2:   return FALSE
3: else
4:   return (BETWEEN( $P.x, S.P.x, S.Q.x$ ) and
             BETWEEN( $P.y, S.P.y, S.Q.y$ ))
5: end if

```

bagian 2.5.2.2. Pseudocode dari fungsi CONVEXHULL yang digunakan dapat dilihat pada Pseudocode 3.15. Masukan, proses, dan keluaran dari fungsi ini tercantum pada tabel 3.14.

3.14 Fungsi InSimplePolygon

Fungsi INSIMPLEPOLYGON bertanggung jawab untuk mencari tahu apakah sebuah titik POINT P berada di dalam POLYGON A atau tidak. Algoritma yang digunakan pada fungsi ini adalah algoritma *point inside polygon* yang dapat dilihat pada bagian 2.6. Pada desain fungsi INSIMPLEPOLYGON ada 3 macam keluaran yaitu -1 untuk menandakan bahwa POINT P berada di dalam POLYGON A , 0 untuk menandakan bahwa POINT P berada di salah satu si-

Pseudocode 3.15 Fungsi CONVEXHULL

Input: pts

```

1: SORT( $pts$ )
2:  $hull \leftarrow$  Array POINT
3: for  $i \leftarrow 0$  to 2 do
4:    $start \leftarrow$  SIZE( $hull$ )
5:   for  $pt$  in  $pts$  do
6:     while (SIZE( $hull$ )  $\geq$   $start + 2$ ) and
      (ORIENTATION( $hull$ [SIZE( $hull$ )-1],  $hull$ [SIZE( $hull$ )-2],
       $pt$ )  $\leq 0$ ) do
7:        $hull.ERASE(hull[SIZE(hull)-1])$ 
8:     end while
9:      $hull.INSERT(pt)$ 
10:  end for
11:   $hull.ERASE(hull[SIZE(hull)-1])$ 
12:  REVERSE( $pts$ )
13: end for
14: return POLYGON( $hull$ );

```

Tabel 3.14 Masukan, Proses, dan Keluaran dari Fungsi
CONVEXHULL

Masukan	Proses	Keluaran
Sebuah array POINT pts yang merupakan sekum- pulan titik	Mencari POLYGON dengan keliling terkecil dari se- kumpulan titik	Sebuah POLYGON yang mengelilingi kumpulan POINT pts

si POLYGON A , dan 1 untuk menandakan bahwa POINT P berada di luar POLYGON A . Pseudocode fungsi INSIMPLEPOLYGON dapat dilihat pada pseudocode 3.16. Masukan, proses, dan keluaran dari fungsi ini tercantum pada tabel 3.15.

Tabel 3.15 Masukan, Proses, dan Keluaran dari Fungsi
INSIMPLEPOLYGON

Masukan	Proses	Keluaran
Sebuah POINT A dan sebuah POLYGON P yang akan dicari tahu apakah POINT A berada di dalam POLYGON P	Mencari tahu apakah POINT A berada di dalam POLYGON P	Sebuah bilangan yang apakah POINT A berada di dalam POLYGON P

3.15 Fungsi GetBetween

Fungsi GETBETWEEN bertanggung jawab untuk mencari list POINT yang akan menggantikan POINT yang akan dibuang. Pseudocode fungsi GETBETWEEN dapat dilihat pada pseudocode 3.17. Masukan, proses, dan keluaran dari fungsi ini tercantum pada tabel 3.16.

Pseudocode 3.16 Fungsi INSIMPLEPOLYGON

Input: P, A

```

1:  $ret \leftarrow \text{INTEGER}$ 
2: for  $i \leftarrow 0$  to  $\text{SIZE}(A.P)$  do
3:   if  $P = A.P[i]$  then
4:     return 0
5:   end if
6:    $j \leftarrow A.\text{NEXT}(i)$ 
7:   if  $\text{ONSEGMENT}(P, \text{SEGMENT}(A.P[i], A.P[j]))$  then
8:     return 0
9:   end if
10:   $below \leftarrow (A.P[i].y < P.y)$ 
11:  if  $below \neq (A.P[j].y < P.y)$  then
12:     $o \leftarrow \text{ORIENTATION}(P, A.P[i], A.P[j])$ 
13:    if  $o = 0$  then
14:      return 0
15:    end if
16:    if  $below = (o > 0)$  and  $below = \text{TRUE}$  then
17:       $ret+ = 1$ 
18:    else
19:      if  $below = (o > 0)$  and  $below = \text{FALSE}$  then
20:         $ret- = 1$ 
21:      end if
22:    end if
23:  end if
24: end for
25: if  $ret = 0$  then
26:   return 1
27: else
28:   return -1
29: end if

```

Pseudocode 3.17 Fungsi GETBETWEEN

Input: *triangle, q, trees*

```

1: points, pts  $\leftarrow$  Array POINT
2: while q not EMPTY do
3:   if INSIMPLEPOLYGON(q.FRONT(), triangle)  $\neq$  1 then
4:     points.INSERT(q.FRONT())
5:   end if
6:   q.POP
7: end while
8: for pt in trees do
9:   if INSIMPLEPOLYGON(pt, triangle)  $\neq$  1 then
10:    points.INSERT(pt)
11:   end if
12: end for
13: P  $\leftarrow$  CONVEXHULL(points)
14: i  $\leftarrow$  0
15: while TRUE do
16:   if P.P[i] = triangle.P[0] then
17:     if P.P[P.NEXT(i)] = triangle.P[2] then
18:       i  $\leftarrow$  P.PREV(i)
19:       while P.P[i]  $\neq$  triangle.P[2] do
20:         pts.INSERT(P.P[i])
21:         i = P.PREV(i)
22:       end while
23:     else
24:       i  $\leftarrow$  P.NEXT(i)
25:       while P.P[i]  $\neq$  triangle.P[2] do
26:         pts.INSERT(P.P[i])
27:         i = P.NEXT(i)
28:       end while
29:     end if
30:     BREAK
31:   end if
32:   i  $\leftarrow$  P.NEXT(i)
33: end while
34: return pts

```

Tabel 3.16 Masukan, Proses, dan Keluaran dari Fungsi
GETBETWEEN

Masukan	Proses	Keluaran
Sebuah POLYGON <i>triangle</i> yang merupakan segitiga, QUEUE POINT <i>q</i> yang merupakan pembentuk polygon luar, dan array POINT <i>trees</i> yang merupakan titik yang berada di dalam polygon luar	Mencari POINT mana saja yang akan menggantikan POINT <i>triangle</i> [1] yang akan dibuang	Sebuah LIST POINT yang berisikan daftar POINT yang akan menggantikan POINT <i>triangle</i> [1]

3.16 Fungsi Solve

Fungsi SOLVE bertanggung jawab untuk mencari *relative convex polygon* yang mengelilingi semua titik yang ada di dalam polygon luar. Pseudocode fungsi SOLVE dapat dilihat pada pseudocode 3.18. Masukan, proses, dan keluaran dari fungsi ini tercantum pada tabel 3.17.

Pseudocode 3.18 Fungsi SOLVE

Input: *perimeter, trees*

```

1:  $q \leftarrow \text{QUEUE}$ 
2: for  $pt$  in  $perimeter.P$  do
3:    $q.PUSH\ pt$ 
4: end for
5:  $bef \leftarrow perimeter.P[perimeter.SIZE() - 1]$ 
6: while TRUE do
7:    $erased \leftarrow \text{FALSE}$ 
8:    $count \leftarrow q.SIZE()$ 
9:   while  $count > 0$  do
10:     $cur \leftarrow q.FRONT() \ q.POP()$ 
11:    if  $cur.fixed = \text{FALSE}$  and ( $FINE(q, cur)$  or  $cur = bef$ 
    or  $cur = q.FRONT()$ ) then
12:       $erased \leftarrow \text{TRUE}$ 
13:       $triangle \leftarrow \text{POLYGON}$ 
14:       $triangle.P.INSERT(bef, cur, q.FRONT())$ 
15:       $points \leftarrow \text{GETBETWEEN}(triangle, q, trees)$ 
16:      for  $pt$  in  $points$  do
17:         $q.PUSH(pt); bef \leftarrow pt$ 
18:      end for
19:    else
20:       $q.PUSH(cur); bef \leftarrow cur$ 
21:    end if
22:  end while
23:  if  $erased = \text{FALSE}$  then BREAK
24:  end if
25: end while
26:  $hull \leftarrow \text{array of POINT}$ 
27: while  $q$  not empty do
28:    $hull.INSERT(q.FRONT())$ 
29:    $q.POP()$ 
30: end while
31: return  $\text{POLYGON}(hull)$ 

```

Tabel 3.17 Masukan, Proses, dan Keluaran dari Fungsi SOLVE

Masukan	Proses	Keluaran
Sebuah POLYGON <i>perimeter</i> yang merupakan po-lygon sederhana yang merupakan polygon pembatas, dan array POINT <i>trees</i> yang meru-pakan titik yang berada di dalam polygon pembatas	Mencari <i>relative convex hull</i> dari semua titik <i>trees</i> di dalam POLYGON <i>perimeter</i>	Sebuah POLYGON yang merupakan <i>relative convex hull</i> dari semua titik <i>trees</i>

BAB IV

IMPLEMENTASI

4.1 Lingkungan implementasi

Lingkungan implementasi dan pengembangan yang dilakukan adalah sebagai berikut.

1. Perangkat Keras

- (a) Processor Intel® Core™ i7-6500U CPU @ 2.50GHz (4 CPUs), 2.6GHz
- (b) Random Access Memory 8192MB

2. Perangkat Lunak

- (a) Sistem Operasi Windows 10 Home Single Language 64-bit
- (b) Visual Studio Code
- (c) Bahasa Pemrograman C++
- (d) Kompiler GCC 7.4.0 (Ubuntu 7.4.0-1ubuntu1 18.04.1) untuk Windows Subsystem Linux

4.2 Implementasi Program Utama

Subbab ini menjelaskan implementasi proses algoritma secara keseluruhan berdasarkan desain yang telah dijelaskan pada bab 3. Program ini merupakan program yang digunakan untuk menyelesaikan permasalahan LL and ErBao.

4.2.1 Header yang diperlukan

Implementasi algoritma ini membutuhkan enam buah *header* yaitu `iostream`, `vector`, `cmath`, `map`, `queue`, dan `algorithm` seperti yang terlihat pada kode sumber 4.1.

```

1 #pragma GCC optimize("O3")
2 #pragma GCC target("avx")
3
4 #include <iostream>
5 #include<vector>
6 #include<cmath>
7 #include<map>
8 #include<queue>
9 #include<algorithm>

```

Kode Sumber 4.1 *Header* yang Diperlukan

Selain header, terdapat juga preprocessor *pragma*, digunakan untuk mengganti flag kompiler yang digunakan pada daring SPOJ. *Header* *iostream* berisi fungsi standar input output operasi yang digunakan oleh bahasa C++. *Header* *vector* berisi struktur data yang digunakan untuk menyimpan data array secara dinamis. *Header* *cmath* berisi fungsi-fungsi untuk operasi matematika seperti fungsi *hypot*. *Header* *map* berisi struktur data untuk menyimpan data *key value*. *Header* *queue* berisi struktur data yang digunakan untuk menyimpan antrian data. *Header* *algorithm* berisi modul yang memiliki fungsi-fungsi yang sangat berguna dalam membantu mengimplementasi algoritma yang telah dibangun, contohnya adalah fungsi *reverse* dan *sort*.

4.2.2 Preprocessor

Preprocessor seperti *using* digunakan untuk membuat alias dari tipe data sesungguhnya. Terdapat empat alias yang digunakan yaitu *push_back(x)* sebagai *pb(x)*, *pop_back(x)* sebagai *pob(x)*, *getchar(x)* sebagai *gc(x)*, dan *for (int i = 0; i < n; i++)* sebagai *FOR(i,n)*. Preprocessor dapat dilihat pada kode sumber 4.2.

```

1 #define pb(x) push_back(x)
2 #define pob(x) pop_back(x)
3 #define FOR(i, n) for (int i = 0; i < n; i++)
4 #define gc(x) getchar(x)
5 using namespace std;

```

Kode Sumber 4.2 *Preprocessor* yang Diperlukan

4.2.3 Variabel Global

Variabel global digunakan untuk memudahkan dalam mengakses data yang digunakan lintas fungsi/struct. kode sumber implementasi variabel global dapat dilihat pada kode sumber 4.3. Variabel tersebut didefinisikan secara global agar dapat digunakan pada setiap fungsi.

```

1 const double EPS = 0.0;
2 const double INF = 1E9;
3 map<point, int> pool;

```

Kode Sumber 4.3 Variabel Global yang Didefinisikan Untuk Program

4.2.4 Implementasi Fungsi Main

Fungsi main adalah implementasi algoritma yang dirancang pada pseudocode 3.1. Implementasi fungsi main dapat dilihat pada kode sumber 4.4.

```

1  int main(){
2      int kase = 1;
3      int n, m;
4      while (cin >> n){
5          pool.clear();
6          cin >> m;
7          polygon perimeter;
8          vector<point> trees;
9
10         for (int i = 0; i < n; i++){
11             double a = readint(), b = readint();
12             pool[point(a, b, false)]++;
13             perimeter.P.push_back(point(a, b, false));
14         }
15
16         if (m == 0 || m == 1){
17             printf("Case #d: %.3lf\n", kase++, 0.0);
18             continue;
19         }
20
21         for (int i = 0; i < m; i++){
22             double a = readint(), b = readint();
23             pool[point(a, b, true)]++;
24             trees.push_back(point(a, b, true));
25         }
26
27         polygon hasil = solve(perimeter, trees);
28
29         printf("Case #d: %.3lf\n", kase++,
30             hasil.perimeter());
31     }
32 }

```

Kode Sumber 4.4 Fungsi Main

4.2.5 Implementasi Class Point

Pada subbab ini akan dijelaskan mengenai implementasi dari class Point pada subbab 3.3 dan pseudocode 3.2. Implementasi

dari class Point dapat dilihat pada kode sumber 4.5.

```

1 struct point{
2     double x, y;
3     bool fixed;
4     point(){
5         x = y = 0.0;
6         fixed = 0;
7     }
8     point(double _x, double _y, bool _fixed =
          false){
9         x = _x;
10        y = _y;
11        fixed = _fixed;
12    }
13    bool operator<(point other) const{
14        if (y < other.y + EPS)
15            return true;
16        if (y + EPS > other.y)
17            return false;
18        return x < other.x + EPS;
19    }
20    bool operator==(point other) const{
21        return same_d(x, other.x) && same_d(y,
          other.y);
22    }
23 };

```

Kode Sumber 4.5 Struct Point

4.2.6 Implementasi Class Vec

Pada subbab ini akan dijelaskan mengenai implementasi dari class Vec pada subbab 3.4 dan pseudocode 3.3. Implementasi dari class Vec dapat dilihat pada kode sumber 4.6.

```
1 struct vec{
2     double x, y;
3     vec(){
4         x = y = 0.0;
5     }
6     vec(double _x, double _y){
7         x = _x;
8         y = _y;
9     }
10    vec(point A){
11        x = A.x;
12        y = A.y;
13    }
14    vec(point A, point B){
15        x = B.x - A.x;
16        y = B.y - A.y;
17    }
18 };
```

Kode Sumber 4.6 Struct Vec

4.2.7 Implementasi Class Line

Pada subbab ini akan dijelaskan mengenai implementasi dari class `Line` pada subbab 3.5 dan pseudocode 3.4. Implementasi dari class `Line` dapat dilihat pada kode sumber 4.7.


```

1 struct line{
2     double a, b, c;
3     line(){
4         a = b = c = 0.0;
5     }
6     line(double _a, double _b, double _c){
7         a = _a;
8         b = _b;
9         c = _c;
10    }
11    line(point P1, point P2){
12        if (P2 < P1){
13            point T;
14            T = P1;
15            P1 = P2;
16            P2 = T;
17        }
18        if (same_d(P1.x, P2.x))
19            a = 1.0, b = 0.0, c = -P1.x;
20        else
21            a = -(P1.y - P2.y) / (P1.x - P2.x), b =
                1.0, c = -(a * P1.x) - P1.y;
22    }
23    line(point P, double slope){
24        if (same_d(slope, INF))
25            a = 1.0, b = 0.0, c = -P.x;
26        else
27            a = -slope, b = 1.0, c = -(a * P.x) - P.y;
28    }
29    bool operator==(line other) const{
30        return same_d(a, other.a) && same_d(b,
            other.b) && same_d(c, other.c);
31    }
32 };

```

Kode Sumber 4.7 Struct Line

4.2.8 Implementasi Class Segment

Pada subbab ini akan dijelaskan mengenai implementasi dari class `Segment` pada subbab 3.6 dan pseudocode 3.5. Implementasi dari class `Segment` dapat dilihat pada kode sumber 4.8.

```

1  struct segment{
2      point P, Q;
3      line L;
4      segment(){
5          point T1;
6          P = Q = T1;
7          line T2;
8          L = T2;
9      }
10     segment(point _P, point _Q){
11         if (_Q < _P){
12             point T1 = _P;
13             _P = _Q;
14             _Q = T1;
15         }
16         P = _P;
17         Q = _Q;
18         line T2(_P, _Q);
19         L = T2;
20     }
21     bool operator==(segment other) const{
22         return P == other.P && Q == other.Q;
23     }
24 };

```

Kode Sumber 4.8 Struct Segment

4.2.9 Implementasi Class Polygon

Pada subbab ini akan dijelaskan mengenai implementasi dari class `Polygon` pada subbab 3.7 dan pseudocode 3.6. Implementasi dari class `Polygon` dapat dilihat pada kode sumber 4.9.

```

1 struct polygon{
2     vector<point> P;
3     polygon(){
4         P.clear();
5     }
6     polygon(vector<point> &_P){
7         P = _P;
8     }
9     int prev(int idx){
10         return (idx == 0 ? P.size() - 1 : idx - 1);
11     }
12     int next(int idx){
13         return (idx == P.size() - 1 ? 0 : idx + 1);
14     }
15     double perimeter(){
16         double ret = 0;
17         FOR(i, P.size()){
18             ret += e_dist(P[i], P[next(i)]);
19         }
20         return ret;
21     }
22 };

```

Kode Sumber 4.9 Struct Polygon

4.2.10 Implementasi Fungsi BetweenD

Pada subbab ini akan dijelaskan mengenai implementasi dari fungsi BetweenD pada pseudocode 3.10. Implementasi dapat dilihat pada kode sumber 4.10.

```

1 double between_d(double x, double l, double r){
2     return (min(l, r) <= x + EPS && x <= max(l, r)
3         + EPS);
3 }

```

Kode Sumber 4.10 Fungsi BetweenD

4.2.11 Implementasi Fungsi EDist

Pada subbab ini akan dijelaskan mengenai implementasi dari fungsi EDist pada pseudocode 3.11. Implementasi dapat dilihat pada kode sumber 4.11.

```
1 double e_dist(point P1, point P2){  
2     return hypot(P1.x - P2.x, P1.y - P2.y);  
3 }
```

Kode Sumber 4.11 Fungsi EDist

4.2.12 Implementasi Fungsi Cross

Pada subbab ini akan dijelaskan mengenai implementasi dari fungsi Cross pada pseudocode 3.12. Implementasi dapat dilihat pada kode sumber 4.12.

```
1 double cross(vec u, vec v){  
2     return (u.x * v.y - u.y * v.x);  
3 }
```

Kode Sumber 4.12 Fungsi Cross

4.2.13 Implementasi Fungsi Orientation

Pada subbab ini akan dijelaskan mengenai implementasi dari fungsi Orientation pada pseudocode 3.13. Implementasi dapat dilihat pada kode sumber 4.13.

```
1 double orientation(point O, point P, point Q){  
2     vec OP(O, P), OQ(O, Q);  
3     double c = cross(OP, OQ);  
4     return c;  
5 }
```

Kode Sumber 4.13 Fungsi Orientation

4.2.14 Implementasi Fungsi OnSegment

Pada subbab ini akan dijelaskan mengenai implementasi dari fungsi OnSegment pada pseudocode 3.14. Implementasi dapat dilihat pada kode sumber 4.14.

```

1 bool onSegment(point P, segment S){
2   if (orientation(S.P, S.Q, P) != 0.0)
3     return false;
4   return between_d(P.x, S.P.x, S.Q.x) &&
        between_d(P.y, S.P.y, S.Q.y);
5 }

```

Kode Sumber 4.14 Fungsi OnSegment

4.2.15 Implementasi Fungsi ConvexHull

Pada subbab ini akan dijelaskan mengenai implementasi dari fungsi ConvexHull pada pseudocode 3.15. Implementasi dapat dilihat pada kode sumber 4.15.

```

1 polygon convexHull(vector<point> &pts){
2   sort(pts.begin(), pts.end());
3   vector<point> hull;
4   for (int i = 0; i < 2; i++){
5     int start = (int)hull.size();
6     for (int i = 0; i < pts.size(); i++){
7       while ((int)hull.size() >= start + 2 &&
            orientation(hull[(int)hull.size() - 1],
            hull[(int)hull.size() - 2], pts[i]) <=
            0.0)
8         hull.pop();
9       hull.pb(pts[i]);
10    }
11    hull.pop_back();
12    reverse(pts.begin(), pts.end());
13  }
14  return polygon(hull);
15 }

```

Kode Sumber 4.15 Fungsi ConvexHull

4.2.16 Implementasi Fungsi InSimplePolygon

Pada subbab ini akan dijelaskan mengenai implementasi dari fungsi `inSimplePolygon` pada pseudocode 3.16. Implementasi dapat dilihat pada kode sumber 4.16.

```

1  int inSimplePolygon(point P, polygon &A){
2      int ret = 0;
3      FOR(i, A.P.size()){
4          if (P == A.P[i])
5              return 0;
6          int j = A.next(i);
7          if (onSegment(P, segment(A.P[i], A.P[j])))
8              return 0;
9          bool below = (A.P[i].y < P.y);
10         if (below != (A.P[j].y < P.y)){
11             double o = orientation(P, A.P[i], A.P[j]);
12             if (o == 0.0)
13                 return 0;
14             if (below == (o > 0.0))
15                 ret += below ? 1 : -1;
16         }
17     }
18     return ret == 0 ? 1 : -1;
19 }

```

Kode Sumber 4.16 Fungsi InSimplePolygon

4.2.17 Implementasi Fungsi GetBetween

Pada subbab ini akan dijelaskan mengenai implementasi dari fungsi `GetBetween` pada pseudocode 3.17. Implementasi dapat dilihat pada kode sumber 4.17.

```

1  vector<point> getBetween(polygon &triangle,
    queue<point> q, vector<point> trees){
2  vector<point> points,pts;
3  while (!q.empty()){
4      if (inSimplePolygon(q.front(), triangle) !=
        1){
5          points.pb(q.front());
6      }
7      q.pop();
8  }
9  for (int i = 0; i < trees.size(); i++){
10     if (inSimplePolygon(trees[i], triangle) != 1){
11         points.pb(trees[i]);
12     }
13 }
14 polygon P = convexHull(points);
15 int i = 0;
16 while (1){
17     if (P.P[i] == triangle.P[0]){
18         if (P.P[P.next(i)] == triangle.P[2]){
19             i = P.prev(i);
20             while (!(P.P[i] == triangle.P[2])){
21                 pts.pb(P.P[i]); i = P.prev(i);
22             }
23         }
24         else{
25             i = P.next(i);
26             while (!(P.P[i] == triangle.P[2])){
27                 pts.pb(P.P[i]); i = P.next(i);
28             }
29         }
30         break;
31     }
32     i = P.next(i);
33 }
34 return pts;
35 }

```

Kode Sumber 4.17 Fungsi GetBetween

4.2.18 Implementasi Fungsi Solve

Pada subbab ini akan dijelaskan mengenai implementasi dari fungsi Solve pada pseudocode 3.18. Implementasi dapat dilihat pada kode sumber 4.18.

```

1 polygon solve(polygon &perimeter, vector<point>
    &trees){
2     queue<point> q;
3     for (int i = 0; i < perimeter.P.size(); i++){
4         q.push(perimeter.P[i]);
5     }
6
7     point bef = perimeter.P[perimeter.P.size() - 1];
8     while (1){
9         bool erased = 0;
10        int count = q.size();
11        while (count--){
12            point cur = q.front();
13            q.pop();
14            pool[cur]--;
15            if (!cur.fixed && (!find(q, cur) || cur ==
                bef || cur == q.front()) &&
                orientation(cur, bef, q.front()) <=
                0.0){
16                erased = true;
17                polygon triangle;
18                triangle.P.pb(bef);
19                triangle.P.pb(cur);
20                triangle.P.pb(q.front());
21                vector<point> points =
                    getBetween(triangle, q, trees);
22                for (int i = 0; i < points.size(); i++){
23                    q.push(points[i]);
24                    pool[points[i]]++;
25                    bef = points[i];
26                }
27            }
28            else{
29                q.push(cur);
30                pool[cur]++;

```



```
31         bef = cur;
32     }
33 }
34     if (!erased)
35         break;
36 }
37
38     vector<point> hull;
39     while (!q.empty()){
40         hull.pb(q.front());
41         q.pop();
42     }
43     return polygon(hull);
44 }
```

Kode Sumber 4.18 Fungsi Solve

[Halaman ini sengaja dikosongkan]

BAB V

UJI COBA DAN EVALUASI

Pada bab ini dijelaskan tentang uji coba dan evaluasi dari implementasi yang telah dilakukan pada tugas akhir ini.

5.1 Lingkungan Uji Coba

Lingkungan uji coba digunakan untuk uji coba kebenaran adalah salah satu sistem yang digunakan situs penilaian daring Sphere Online Judge, yaitu kluster *Cube* dengan spesifikasi sebagai berikut :

1. Perangkat Keras
 - (a) Processor Intel Xeon E3-1220 v5 (5 CPUs)
 - (b) Random Access Memory 1536MB
2. Perangkat Lunak
 - (a) Kompiler GCC 6.3.0

Lingkungan uji coba yang digunakan untuk melakukan uji coba kinerja menggunakan komputer pribadi penulis yang memiliki spesifikasi sebagai berikut

1. Perangkat Keras
 - (a) Processor Intel® Core™ i7-6500U CPU @ 2.50GHz (4 CPUs), 2.6GHz
 - (b) Random Access Memory 8192MB
2. Perangkat Lunak
 - (a) Sistem Operasi Windows 10 Home Single Language 64-bit
 - (b) Visual Studio Code
 - (c) Bahasa Pemrograman C++

- (d) Kompiler GCC 7.4.0 (Ubuntu 7.4.0-1ubuntu1 18.04.1)
untuk Windows Subsystem Linux

5.2 Skenario Uji Coba

Subbab ini akan menjelaskan hasil pengujian program untuk menyelesaikan permasalahan LL and ErBao. Metode pengujian yang dilakukan adalah sebagai berikut.

1. Pengujian luar. Pengujian ini menggunakan Online Judge untuk menguji kebenaran dan kinerja program.
2. Pengujian lokal. Pengujian ini menggunakan mesin yang digunakan dalam pengembangan untuk mengukur kinerja program.

Dalam pengujian lokal, dibuat beberapa kasus uji berdasarkan batasan yang ada pada soal.

Untuk pengujian luar, uji coba dilakukan dengan mengirimkan kode program dengan algoritma reduksi polygon ke situs penilaian daring Sphere Online Judge sebanyak 10 kali.

5.3 Uji Coba Kebenaran

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



Gambar 5.1 Hasil Uji Coba Kebenaran Situs Penilaian Sphere Online Judge

5.4 Uji Coba Kinerja Lokal

Pada subbab ini akan ditampilkan hasil uji coba kinerja dari algoritma reduksi polygon untuk menyelesaikan permasalahan LL and ErBao. Pengujian dilakukan terhadap kelompok masukan yang telah dijelaskan pada subbab 5.2. Detil masukan dapat dilihat pada Lampiran A. Langkah pengujian kinerja dilakukan dengan langkah sebagai berikut:

1. Rekam waktu tepat sebelum komputasi penyelesaian masalah.
2. Melakukan komputasi penyelesaian masalah untuk masukan kasus uji yang sama sebanyak 10 kali secara berturut-turut.
3. Rekam waktu tepat setelah komputasi dengan mengurangi waktu selesai komputasi dengan waktu sebelum komputasi.
4. Ulangi untuk semua kasus uji.

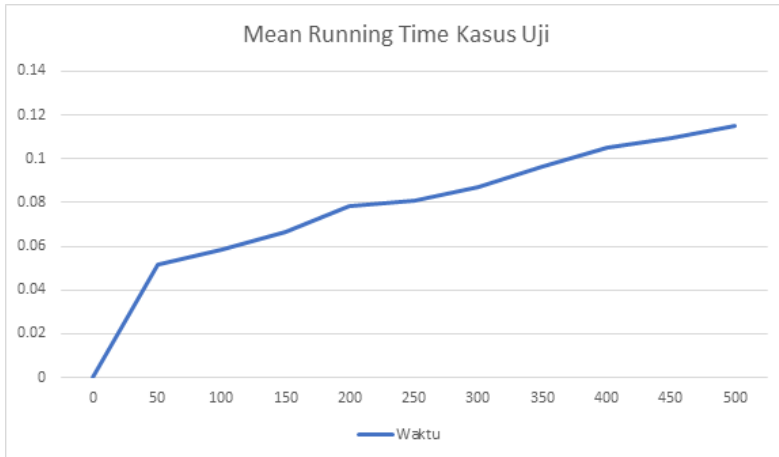
Grafik 5.2 menampilkan rata-rata kinerja masing-masing metode. Pada grafik tersebut dapat dilihat bahwa rata-rata *running time* reduksi polygon dengan algoritma *Melkman convex hull* berbanding lurus dengan banyaknya titik di dalam polygon tersebut.

5.5 Evaluasi Kebenaran Uji Coba Lokal

Evaluasi dilakukan dengan memeriksa hasil keluaran program apakah sama dengan contoh keluaran pada Sphere Online Judge 5637 LL and ErBao. Tabel kebenaran dapat dilihat pada gambar 5.1

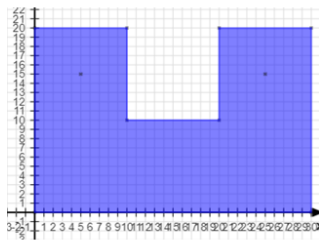
Gambar 5.3 menunjukkan ilustrasi kondisi awal program sebelum iterasi dimulai.

Pada awal iterasi ke-1, memeriksa titik $(0, 0)$. Titik tersebut akan dibuang karena titik tersebut merupakan titik luar dan orientasi terhadap titik sebelumnya dan sesudahnya membentuk *convex*. Sebelum membuang titik tersebut, program membuat segitiga dengan titik sebelumnya, titik tersebut dan titik sesudahnya. Kemudian pro-



Gambar 5.2 Grafik Mean Running Time Kasus Uji

Contoh testcase 1



Queue: (0,0), (30,0), (30,20), (20,20), (20,10), (10,10), (10,20), (0,20)

Gambar 5.3 Ilustrasi Kondisi Awal

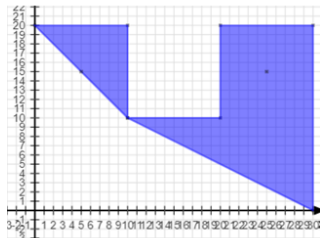
gram mencari semua titik yang berada di dalam segitiga tersebut. Selanjutnya program mencari *convex hull* dari semua titik yang didapatkan dan disisipkan ke queue polygon luar untuk menggantikan

Data Masukan	Hasil Keluaran
8 2	Case #1: 48.284
0 0	
30 0	
30 20	
20 20	
20 10	
10 10	
10 20	
0 20	
5 15	
25 15	

Tabel 5.1 Tabel Data Uji Coba Kebenaran Lokal dengan Sampel Data

titik yang dibuang. Kondisi setelah iterasi ke-1 dapat dilihat pada gambar 5.4.

Cek titik(0,0)

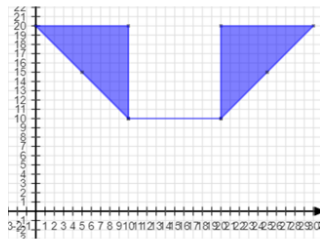


Queue: (30,0), (30,20), (20,20), (20,10), (10,10), (10,20), (0,20), (10,10)

Gambar 5.4 Ilustrasi Iterasi 1

Pada awal iterasi ke-2, memeriksa titik $(30, 0)$. Titik tersebut akan dibuang karena titik tersebut merupakan titik luar dan orientasi terhadap titik sebelumnya dan sesudahnya membentuk *convex*. Sebelum membuang titik tersebut, program membuat segitiga dengan titik sebelumnya, titik tersebut dan titik sesudahnya. Kemudian program mencari semua titik yang berada di dalam segitiga tersebut. Selanjutnya program mencari *convex hull* dari semua titik yang didapatkan dan disisipkan ke queue polygon luar untuk menggantikan titik yang dibuang. Kondisi setelah iterasi ke-2 dapat dilihat pada gambar 5.5.

Cek titik(30,0)



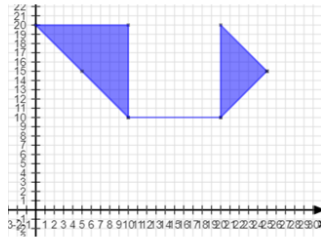
Queue: $(30,20)$, $(20,20)$, $(20,10)$, $(10,10)$, $(10,20)$, $(0,20)$, $(10,10)$, $(20,10)$

Gambar 5.5 Ilustrasi Iterasi 2

Pada awal iterasi ke-3, memeriksa titik $(30, 20)$. Titik tersebut akan dibuang karena titik tersebut merupakan titik luar dan orientasi terhadap titik sebelumnya dan sesudahnya membentuk *convex*. Sebelum membuang titik tersebut, program membuat segitiga dengan titik sebelumnya, titik tersebut dan titik sesudahnya. Kemudian program mencari semua titik yang berada di dalam segitiga tersebut. Selanjutnya program mencari *convex hull* dari semua titik yang didapatkan dan disisipkan ke queue polygon luar untuk meng-

gantikan titik yang dibuang. Kondisi setelah iterasi ke-3 dapat dilihat pada gambar 5.6.

Cek titik(30,20)



Queue : (20,20), (20,10), (10,10), (10,20), (0,20), (10,10), (20,10), (25,15)

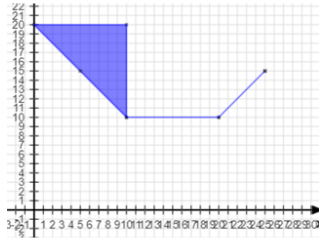
Gambar 5.6 Ilustrasi Iterasi 3

Pada awal iterasi ke-4, memeriksa titik (20, 20). Titik tersebut akan dibuang karena titik tersebut merupakan titik luar dan orientasi terhadap titik sebelumnya dan sesudahnya membentuk *convex*. Sebelum membuang titik tersebut, program membuat segitiga dengan titik sebelumnya, titik tersebut dan titik sesudahnya. Kemudian program mencari semua titik yang berada di dalam segitiga tersebut. Selanjutnya program mencari *convex hull* dari semua titik yang didapatkan dan disisipkan ke queue polygon luar untuk menggantikan titik yang dibuang. Kondisi setelah iterasi ke-4 dapat dilihat pada gambar 5.7.

Pada awal iterasi ke-5, memeriksa titik (20, 10). Titik tersebut tidak akan dibuang karena titik tersebut merupakan titik luar tetapi orientasi terhadap titik sebelumnya dan sesudahnya membentuk *concave*. Kondisi setelah iterasi ke-5 dapat dilihat pada gambar 5.8.

Pada awal iterasi ke-6, memeriksa titik (10, 10). Titik ter-

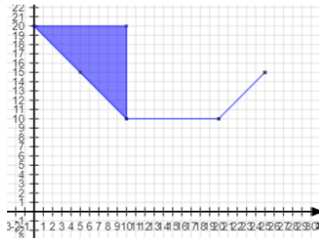
Cek titik(20,20)



Queue: (20,10), (10,10), (10,20), (0,20), (10,10), (20,10), (25,15)

Gambar 5.7 Ilustrasi Iterasi 4

Cek titik(20,10)

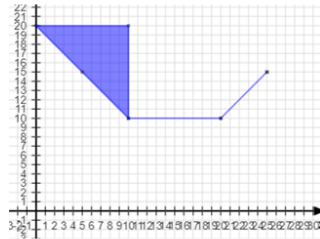


Queue: (10,10), (10,20), (0,20), (10,10), (20,10), (25,15), (20,10)

Gambar 5.8 Ilustrasi Iterasi 5

sebut tidak akan dibuang karena titik tersebut merupakan titik luar tetapi orientasi terhadap titik sebelumnya dan sesudahnya membentuk *concave*. Kondisi setelah iterasi ke-6 dapat dilihat pada gambar 5.9.

Cek titik(10,10)



Queue : (10,20), (0,20), (10,10), (20,10), (25,15), (20,10), (10,10)

Gambar 5.9 Ilustrasi Iterasi 6

Pada awal iterasi ke-7, memeriksa titik (10, 20). Titik tersebut akan dibuang karena titik tersebut merupakan titik luar dan orientasi terhadap titik sebelumnya dan sesudahnya membentuk *convex*. Sebelum membuang titik tersebut, program membuat segitiga dengan titik sebelumnya, titik tersebut dan titik sesudahnya. Kemudian program mencari semua titik yang berada di dalam segitiga tersebut. Selanjutnya program mencari *convex hull* dari semua titik yang didapatkan dan disisipkan ke queue polygon luar untuk menggantikan titik yang dibuang. Kondisi setelah iterasi ke-7 dapat dilihat pada gambar 5.10.

Pada awal iterasi ke-8, memeriksa titik (0, 20). Titik tersebut akan dibuang karena titik tersebut merupakan titik luar dan orientasi terhadap titik sebelumnya dan sesudahnya membentuk *convex*. Sebelum membuang titik tersebut, program membuat segitiga dengan titik sebelumnya, titik tersebut dan titik sesudahnya. Kemudian program mencari semua titik yang berada di dalam segitiga tersebut. Selanjutnya program mencari *convex hull* dari semua titik yang didapatkan dan disisipkan ke queue polygon luar untuk menggantikan

Cek titik(10,20)

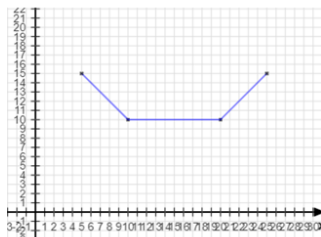


Queue: (0,20), (10,10), (20,10), (25,15), (20,10), (10,10)

Gambar 5.10 Ilustrasi Iterasi 7

titik yang dibuang. Kondisi setelah iterasi ke-8 dapat dilihat pada gambar 5.11.

Cek titik(0,20)

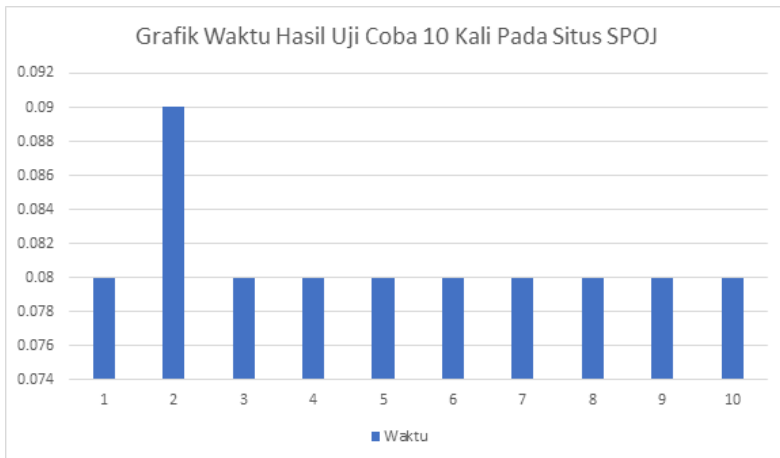


Queue: (10,10), (20,10), (25,15), (20,10), (10,10), (5,15)

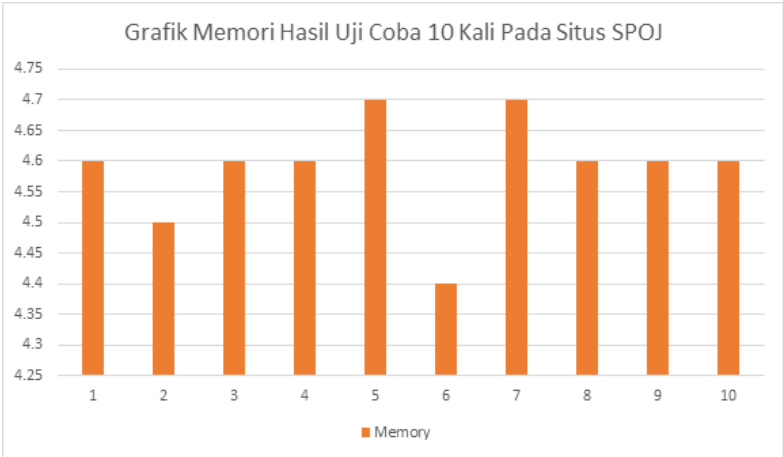
Gambar 5.11 Ilustrasi Iterasi 8

5.6 Uji Coba Kinerja Luar

Pada subbab ini akan ditampilkan hasil uji coba kinerja dari algoritma reduksi polygon. Pengujian dilakukan dengan cara mengirimkan kode program ke situs penilaian daring Sphere Online Judge. Detil mengenai hasil uji kinerja dapat dilihat pada Lampiran B. Rata-rata hasil pengumpulan kode berkas dengan algoritma *Melkman Convex Hull* adalah 0.08 detik dengan memori 4.6 MB. Hasil uji coba pada situs Sphere Online Judge dapat dilihat pada gambar 5.12 dan 5.13.



Gambar 5.12 Grafik Waktu Uji Coba 10 Kali pada Situs SPOJ



Gambar 5.13 Grafik Memori Uji Coba 10 Kali pada Situs SPOJ

BAB VI

KESIMPULAN

Pada bab ini dijelaskan mengenai 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 LL and Erbao.

1. Permasalahan LL and ErBao dapat diselesaikan dengan melakukan reduksi polygon luar terhadap titik di dalamnya.
2. Permasalahan LL and Erbao dapat diselesaikan dengan batasan pada soal dapat diselesaikan dengan reduksi polygon dengan waktu minimum 0.08 detik, waktu maksimum 0.08 detik, dan memori minimum 4.4 MB, memori maksimum 4.7 MB.
3. Algoritma *Melkman convex hull* terbukti efektif untuk melakukan reduksi polygon untuk mencari *relative convex polygon*.

6.2 Saran

Pada Tugas Akhir kali ini tentunya terdapat kekurangan serta nilai-nilai yang dapat penulis ambil. Berikut adalah saran-saran yang dapat diambil melalui Tugas Akhir ini:

1. Untuk kedepannya, algoritma pada Tugas Akhir ini dapat menjadi bahan riset untuk mencari optimasi lebih lanjut.
2. Metode reduksi polygon dengan menggunakan algoritma

Melkman convex hull yang dimodifikasi dapat digunakan untuk mencari *relative convex hull* dengan polygon yang membatasi segmen garis ataupun polygon sederhana.

DAFTAR PUSTAKA

- [1] SPOJ. (2009). LL and ErBao, **url:** <https://www.spoj.com/problems/ISUN1/>.
- [2] A. A. Melkman, “On-line construction of the convex hull of a simple polyline”, *Information Processing Letters* 25, **pages** 11–12, 1987.
- [3] A. Andrew., “Another Efficient Algorithm for Convex Hulls in Two Dimensions”, *Information Processing Letters* 9, **pages** 216–219, 1979.
- [4] geeksforgeeks. (2019). How to check if a given point lies inside or outside a polygon, **url:** <https://www.geeksforgeeks.org/how-to-check-if-a-given-point-lies-inside-a-polygon/>.

[Halaman ini sengaja dikosongkan]

LAMPIRAN A: DATA UJI

Data Masukan					
-10000 -10000	-9840 -10000	-9680 -10000	-9520 -10000	-9360 -10000	-9200 -10000
-9040 -10000	-8880 -10000	-8720 -10000	-8560 -10000	-8400 -10000	-8240 -10000
-8080 -10000	-7920 -10000	-7760 -10000	-7600 -10000	-7440 -10000	-7280 -10000
-7120 -10000	-6960 -10000	-6800 -10000	-6640 -10000	-6480 -10000	-6320 -10000
-6160 -10000	-6000 -10000	-5840 -10000	-5680 -10000	-5520 -10000	-5360 -10000
-5200 -10000	-5040 -10000	-4880 -10000	-4720 -10000	-4560 -10000	-4400 -10000
-4240 -10000	-4080 -10000	-3920 -10000	-3760 -10000	-3600 -10000	-3440 -10000
-3280 -10000	-3120 -10000	-2960 -10000	-2800 -10000	-2640 -10000	-2480 -10000
-2320 -10000	-2160 -10000	-2000 -10000	-1840 -10000	-1680 -10000	-1520 -10000
-1360 -10000	-1200 -10000	-1040 -10000	-880 -10000	-720 -10000	-560 -10000
-400 -10000	-240 -10000	-80 -10000	80 -10000	240 -10000	400 -10000
560 -10000	720 -10000	880 -10000	1040 -10000	1200 -10000	1360 -10000
1520 -10000	1680 -10000	1840 -10000	2000 -10000	2160 -10000	2320 -10000
2480 -10000	2640 -10000	2800 -10000	2960 -10000	3120 -10000	3280 -10000
3440 -10000	3600 -10000	3760 -10000	3920 -10000	4080 -10000	4240 -10000
4400 -10000	4560 -10000	4720 -10000	4880 -10000	5040 -10000	5200 -10000
5360 -10000	5520 -10000	5680 -10000	5840 -10000	6000 -10000	6160 -10000
6320 -10000	6480 -10000	6640 -10000	6800 -10000	6960 -10000	7120 -10000
7280 -10000	7440 -10000	7600 -10000	7760 -10000	7920 -10000	8080 -10000
8240 -10000	8400 -10000	8560 -10000	8720 -10000	8880 -10000	9040 -10000
9200 -10000	9360 -10000	9520 -10000	9680 -10000	9840 -10000	10000 -10000
10000 -9840	10000 -9680	10000 -9520	10000 -9360	10000 -9200	10000 -9040
10000 -8880	10000 -8720	10000 -8560	10000 -8400	10000 -8240	10000 -8080
10000 -7920	10000 -7760	10000 -7600	10000 -7440	10000 -7280	10000 -7120

Data Masukan									
10000 -6960	10000 -6800	10000 -6640	10000 -6480	10000 -6320	10000 -6160				
10000 -6000	10000 -5840	10000 -5680	10000 -5520	10000 -5360	10000 -5200				
10000 -5040	10000 -4880	10000 -4720	10000 -4560	10000 -4400	10000 -4240				
10000 -4080	10000 -3920	10000 -3760	10000 -3600	10000 -3440	10000 -3280				
10000 -3120	10000 -2960	10000 -2800	10000 -2640	10000 -2480	10000 -2320				
10000 -2160	10000 -2000	10000 -1840	10000 -1680	10000 -1520	10000 -1360				
10000 -1200	10000 -1040	10000 -880	10000 -720	10000 -560	10000 -400				
10000 -240	10000 -80	10000 80	10000 240	10000 400	10000 560				
10000 720	10000 880	10000 1040	10000 1200	10000 1360	10000 1520				
10000 1680	10000 1840	10000 2000	10000 2160	10000 2320	10000 2480				
10000 2640	10000 2800	10000 2960	10000 3120	10000 3280	10000 3440				
10000 3600	10000 3760	10000 3920	10000 4080	10000 4240	10000 4400				
10000 4560	10000 4720	10000 4880	10000 5040	10000 5200	10000 5360				
10000 5520	10000 5680	10000 5840	10000 6000	10000 6160	10000 6320				
10000 6480	10000 6640	10000 6800	10000 6960	10000 7120	10000 7280				
10000 7440	10000 7600	10000 7760	10000 7920	10000 8080	10000 8240				
10000 8400	10000 8560	10000 8720	10000 8880	10000 9040	10000 9200				
10000 9360	10000 9520	10000 9680	10000 9840	10000 10000	9840 10000				
9680 10000	9520 10000	9360 10000	9200 10000	9040 10000	8880 10000				
8720 10000	8560 10000	8400 10000	8240 10000	8080 10000	7920 10000				
7760 10000	7600 10000	7440 10000	7280 10000	7120 10000	6960 10000				
6800 10000	6640 10000	6480 10000	6320 10000	6160 10000	6000 10000				
5840 10000	5680 10000	5520 10000	5360 10000	5200 10000	5040 10000				
4880 10000	4720 10000	4560 10000	4400 10000	4240 10000	4080 10000				

Data Masukan									
3920 10000	3760 10000	3600 10000	3440 10000	3280 10000	3120 10000				
2960 10000	2800 10000	2640 10000	2480 10000	2320 10000	2160 10000				
2000 10000	1840 10000	1680 10000	1520 10000	1360 10000	1200 10000				
1040 10000	880 10000	720 10000	560 10000	400 10000	240 10000				
80 10000	-80 10000	-240 10000	-400 10000	-560 10000	-720 10000				
-880 10000	-1040 10000	-1200 10000	-1360 10000	-1520 10000	-1680 10000				
-1840 10000	-2000 10000	-2160 10000	-2320 10000	-2480 10000	-2640 10000				
-2800 10000	-2960 10000	-3120 10000	-3280 10000	-3440 10000	-3600 10000				
-3760 10000	-3920 10000	-4080 10000	-4240 10000	-4400 10000	-4560 10000				
-4720 10000	-4880 10000	-5040 10000	-5200 10000	-5360 10000	-5520 10000				
-5680 10000	-5840 10000	-6000 10000	-6160 10000	-6320 10000	-6480 10000				
-6640 10000	-6800 10000	-6960 10000	-7120 10000	-7280 10000	-7440 10000				
-7600 10000	-7760 10000	-7920 10000	-8080 10000	-8240 10000	-8400 10000				
-8560 10000	-8720 10000	-8880 10000	-9040 10000	-9200 10000	-9360 10000				
-9520 10000	-9680 10000	-9840 10000	-10000 10000	-10000 9840	-10000 9680				
-10000 9520	-10000 9360	-10000 9200	-10000 9040	-10000 8880	-10000 8720				
-10000 8560	-10000 8400	-10000 8240	-10000 8080	-10000 7920	-10000 7760				
-10000 7600	-10000 7440	-10000 7280	-10000 7120	-10000 6960	-10000 6800				
-10000 6640	-10000 6480	-10000 6320	-10000 6160	-10000 6000	-10000 5840				
-10000 5680	-10000 5520	-10000 5360	-10000 5200	-10000 5040	-10000 4880				
-10000 4720	-10000 4560	-10000 4400	-10000 4240	-10000 4080	-10000 3920				
-10000 3760	-10000 3600	-10000 3440	-10000 3280	-10000 3120	-10000 2960				
-10000 2800	-10000 2640	-10000 2480	-10000 2320	-10000 2160	-10000 2000				
-10000 1840	-10000 1680	-10000 1520	-10000 1360	-10000 1200	-10000 1040				

Data Masukan									
-10000 880	-10000 720	-10000 560	-10000 400	-10000 240	-10000 80				
-10000 -80	-10000 -240	-10000 -400	-10000 -560	-10000 -720	-10000 -880				
-10000 -1040	-10000 -1200	-10000 -1360	-10000 -1520	-10000 -1680	-10000 -1840				
-10000 -2000	-10000 -2160	-10000 -2320	-10000 -2480	-10000 -2640	-10000 -2800				
-10000 -2960	-10000 -3120	-10000 -3280	-10000 -3440	-10000 -3600	-10000 -3760				
-10000 -3920	-10000 -4080	-10000 -4240	-10000 -4400	-10000 -4560	-10000 -4720				
-10000 -4880	-10000 -5040	-10000 -5200	-10000 -5360	-10000 -5520	-10000 -5680				
-10000 -5840	-10000 -6000	-10000 -6160	-10000 -6320	-10000 -6480	-10000 -6640				
-10000 -6800	-10000 -6960	-10000 -7120	-10000 -7280	-10000 -7440	-10000 -7600				
-10000 -7760	-10000 -7920	-10000 -8080	-10000 -8240	-10000 -8400	-10000 -8560				
-10000 -8720	-10000 -8880	-10000 -9040	-10000 -9200	-10000 -9360	-10000 -9520				
-10000 -9680	-10000 -9840	-10000 -9840	-10000 -9840	-10000 -9840	-10000 -9840				

No	Data Masukan					Hasil Keluaran
1	7297 -5179	-2569 -3179	4094 3274	719 -1563	-6680 348	PASS
	8612 8164	4715 6559	7379 4515	-1673 453	4384 1550	
	-9492 2456	8566 -7924	-5067 8717	-348 -6272	-9614 7411	
	-7094 2912	-1607 -8546	-8967 3377	7968 -6804	4502 2445	
	-522 -5220	-9762 -2864	-9553 -7675	-2831 9356	-5142 -8928	
	1322 -7910	3468 5862	-3159 -7017	-6292 2816	-6690 9999	
	-4586 2554	-8768 -5072	8616 1655	122 -6517	-5057 366	
	427 9209	-1959 -8021	-9683 2936	-2060 1045	-5261 -710	
	1589 -3109	-9898 8125	-8467 -3993	-6508 -803	-9344 -8748	
	848 -6711	2014 -7074	-8930 -5099	-2573 -7553	-5884 2478	
	1077 -4209	811 -5936	-9315 -722	3288 -7436	2351 1858	
	4669 -4219	-7102 -6866	443 -2655	-7890 6739	2139 549	
	-315 7509	6470 -3119	3990 -5881	4775 6120	4932 4153	
2	8799 1869	-7971 -8298	-7431 -2162	-697 2220	-2369 9155	PASS
	-2883 -4402	6095 5840	-2951 1527	-1431 -9392	-9263 -4014	
	9932 -5737	-2457 -2634	-1697 -2099	-6979 469	2322 7104	
	1693 -2996	3774 4563	-7785 189	-1638 1867	-8374 984	
	827 4990	-5251 -4969	2591 424	-9503 -1857	-4563 -6567	
	-8170 9456	2023 202	-9101 -6438	8055 -6072	-1446 -1076	
	-5018 -2154	4180 277	760 -1460	-5826 -5307	-6511 3749	
	2911 -7372	-457 -2377	-4228 -1306	2046 -6622	-4880 2668	
	7905 8624	1560 -3250	-1171 788	6027 5097	6690 -4013	
	401 8884	4796 -9111	-8909 1862	-9652 -8018	-6061 -7140	
	-3536 -9905	4246 -6408	-8870 -9661	-7387 6087	-4016 -446	

No	Data Masukan				Hasil Keluaran	
	7078 -2507 -9517 -6848 -926 728 -8255 -9737 -6326 -1839 -1241 6405	-3894 510 -8055 -336 3780 -2491 -8773 -436 -5274 3881 806 9132	-6814 -833 8336 -170 9876 -2145 4348 -4649 9697 -8594 44 -1512	-433 1790 -6673 9657 -8026 7471 -6041 754 -3225 1665 9233 -4679	-4181 -2535 -6677 9221 -1095 6075 -4925 -8434 -2355 9560 1922 1663	
3	8761 -4183 2045 -4180 -6398 -1990 -2916 696 -628 -8789 1598 4903 -2602 -4231 6830 -2630 6696 976 -5268 -167 -357 -3326 -8815 317 9279 1758 -4774 1424 -6794 -2216 1890 -8680 -2652 5900 1192 -5551	-3291 -6862 5072 8708 -1903 -3155 -1599 -6654 5451 -7693 9716 -3150 7478 -8387 -9463 2242 8612 -6592 -9113 -3551 1880 -6625 8609 -5267 -3535 -8477 -4104 -3317 8642 8839 8425 7739 5854 -7529 4351 -6430	2098 3491 -153 1119 -7243 6177 -6929 4331 4471 5455 -7315 -3346 8605 -6284 6356 -3555 -404 -5369 -4477 -2382 8685 -1973 293 1511 -6225 -832 -5094 3519 -4111 -9544 8664 -2943 1025 2247 -5845 -6606	-7363 -2537 1925 7036 -8770 -7995 -800 -2393 3581 -1871 9805 -2696 -1445 -6206 7386 -8249 3972 -9381 -2217 -378 -7519 -3665 -301 -4734 -2835 382 257 2127 -9169 4801 -1735 -7201 8797 -6179 9510 -8169	-5936 -9889 3217 2863 -5322 7841 5234 -9979 -2558 5304 -8931 -5124 2464 7012 9363 9807 -7305 8512 -941 -3806 1826 7089 3677 -2437 5368 -3039 6299 5939 -8878 7787 2877 2120 -8980 -394 -5307 -9739	PASS

No	Data Masukan				Hasil Keluaran
	-2098 -6423	1529 -2662	30 -6882	-8762 2174	-501 -3128
	7034 -8872	2440 2577	-2751 5852	-4997 6460	-732 3620
	7767 4265	5297 7983	9957 -9000	-1483 -216	4227 5747
	-2735 108	-9583 -858	-1652 -3927	2247 -8401	7781 -4928
	-2738 -3969	-8693 -3753	8681 -2603	6641 -8823	-3088 2002
	2107 27	-7224 -8594	-9569 4055	2127 -2718	-5719 -5646
	-4176 7828	379 2415	1986 1640	-6303 -406	-5747 7322
	-8900 880	-4022 7278	-9798 6859	-402 3972	6002 6053
	-3992 -557	-1303 -3766	2123 -2164	3160 2640	-9627 -791
	-9331 -4941	-4423 -4799	7853 486	-2199 -7720	-3773 9450
	-5451 -8543	-4014 -5580	-9754 -35	9358 -7106	-254 2947
	-9012 6712	6187 -5532	3256 -1255	5668 -9201	966 641
	2529 -5807	-9526 -9908	-6042 -4144	7994 7122	-156 -9767
	52 -7491	-3046 -1832	-3902 2969	-4144 -2977	9538 -542
	212 938	-834 2748	-1497 2754	5854 219	6228 7622
	9024 -5144	-8706 -6363	-6810 -3866	8325 -2414	-3665 3346
	2240 5395	-7893 -8310	5995 -5672	5259 -9587	4328 9909
	-6411 -948	-4386 -7736	-2049 855	7111 9371	-900 -6500
	-7734 368	-8109 -5553	-156 -1640	-4401 -4471	-5813 -6682
	-7426 725	2326 -2748	5491 1703	-3816 -5595	-8211 -2576
	-2069 -6884	889 -7184	-6667 2091	-4258 6219	7110 -9315
	-2064 608	2591 -4424	-1242 -2346	-7559 -2385	-2445 -5257
	-1915 628	6830 9538	9264 3637	7956 3259	1352 -7738
	-2192 7302	6150 -7683	-3013 7732	-1149 1532	2079 -9412

No	Data Masukan				Hasil Keluaran
4	-4078 -4147	-738 -6964	-243 -1353	-6795 -1068	3843 7899
	2028 -2667	-2272 6383	2524 -4662	2014 -5435	-3447 722
	6900 -4261	-2792 8004	2532 -8062	947 748	-3049 -9138
	9382 -3485	7679 2457	-7268 -6755	958 738	-4466 -3547
	1518 4821	-3104 -1690	5239 9458	810 3724	1829 5445
	-553 4300	-9892 -4034	1100 -6850	-1876 -6980	-8757 4851
	-381 7275	-8426 -541	-2624 -4528	-9833 1704	-7793 8098
	-898 -5301	4688 9948	-52 -2853	2459 9488	-1457 -5978
	-8319 2430	6064 8866	3349 4965	-1110 8522	-3573 9753
	-6179 5040	7264 9046	4354 -2927	3419 -3661	-322 9092
	3666 1563	-9604 7799	2767 -1127	-779 -4260	-4461 7154
	1940 8736	-8868 7831	1146 2236	-7516 -4999	-9637 -2513
	9739 -9023	-7215 -3988	8119 985	-9038 5617	-9896 -5845
	-8774 -7591	-5873 -555	2609 3729	242 -5323	423 -5649
	-5207 4159	9570 -5951	-5251 -4167	5404 -7707	-5228 -635
	421 -7368	1971 1243	6513 -7472	110 1691	-1938 -498
	576 960	3508 -5746	-7920 2778	-414 -7550	-5088 5079
	1830 -4994	230 3232	-2113 -1869	3182 -9205	8286 -9182
	2698 -2099	-6447 -5943	-7957 -8561	9347 -5052	-7440 -1979
	-8099 7887	-7015 666	9491 -8244	-4291 -8080	-8006 -6864
	-7644 2439	-611 -3602	1550 8125	8468 5598	-9612 1718
	745 -3108	142 -1287	-7865 7247	-2276 -4601	-2864 1996
	2786 -9379	-8919 3612	4334 -5408	-1891 -6820	7358 -4512
	1153 1412	4229 -9008	-3918 -1014	-7380 -9155	2472 3048

PASS

No	Data Masukan				Hasil Keluaran
	3640 2240	9702 -3108	-4852 5398	-3569 5197	-8210 -1176
	-1 -5835	-6034 -4369	-2375 1501	-605 1467	7037 6854
	-935 -9647	-6501 2117	-4503 3446	-9136 1595	-8536 2728
	-8285 -6471	-2014 -2123	-3652 -6137	3752 871	6288 -2644
	-9665 -7729	8968 -4248	9979 9443	-1855 -4692	-9248 -5969
	4697 9838	2607 -3247	-3976 -3800	-6569 -8111	-7601 -1705
	79 7182	-6933 10	-5947 3869	-131 -5824	-7493 1613
	-227 4075	9264 1167	-4397 -1032	-964 -9865	-5166 9969
	-7872 8957	-7000 -5471	1725 -9465	5845 -3687	-6619 9802
	-6889 2738	-7537 -3795	5167 5269	-9393 7121	4256 7320
	-1115 -9939	-6703 9066	-7852 -1505	-1502 -8943	-9899 643
	4627 6385	1708 -7143	6694 -6961	-9188 9750	2827 8151
	-2312 -8012	-2378 6395	-6470 -1561	-3498 -8963	2583 -4103
	-6015 -8593	2579 -2674	-9996 -8551	1136 -5785	-7493 -4101
	-1652 -9874	-7189 -1709	5165 -1936	887 -7295	-5420 -9978
	-7006 -6998	-5191 -5894	3956 -7911	-7048 -3065	2024 -115
	-1655 4608	-9044 9103	4718 371	1963 1569	-9008 2447
	-2875 4192	-6517 1059	-1534 2626	2921 -6168	9798 -8481
	3341 5827	-2139 -806	9649 8741	311 2229	-1229 -6414
	6639 1418	-7927 -9877	7240 957	-737 -1593	5666 -5333
	2298 7005	-9045 -9946	-3654 1992	-9374 3362	-919 -887
	-9416 -174	-5297 -7823	-1737 -1746	1862 -1831	-472 -8424
	4623 -9166	7345 -4321	3331 709	-452 7109	5772 9490
	2461 -8675	1013 -4134	-4460 -7275	-6626 2020	-4049 444

No	Data Masukan				Hasil Keluaran
5	4579 -4104	-2828 -4106	-1964 1664	7919 -6141	2221 -6826
	-6446 -8921	-8553 -8336	-3371 9748	-804 7092	2001 -1653
	3871 -7782	9189 -4674	7851 -6065	9831 -1737	5539 -1982
	5526 9787	1878 360	1323 -2575	-5266 -4092	946 1381
	8204 -8666	2809 -7027	-8688 -2097	-4761 -563	-2339 -8609
	-6643 -2182	-1217 -8728	-9053 -6341	-31 -4189	-8116 -9388
	4942 -7063	2524 -5661	-9966 6103	-3971 -6995	1083 -8423
	-1045 -9512	9419 -2964	-5946 146	-9178 861	3243 -3466
	-1297 -9041	-2981 -6217	1330 -6425	-7842 5879	-9021 256
	9933 -2864	3753 -4193	-1624 256	5171 -6324	-9583 -2112
	-267 -2980	-7721 5856	-2590 -9831	4997 -4018	2018 -3934
	-1838 -7328	3956 8080	-3352 -2370	-8541 -6630	-5056 -3778
	3435 -5672	4281 1774	-6207 -5906	-3131 -9575	7200 1979
	-3528 8491	-3152 -6280	-263 -9331	6 726	-8066 -6092
	-7917 781	-1191 -4766	3607 2132	-2676 36	-3401 242
	-5756 7452	-7601 375	33 9506	1896 -9980	7420 -4629
	9775 690	1254 -6868	-4324 -2032	-2038 411	-6341 -625
	-3041 7319	-2904 9981	3162 -4153	-644 7662	8095 869
	1223 -8226	-645 -1443	510 5436	8052 2290	-8584 -7582
	-3232 8036	1842 544	-6187 -3506	-4195 -3996	7394 -9838
	-6162 -5900	3324 -8647	-4901 -1095	-3046 9941	-6326 -4349
	-5177 905	-9344 5802	-6259 -5908	-6367 1777	5174 6890
	2617 -2737	-167 1355	-6380 -2257	-2610 -2186	-4906 2905
	-8473 -3334	-9463 -2933	-3907 -6843	-1325 -6678	5990 -4810

PASS

No	Data Masukan					Hasil Keluaran
	-9943 5823	-7411 -7687	-8537 -9190	-767 -9779	4191 3164	
	-8081 5293	401 5653	-8402 1575	1883 -4472	-6629 -6737	
	4980 567	-538 -2790	309 3645	-3222 4034	-2185 2423	
	-5567 881	-6973 -7834	-7079 8028	-3452 -5127	-7807 -7411	
	97 -7898	-9207 -9114	-5686 -8966	-119 9955	-1104 -6894	
	-3808 -4065	-7397 3240	-1343 9626	728 -9295	-3894 -7784	
	-5480 -9013	8005 1793	-1304 546	-930 -120	-2628 -8657	
	826 -7034	867 1370	-1968 7226	5758 4586	-4332 -2646	
	9222 -2640	-8451 -28	-694 -2943	8865 -7136	-1893 -9365	
	2747 -9175	-7931 737	-9482 -290	4256 6297	-3699 7956	
	8776 7255	6890 -2369	5606 3175	206 -5745	6650 -8464	
	7068 9345	-2798 4842	2478 7310	-3972 -4280	-922 -5113	
	2044 249	-3989 -7613	1441 -715	-8012 8791	-6955 1232	
	-353 -7704	-7889 -1612	-7035 2621	-5799 -1426	-7086 -4200	
	-1531 -1335	-6256 4807	-7553 9401	-5341 8438	-5331 1048	
	5243 9671	-2351 -7498	-3386 9125	-6065 4459	1153 1621	
	-2450 3921	1277 9165	-9665 -7409	-3082 -3799	-6815 -8902	
	-1852 -2948	-4921 -7976	1947 8645	-7339 6021	-8502 7363	
	-8405 -39	-8120 1417	-6942 9257	1393 -7946	-6790 5006	
	-3766 -4578	-1006 5763	1358 -2451	9819 8626	-4542 -8724	
	-469 943	9292 -6844	-2836 8585	5700 -9596	9821 -4715	
	-1145 9118	-7880 -4171	-3178 -8265	-9324 -324	-1419 -8482	
	8040 -8348	7062 4829	-5427 -9011	-8551 -4243	-9011 2	
	7540 -9074	-9991 4537	2191 -1417	1589 63	-7661 -2965	

No	Data Masukan				Hasil Keluaran	
6	9870 -3794	9428 1822	-2140 -1491	-8028 -5371	-4788 1678	PASS
	4637 -2157	-6011 -242	2849 -5996	-1555 -7593	-9422 4054	
	571 -6649	-448 -8425	4012 -3656	-4251 2201	40 -9641	
	4926 -8855	-9038 -9897	-8168 -5026	-8679 3295	6513 1863	
	-4719 131	-675 -57	2152 -6404	-6197 1821	-6030 7090	
	5411 2749	-5013 -2827	-6002 1167	-1996 -586	-4511 9294	
	-7453 -3821	7828 6765	-3165 6702	-7989 5413	-8081 3810	
	6276 -1400	-4283 -4075	-696 663	1691 -1263	126 -2473	
	2866 -3152	-1747 -3811	-6390 -1899	-4488 -1199	-6730 9827	
	2730 7763	4082 1832	8247 -3049	4049 4956	-4768 9079	
	-7685 -7864	-5195 9640	919 3591	9794 -8084	6183 -2927	
	-8987 -5878	-3792 151	-6310 3165	-379 -2196	581 -6232	
	7355 -925	-4856 -1295	6684 4192	-143 -688	4804 -7961	
	-1255 4797	-3931 6841	-1616 -7504	-6833 1408	6454 -8785	
	-1104 -6776	-2621 -4479	88 113	-2188 -4701	-8576 6211	
	-1953 -9893	5578 -2240	4675 -9486	-3448 6689	-5785 -4866	
	4610 7002	-2526 -3012	-9100 1982	-6054 -7799	-6409 -7702	
	-790 -3865	1731 -1635	-1340 -6846	-6095 -8283	-1792 -2952	
	-8412 7041	171 8307	-661 -6109	-767 -4517	-7649 8243	
	2468 7690	2292 -934	-6622 -2623	-5776 -7362	-9221 -2535	
	-3371 2125	-507 2620	-8189 -2538	-2289 2132	-4095 -7107	
	-1562 5700	-390 -7730	501 -1930	-6651 -1584	-4530 437	
	-8628 1566	880 3933	-8315 -3437	-2847 1582	7618 451	
	-2242 6217	4496 -4143	1939 -6091	-9573 -1794	9741 -5983	

No	Data Masukan					Hasil Keluaran
	-655 -6561	3567 -8626	-8088 -5101	1536 -7218	99 -4141	
	-8464 -2199	-1102 -5140	-3631 -7202	-9000 3529	-4537 -5775	
	2353 2547	7175 -7166	-7279 8638	-4550 -8388	-4206 -5157	
	-4924 -4881	-9206 564	7395 -2975	1849 8952	7065 -6655	
	5388 -6316	-146 5597	-1156 8419	-4561 -7564	-6537 -9658	
	2011 9358	6629 -2921	-1855 6442	-2621 251	-2334 -9833	
	3057 2645	-8069 2663	2899 -6652	-7542 3971	-302 2244	
	-2939 -8423	1095 9528	-3953 1914	-2415 -7510	2483 -2551	
	76 -9678	-6298 -9971	-7194 1525	434 7082	-6616 7888	
	497 -8687	4287 -7626	5342 -4142	-3916 6581	-4437 827	
	-7330 -1920	-1683 1276	-358 7368	4480 -9560	-7111 -8883	
	8710 -8349	3453 1427	-3823 -4041	-4291 -9448	-3839 -3221	
	-2098 614	-6477 686	1251 6884	-9005 3128	-8682 1393	
	9488 -5423	-9565 -6444	-4488 -7309	1233 -5875	-3453 464	
	4990 -5103	1517 -4693	-8782 723	2113 -4138	-5680 -9082	
	-198 -461	-295 -2332	-8340 -1543	7113 -1017	2463 2111	
	1184 -3681	-4258 -495	-5944 -7674	-1552 -7481	-7853 8591	
	-5915 1335	-8956 9918	2356 -2101	9843 8219	7393 -4881	
	-5436 -9161	-8216 -8836	3267 9971	-2505 -6070	8129 -6827	
	-3694 -8385	7881 -3616	9091 1828	2233 -6874	3286 -5420	
	-2410 361	4501 -6098	-2019 2020	2545 -5080	7134 -4786	
	-321 2763	-8251 -8930	-5401 5787	-6377 9227	-1130 -4138	
	7016 -7595	-6627 -1987	6817 -1176	-9996 -9351	930 3967	
	-4034 -454	-2058 104	-8588 -1287	9882 -6810	-3046 8553	

No	Data Masukan				Hasil Keluaran	
	-131 -6403	6037 3594	849 8396	-4287 1049	-8644 -3245	
	-3649 -4748	8562 -9347	619 1136	5409 737	-3880 6708	
	9515 -4924	-6507 -6588	24 1722	9678 -8836	-1620 -3592	
	-8646 -9663	-9095 -1340	-6416 -3792	4687 5736	765 -1629	
	-5769 2131	-4842 -244	-9816 -1258	-7128 -3073	-9038 -7372	
	-4907 -9253	8185 6139	-2558 -3459	-3897 3147	-9653 -8420	
	-3647 2877	-8805 -8456	-3270 2247	-9512 -664	-4698 1244	
	-5500 -2945	82 2708	-3996 -1465	-1486 -2806	-1033 -6317	
	9149 -3382	1138 -2218	-6412 8238	-5971 -5587	-4889 -6081	
	-3436 -2044	4188 -4801	-668 -2503	-6104 -1551	-1288 -6790	
	6720 -8304	4607 -5708	5108 1916	-6607 -3283	-3367 3455	
	-4219 -9647	168 2603	-8806 -1203	-1178 2148	-354 3162	
	-5020 -8641	-5807 2555	-6809 -7096	3262 9704	-2678 202	
	-5997 5099	-3357 -5366	3411 -375	-7196 -1802	-2681 -6298	
	-3164 4669	1430 -60	-4644 1022	1525 747	-8267 -1659	
	-2976 5741	-9694 6099	-4344 -8088	3884 5214	7889 -3506	
	-3541 -6247	-5163 -7291	-8007 692	7882 2186	-2197 -5625	
	-2900 -2374	-6597 -7860	372 -1134	-1119 1163	-3352 -9612	
	-8917 -1318	-2633 2989	-2547 -3452	-3899 -4236	-1232 -1719	
	2756 -8682	-2577 5894	-8221 -4751	1673 2967	-372 1756	
	-3501 2746	-2455 -7407	-2939 -5900	7238 -4093	2208 90	
	2109 4915	-2383 -5269	-9012 868	-3319 -39	2765 5352	
	-3205 -4815	-5185 -8956	-4670 -8806	5353 -1036	-1733 -2985	

No	Data Masukan					Hasil Keluaran
7	617 -3791	723 6919	1787 -3944	1812 -2258	1831 5422	PASS
	-9569 -9528	1952 7220	-2748 2514	2220 -4289	1300 -574	
	-4833 2644	-8292 -4209	-5942 510	-7160 -9545	-9982 1371	
	5608 -6582	6135 -4065	8428 -6172	1961 -4166	-4598 7770	
	-3656 -2977	-8817 -1642	1757 -514	-9688 -3802	5272 4594	
	-6295 1204	425 5280	5803 -4018	-5950 1436	-4716 -4816	
	-3843 -880	8382 -5790	-8598 -2534	-176 -5989	3629 -4525	
	-2735 5383	6919 -362	8480 -4978	1872 -8224	-2628 2262	
	4591 -234	-8695 -7392	6243 -538	244 1659	-6783 7087	
	854 3609	8324 4866	-4137 -7590	6431 -6358	2645 5521	
	-7753 -9846	71 -8507	-9777 5670	-4741 7961	-4445 2405	
	103 -1944	2234 -2828	61 -2637	949 2735	-8097 -9506	
	1933 2268	-8810 2167	-7427 -9192	1086 5272	-5732 8023	
	-4828 -5361	-1908 7687	1203 -5476	8473 -1613	-5367 9788	
	-5570 -9776	8434 9669	2240 2452	-9425 2607	-7062 -6815	
	-8555 -37	-4049 -4052	-9125 9650	6098 2459	3639 484	
	-57 -4438	-1374 -1699	-6791 3666	4603 -8363	-5203 -311	
	-5645 5473	-4281 171	-6685 -844	-436 -9061	-3613 -3231	
	-5497 9447	3323 -320	-1456 1256	-9166 -2797	617 -6893	
	-1396 -5527	6636 6442	-6664 -6973	-928 -7470	9705 7030	
	4308 -531	-7650 -2129	-3290 -577	6725 -4387	9444 9366	
	-2618 -2771	-7531 -3771	-5286 4592	2199 1015	-9540 6626	
	-1013 421	-5147 3468	-2193 2239	4273 -6775	-6609 -3503	
	-3036 -4666	-5900 -6519	-1580 9425	-5243 -5995	-5570 -1429	

No	Data Masukan					Hasil Keluaran
	5263 -5847	-3619 8190	-7973 6024	-7877 85	3084 -5022	
	2728 -698	-1407 -1220	2280 -5867	-4931 1457	-656 -1579	
	3306 -8905	-6604 -7310	-7816 -7292	-228 -6773	5580 -3983	
	-2313 7363	-4843 -8579	848 -1031	5668 -2604	4775 5815	
	-651 -175	-3688 1736	722 -5824	-960 -5363	312 -7703	
	-7198 -5037	-2279 5165	8740 2269	-5916 -7705	-4513 1960	
	-2271 -7426	-2921 -8242	-9055 -8549	-9879 -8758	-2987 2226	
	1218 -2594	-7170 1930	-1001 5554	-6043 -6855	-2925 -5348	
	812 -112	7371 -1849	-3323 854	-712 -2319	-990 4314	
	-5178 -1043	-7410 -3132	3143 1331	6425 -5783	-754 -9898	
	-3017 -5365	-7272 4753	109 -8170	-4018 264	-284 2766	
	9354 -7414	-1406 -951	-651 -8195	-5256 -4672	2362 -427	
	9852 -9470	-2795 -5627	-2856 2634	-1490 -9444	-8155 -8911	
	9607 229	2099 5894	-2864 -6279	1450 -3010	-4606 7732	
	-7753 -3802	-2901 -4416	-2337 3234	743 1465	-9609 -9874	
	3537 -972	-2986 -3126	-4359 9016	-7093 -3081	-4234 -7986	
	-564 -4100	-3 -6878	-5520 -3252	-6702 -1089	6333 1802	
	-4907 -9253	8185 6139	-2558 -3459	-3897 3147	-9653 -8420	
	-3647 2877	-8805 -8456	-3270 2247	-9512 -664	-4698 1244	
	-5500 -2945	82 2708	-3996 -1465	-1486 -2806	-1033 -6317	
	9149 -3382	1138 -2218	-6412 8238	-5971 -5587	-4889 -6081	
	-3436 -2044	4188 -4801	-668 -2503	-6104 -1551	-1288 -6790	
	6720 -8304	4607 -5708	5108 1916	-6607 -3283	-3367 3455	

No	Data Masukan					Hasil Keluaran
	-3694 -8385	7881 -3616	9091 1828	2233 -6874	3286 -5420	
	-2410 361	4501 -6098	-2019 2020	2545 -5080	7134 -4786	
	-321 2763	-8251 -8930	-5401 5787	-6377 9227	-1130 -4138	
	7016 -7595	-6627 -1987	6817 -1176	-9996 -9351	930 3967	
	-4034 -454	-2058 104	-8588 -1287	9882 -6810	-3046 8553	
	-131 -6403	6037 3594	849 8396	-4287 1049	-8644 -3245	
	-3649 -4748	8562 -9347	619 1136	5409 737	-3880 6708	
	9515 -4924	-6507 -6588	24 1722	9678 -8836	-1620 -3592	
	-8646 -9663	-9095 -1340	-6416 -3792	4687 5736	765 -1629	
	-5769 2131	-4842 -244	-9816 -1258	-7128 -3073	-9038 -7372	
	-4219 -9647	168 2603	-8806 -1203	-1178 2148	-354 3162	
	-5020 -8641	-5807 2555	-6809 -7096	3262 9704	-2678 202	
	-5997 5099	-3357 -5366	3411 -375	-7196 -1802	-2681 -6298	
	-3164 4669	1430 -60	-4644 1022	1525 747	-8267 -1659	
	-2976 5741	-9694 6099	-4344 -8088	3884 5214	7889 -3506	
	-3541 -6247	-5163 -7291	-8007 692	7882 2186	-2197 -5625	
	-2900 -2374	-6597 -7860	372 -1134	-1119 1163	-3352 -9612	
	-8917 -1318	-2633 2989	-2547 -3452	-3899 -4236	-1232 -1719	
	2756 -8682	-2577 5894	-8221 -4751	1673 2967	-372 1756	
	-3501 2746	-2455 -7407	-2939 -5900	7238 -4093	2208 90	
	2109 4915	-2383 -5269	-9012 868	-3319 -39	2765 5352	
	-3205 -4815	-5185 -8956	-4670 -8806	5353 -1036	-1733 -2985	
	-9569 -9528	1952 7220	-2748 2514	2220 -4289	1300 -574	
	-4833 2644	-8292 -4209	-5942 510	-7160 -9545	-9982 1371	

No	Data Masukan				Hasil Keluaran
8	617 -3791	723 6919	1787 -3944	1812 -2258	1831 5422
	-5436 -9161	-8216 -8836	3267 9971	-2505 -6070	8129 -6827
	5608 -6582	6135 -4065	8428 -6172	1961 -4166	-4598 7770
	-3656 -2977	-8817 -1642	1757 -514	-9688 -3802	5272 4594
	-6295 1204	425 5280	5803 -4018	-5950 1436	-4716 -4816
	-3843 -880	8382 -5790	-8598 -2534	-176 -5989	3629 -4525
	-2735 5383	6919 -362	8480 -4978	1872 -8224	-2628 2262
	4591 -234	-8695 -7392	6243 -538	244 1659	-6783 7087
	854 3609	8324 4866	-4137 -7590	6431 -6358	2645 5521
	-7753 -9846	71 -8507	-9777 5670	-4741 7961	-4445 2405
	103 -1944	2234 -2828	61 -2637	949 2735	-8097 -9506
	1933 2268	-8810 2167	-7427 -9192	1086 5272	-5732 8023
	-4828 -5361	-1908 7687	1203 -5476	8473 -1613	-5367 9788
	-5570 -9776	8434 9669	2240 2452	-9425 2607	-7062 -6815
	-8555 -37	-4049 -4052	-9125 9650	6098 2459	3639 484
	-57 -4438	-1374 -1699	-6791 3666	4603 -8363	-5203 -311
	-5645 5473	-4281 171	-6685 -844	-436 -9061	-3613 -3231
	-5497 9447	3323 -320	-1456 1256	-9166 -2797	617 -6893
	-1396 -5527	6636 6442	-6664 -6973	-928 -7470	9705 7030
	4308 -531	-7650 -2129	-3290 -577	6725 -4387	9444 9366
	-2618 -2771	-7531 -3771	-5286 4592	2199 1015	-9540 6626
	-1013 421	-5147 3468	-2193 2239	4273 -6775	-6609 -3503
	-3036 -4666	-5900 -6519	-1580 9425	-5243 -5995	-5570 -1429
	5263 -5847	-3619 8190	-7973 6024	-7877 85	3084 -5022

No	Data Masukan				Hasil Keluaran
	2728 -698	-1407 -1220	2280 -5867	-4931 1457	-656 -1579
	3306 -8905	-6604 -7310	-7816 -7292	-228 -6773	5580 -3983
	-2313 7363	-4843 -8579	848 -1031	5668 -2604	4775 5815
	-651 -175	-3688 1736	722 -5824	-960 -5363	312 -7703
	-7198 -5037	-2279 5165	8740 2269	-5916 -7705	-4513 1960
	-2271 -7426	-2921 -8242	-9055 -8549	-9879 -8758	-2987 2226
	1218 -2594	-7170 1930	-1001 5554	-6043 -6855	-2925 -5348
	812 -112	7371 -1849	-3323 854	-712 -2319	-990 4314
	-5178 -1043	-7410 -3132	3143 1331	6425 -5783	-754 -9898
	-3017 -5365	-7272 4753	109 -8170	-4018 264	-284 2766
	9354 -7414	-1406 -951	-651 -8195	-5256 -4672	2362 -427
	9852 -9470	-2795 -5627	-2856 2634	-1490 -9444	-8155 -8911
	9607 229	2099 5894	-2864 -6279	1450 -3010	-4606 7732
	-7753 -3802	-2901 -4416	-2337 3234	743 1465	-9609 -9874
	3537 -972	-2986 -3126	-4359 9016	-7093 -3081	-4234 -7986
	-564 -4100	-3 -6878	-5520 -3252	-6702 -1089	6333 1802
	-1058 2173	-3068 1612	2562 5059	5707 -6113	9899 1190
	2154 -3731	2181 -302	1820 6203	-8394 -3956	-1939 -5450
	-412 -9579	5112 -7826	-617 2000	9077 -446	1091 1960
	6222 -6881	-3194 -4307	-6316 -1211	-5454 5630	-369 -8919
	869 8459	3100 -3615	-6848 -9862	339 -8934	-1547 6506
	5317 1965	-2622 -4171	-6000 2067	1373 -4164	907 8917
	3120 2175	-3739 -3574	-7256 5515	783 9755	4584 -3818
	-7212 -2647	-5950 5871	2962 883	-2135 4242	-7676 -6288

No	Data Masukan				Hasil Keluaran	
	8641 -8968	-7012 610	-9672 3194	-6788 2961	-1319 5323	
	-2181 -8951	-5922 -8566	-6422 -7569	-1018 1213	-3427 -4618	
	-5392 -9309	8330 536	604 -605	3153 7981	-1114 -4996	
	-980 3030	-5105 -1368	148 3664	5941 -5566	-9096 4573	
	-1274 3362	-2548 -4934	-3343 -250	3459 -3025	-3840 -207	
	3845 1933	4196 -5830	-8554 -8137	-9776 -7845	8673 -3465	
	5256 -9678	-7376 5630	8027 -5527	-7430 7045	2445 -3603	
	-2368 2514	-1318 -4634	-6887 -2650	-9032 -2105	-5171 2218	
	-9540 -5821	-3937 -5200	257 -2495	-561 541	-3099 -7723	
	5822 -1792	5012 -9849	-4494 6886	-3385 -9868	-7908 -2385	
	560 -8514	-6759 2019	-5166 1277	4695 6658	2091 -8233	
	-9768 3771	-3073 4818	-2212 -9474	-4563 7013	377 2355	
	-1854 341	1805 -9653	-2734 -7873	1592 -2346	1506 1392	
	4805 -2792	1289 253	-7063 -8329	-7690 273	1964 -4202	
	1111 -1440	3277 1672	401 2470	-415 -6148	-2605 4715	
	-7234 -1313	-6064 -347	-9350 6755	-5557 -7005	2476 8857	
	-1235 -7707	219 -4826	-2754 7891	-4874 -8811	401 5319	
	7041 -5972	-3744 -5430	-5391 -3976	9882 4456	-8391 -5099	
	5401 4507	2152 -3479	-4358 75	-4663 4910	2277 -325	
	-3212 9781	-9674 7902	199 -989	8822 -3001	-5814 2055	
	-1911 -2359	-5602 -1881	-9233 -963	8079 7000	-8702 -3649	
	-5332 -3153	-6266 -2307	2811 -9860	2626 -5805	7365 -2955	
	1514 -1807	-8103 9777	-1119 7071	-8272 -1801	-3045 -5182	
	-1233 -8651	1287 -7745	9162 -9772	-6037 8608	1806 -6868	

No	Data Masukan				Hasil Keluaran	
	274-2476	-5312 8245	9989 7452	-2240 -7810	7350 -543	
	-5580 -927	-8254 -2485	3908 -8969	-7016 -3086	-5856 -3033	
	-4108 5997	-983 -9948	-4398 5200	3278 26	-9825 -964	
	2645 -1444	-3407 -4402	9949 5164	-4932 -101	5986 919	
	-9733 -2272	-8992 -3879	-6891 -1517	-9574 -5907	-3583 -1666	
	-8098 -2306	-6806 -7412	1058 -1856	-9198 772	-654 -6370	
	2578 9016	6190 1939	9736 -8011	-8917 124	-4046 261	
	-6492 1871	8760 -4267	-4945 8069	-8184 -7889	5030 -420	
	-8410 492	113 -2493	-3035 8524	-7486 -497	6351 -8474	
	-7682 6379	-5109 -3652	-3233 311	6110 3604	6614 9328	
	1874 -8829	9556 -1044	-1161 -4310	-8694 -7059	-3672 -5946	
	-4434 -4594	-3858 -6429	706 9983	1388 -2096	-3023 -1929	
	-6895 2106	-4967 -2712	4145 1778	-7025 9071	9051 -6097	
	-9824 3725	-6334 -1797	-2380 3387	-8257 -4697	-4841 -3719	
	-520 -925	-6916 6151	-893 1548	-6503 -2088	8422 -3292	
	5079 -2141	-2847 7113	7150 125	-4849 -2081	-3382 -1852	
	-1177 7376	7765 -9284	-7287 -6731	-2832 -2496	-4360 4372	
	8835 7313	3588 -2356	8438 -1838	-8022 -3346	-3811 -6696	
	-6660 -1680	-9350 1242	-4026 -5060	-8256 -7662	-8439 2029	
	-9002 8607	-149 3998	-9508 -5851	-4590 -8775	9168 3716	
	-2677 -4585	-2032 -8291	-4656 -7895	-9851 8918	-9587 -621	
	-6697 -2620	-8039 6297	9031 -9339	-3255 538	-8517 -6641	
	-1401 -3788	-7982 -7985	5713 6371	-1116 854	-4317 1933	
9	-7957 8773	6459 8806	-8210 -9695	123 7089	2529 6033	PASS

No	Data Masukan				Hasil Keluaran	
	-3912 -2738	-8416 1979	-4981 2660	3689 -3324	-689 3195	
	-239 9536	17 2280	-1056 3304	-2944 -5845	3569 8813	
	-5849 -932	1340 -9780	-8992 -7293	-7080 1090	-290 -8185	
	-8445 -5406	-3627 -6551	-2628 4166	-4135 -1264	-3597 1282	
	1921 4137	4003 9663	-3377 425	3964 2182	7538 5364	
	-2523 -3238	-1815 -5439	-2494 -4148	-7539 -7365	-8781 -598	
	1258 -7411	9884 -7081	9965 -8519	-7732 -2431	-4030 6026	
	1816 9422	-3458 -5048	-4888 -3912	-4229 2384	-9995 411	
	7553 5223	3664 4108	-3924 1382	-857 7021	-5397 -4550	
	-2876 7614	3592 -8176	1622 -7275	-1690 -7671	-5388 -8429	
	-9616 -171	8799 302	-2289 -2785	9223 -1070	-4772 711	
	-1104 -7123	152 -6963	2551 -2871	-9033 -9709	-1839 2485	
	-2813 6374	-7533 -9176	-9344 -7788	2488 -696	-1173 -4671	
	1373 -3419	-9654 6366	-465 -2230	3167 7034	-6134 -3435	
	-2116 2706	1946 6344	2652 8176	421 -9638	-3344 -5967	
	-5750 -7120	-5765 -3770	-3990 7920	-2952 128	-4022 -6348	
	-9565 -1818	-785 -8738	-6089 -630	-3246 4780	-9550 -8356	
	-8200 -4373	-8256 1052	-5161 -2255	2809 -4395	-8449 -7867	
	8227 9031	-6822 -2687	8509 3141	-3791 -1187	-8580 -7135	
	-8333 1410	-6442 357	-1037 838	-2977 -9277	-3503 -8	
	306 -8881	9946 -3461	-2829 1483	1041 1423	947 5892	
	-5618 -6507	-3052 -6308	6792 -4300	-8497 -3641	-1552 -2793	
	5842 763	-5857 -5444	4588 -2016	1137 487	-4788 -7408	
	881 8411	2217 -1637	-618 -6485	-355 -4887	4559 -1939	

No	Data Masukan				Hasil Keluaran	
	-4939 -3097	-1253 -2415	911 -9894	1947 -9103	8986 -328	
	-7730 -472	4737 -7690	-398 6312	-5813 8964	-4048 -9754	
	-5340 8057	-198 -2022	-7302 -6424	886 9029	-4701 573	
	2713 -3183	-9818 -2364	-3059 -9729	1624 7880	-1186 -4067	
	-2843 764	2451 -5707	-2352 5213	-747 3470	-3777 2327	
	-2216 111	-6095 -3355	-9064 1277	-1294 8096	-7838 -6192	
	-5679 175	6135 5956	77 -8379	-1811 3978	-5351 -5235	
	-9128 2945	-7975 9087	-7610 -22	-6292 -843	-9616 -4234	
	-8403 -65	-2063 -603	-3399 -7362	-3665 -4468	-5771 -6681	
	608 -7627	3083 5448	6002 -1619	-5517 -6479	-4125 6616	
	-1030 499	-9554 1478	31 -2011	-4814 -5621	-1769 -9721	
	-1939 -7957	-2818 -9048	-2589 147	-1242 6763	296 2609	
	1997 -985	5020 -9732	-5221 9902	6801 -2827	-2592 -1935	
	-1388 -111	-6845 -3199	-660 -1516	-5347 -4646	861 3553	
	-2179 -3302	6193 -3151	-1415 -4328	8599 962	7038 -6572	
	-6805 1778	-4829 -3287	-584 -5940	7722 -9271	-6435 -9950	
	8829 -7189	875 -707	-6340 -8162	-9032 -1679	-1894 -7038	
	1903 1813	2501 -5319	-6298 -6159	2919 1688	-876 8696	
	-5079 -5440	-6814 -3048	-7747 -5388	-3551 -1425	-1891 -5660	
	-2595 -3392	-9776 -9080	-399 3047	9293 -2539	-710 4457	
	-2484 2866	1090 -6060	-2881 1786	-6899 -9539	-8260 -3407	
	4082 -2682	-304 5985	-1057 -1765	-2719 -6897	-4846 3129	

No	Data Masukan				Hasil Keluaran	
	-5348 -9457	-7892 2674	5175 1586	1577 2031	2410 -3165	
	-5501 -5555	1910 -6354	-1560 -1734	2416 2975	-8712 -5435	
	7096 6363	-9597 8997	-4667 4713	-2861 -969	-7580 -8395	
	776 1104	-3358 -2730	-4473 -9293	-408 -4915	-1524 4443	
	3496 -4527	-892 -6755	9237 -9878	-4864 3440	-3274 1594	
	6532 -7285	-578 -9372	2048 8754	4822 -4635	-7295 3117	
	1051 7528	-1144 -6760	-335 -6151	3165 -7200	-4788 566	
	-4708 1164	1461 2416	3161 -7366	-4944 -7705	-4703 -4245	
	-1161 664	-7010 -2142	2411 -6401	-288 2152	-6965 -106	
	-3656 -2796	2950 -9026	-5242 -2390	-6579 8290	-12 -895	
	-5172 -1449	-4315 -6295	-2885 -1722	-2454 -1619	-7952 -9844	
	-5484 3670	-5607 -5632	1198 -431	-4248 -2602	-6420 16	
	5869 1476	-4642 -8202	4073 -4541	-4086 8368	-512 953	
	-8968 319	-4975 -3402	-1937 -5026	3139 1564	-6782 -8736	
	-4384 7970	-7934 -200	-6330 7545	3456 -4801	-7076 -3281	
	5029 9208	-6749 -6800	1970 -775	-1179 -916	8727 8182	
	8432 3924	-5398 917	1110 -2064	-7104 4297	5275 -7501	
	8100 -4171	-9175 752	2210 655	-319 -9808	-2590 388	
	-6575 -1049	2365 -387	2406 -6881	1932 -4349	2176 -8705	
	510 9726	-7732 -8822	2667 8545	-8723 -7999	-4277 2537	
	-8678 1294	-5462 3781	7985 -5790	-5129 -5038	-3103 -8228	
	-1027 520	-2677 -9876	7445 -4038	-146 -7336	-1127 -9221	
	701 -6689	667 -2884	-6249 -6148	-9546 -2947	1584 -210	
	2535 -6973	-4237 -1931	8118 -4922	1230 4064	-7658 -7152	

No	Data Masukan			Hasil Keluaran
	-3198 -7290	-2762 -350	1924 -4368	-6385 5046
	-5538 -2293	-3995 -2321	-8872 -4579	-2311 8817
	8361 -9922	-2437 66	-8625 19	-3185 1284
	-3415 -6732	-4433 -2986	5625 -6376	1413 1277
	8013 -1661	8519 6457	-7362 -3464	-7813 1316
	-8204 -661	-3587 -3238	-2128 6793	-9742 -8628
	171 -1913	-4839 -446	-4079 -8106	5067 2451
	4661 2374	-906 -5371	-4391 3983	1491 -5628
	-9947 5091	1685 -6867	-3112 6090	6128 -8637
	3087 2068	-6820 9535	-6006 -8249	-5618 -8403
	1603 -8850	-5137 -306	-8029 3634	-483 6933
	9304 -657	-1250 9360	-1565 7840	-3363 9429
	-7368 -8171	-4541 9940	2436 -6365	-923 -186
	-2434 -3431	4942 -9117	-3186 -1038	-4226 -2194
	4749 -5629	-2655 7552	-624 -8161	2403 -7829
	-408 2573	-5623 309	-7452 -6226	-4887 4064
	217 7839	-5263 6518	8450 1759	2490 -4771
	2396 -5203	737 -3356	7185 -8327	159 -1383
	4144 1982	-5710 -690	-2396 -3055	2878 2541
	5056 -3716	216 -3447	9491 -6602	7814 -4821
	-7747 9625	-3289 4881	-656 -1882	-9489 -8892
	2465 3663	-3957 6766	1446 -7614	6768 -3117
	-8431 9720	7630 1793	-1966 1675	-2228 1860
	-774 -8956	-4463 3096	-1469 3492	5730 -3885
				-9374 9345
				-4072 -5325
				-6359 -9259
				1527 6968
				2683 -5552
				-2176 -9114
				151 -9253
				-1330 957
				-5528 -7670
				180 -9700
				-9141 -6437
				-9409 -9225
				-4275 6339
				475 -3886
				-3912 3268
				3702 -2357
				654 -1662
				-672 -999
				-2505 -7624
				2236 9622
				6520 -8248
				2483 9368
				-3870 432
				4614 4102

No	Data Masukan					Hasil Keluaran
10	9347 -3784 -4033 -6897 5553 9845 -1631 -8077 -7622 -8856 -9059 4133 -4151 -471 -1557 -6836 8805 -2216 5017 587 7317 -4761 7688 -1361 -6718 -2593 -615 3852 7101 5475 -6517 -477 -1760 -5224 3751 3556 8146 2498 2326 922 -1585 -8367 -4847 2788 -3034 -7076 -796 -6719	-3919 9879 8752 -7113 5668 5125 9471 -5518 8735 2202 -6213 -9042 2439 9183 -3242 4339 -5554 -3517 -3620 974 -8939 -9243 1099 -5691 -6927 -1577 -9239 -2649 530 -4423 9911 -894 -2208 -50 -3202 -966 4039 5005 4437 -9164 7796 -6443 58 4923 1727 -3452 3558 6813	9640 -3316 9972 2233 1509 -6463 -9038 -1081 -2895 -4136 -8941 -7257 -5426 -4539 -8387 -7152 5340 -5079 -4642 -8679 -6461 -9203 -4656 -5015 -2554 -2591 2337 -3574 -2332 -5525 782 6276 4440 -3521 -4079 -4600 -6466 -5814 -8483 -9718 -739 7921 -2616 6421 -6753 5153 7432 -5479	-4043 3967 -6583 -721 2546 1133 -7815 -8083 3217 -7416 -3151 -8569 -7223 464 -4339 5947 -9110 -921 -3788 -9943 -349 346 2043 9809 -5683 -4659 9637 -9055 4909 2299 -4464 -9109 -6602 -6911 -4970 -8748 9367 -3178 5993 -1451 5422 673 4002 1884 -71 -1497 -4066 -9871	9536 5678 1813 -4928 -2490 -6521 9694 -1390 9863 -7522 -3589 -109 -1551 -4393 -3991 -94 2168 1465 3487 -9095 3472 -6587 5712 -4242 1655 -6815 -3846 1501 478 -1076 1149 -3872 -6266 4053 -6263 -8479 1892 5937 2784 -5679 8073 -9080 -6157 -6824 9508 -814 -2003 -6450	PASS










No	Data Masukan				Hasil Keluaran
	6212 302	1802 -564	2145 286	-1832 -4127	6760 -1852
	-9750 -1743	-2335 -9772	639 488	8806 354	3417 -1301
	-1712 5408	-3991 -4515	-5356 -2271	2829 -1800	-8052 -6653
	1723 1303	-2767 -7699	1819 -927	2318 -5929	-4322 -2179
	7321 -3928	-4348 -1817	-9893 -5151	-542 -4662	2545 -5918
	-9370 -6303	-7913 -539	-1575 4331	-2782 5831	-4824 -2151
	-5587 2191	379 -3676	-5483 -5167	-8401 8042	4938 -246
	-3389 1839	-3204 3760	2602 -4954	-3138 -6514	-5715 -3948
	-1269 -9943	-8758 593	-7360 7753	1852 1105	7959 -1356
	-542 2693	1356 5437	-8678 -3874	-1252 7421	-5907 1424
	-3653 9441	-7698 -4610	6365 2628	-2555 -9261	9714 -4706
	-6905 2701	-7239 -1922	-4052 -2016	-1868 852	-6437 -6086
	-1925 4242	-6619 3461	-1682 -5677	-6852 -3893	-3795 -8164
	-6724 -9732	-662 -7838	-8504 -1709	5567 1869	-6917 -9642
	2428 828	-4261 478	100 2953	2002 -5094	3194 -8211
	-3462 1518	9910 -8761	-6024 5151	-5476 -2983	5260 -8749
	-8466 1461	1381 -1805	-6848 -7818	344 -4523	4012 2290
	-2092 505	-312 334	-7761 -5840	-9259 -4431	-3386 -7038
	-2792 4992	-5434 7795	-6897 1565	-6238 -4036	1181 315
	2767 -5569	9448 4689	-8338 961	-989 -376	-8664 -8530
	-3092 -292	1337 -3487	-6309 -4544	-2739 4713	4114 -5406
	6321 -8572	474 7345	-7326 -3791	2896 -2308	-8180 -1628
	411 -995	-7344 -9129	-5278 -4010	-1599 -343	-5413 -7397
	-1795 6980	-473 -3984	-6191 2457	318 -4171	-4358 7154

No	Data Masukan				Hasil Keluaran
	1412 -9697	-1321 -3384	-4731 9897	9467 881	128 -4597
	-4516 -9186	9086 -727	-7917 3683	-2265 -2385	-3097 -6952

[Halaman ini sengaja dikosongkan]

LAMPIRAN B: Hasil Uji Coba Reduksi Polygon pada Situs SPOJ Sebanyak 10 Kali

Berikut merupakan lampiran hasil uji coba kode program dengan algoritma reduksin polygon sebanyak 10 kali.

84777151		2019-11-09 10:31:38	LL and ErBao	accepted with reason 0	0.08	4.6M	CPP14
84777150		2019-11-09 10:31:31	LL and ErBao	accepted with reason 0	0.08	4.5M	CPP14
84777148		2019-11-09 10:31:27	LL and ErBao	accepted with reason 0	0.08	4.6M	CPP14
84777144		2019-11-09 10:31:16	LL and ErBao	accepted with reason 0	0.08	4.6M	CPP14
84777141		2019-11-09 10:31:05	LL and ErBao	accepted with reason 0	0.08	4.7M	CPP14
84777140		2019-11-09 10:30:56	LL and ErBao	accepted with reason 0	0.08	4.4M	CPP14
84777138		2019-11-09 10:30:51	LL and ErBao	accepted with reason 0	0.08	4.7M	CPP14
84777137		2019-11-09 10:30:47	LL and ErBao	accepted with reason 0	0.08	4.6M	CPP14
84777136		2019-11-09 10:30:43	LL and ErBao	accepted with reason 0	0.09	4.6M	CPP14
84777135		2019-11-09 10:30:37	LL and ErBao	accepted with reason 0	0.08	4.6M	CPP14

Gambar C.1 Hasil Pengumpulan Kode Program Utama Dengan Algoritma reduksi polygon

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Penulis bernama Michael Julian Albertus, putra kedua dari tiga bersaudara yang lahir pada tanggal 2 Juli 1998 di Pekanbaru. Penulis telah mengenyam pendidikan di Sekolah Dasar Mardi Yuana Serang pada tahun 2004 hingga 2006, Sekolah Dasar Palm Kids pada tahun 2006 hingga 2009, Sekolah Dasar Negeri 005 Sukajadi Pekanbaru pada tahun 2009 hingga 2010, Sekolah Menengah Pertama Negeri 5 Pekanbaru pada tahun 2010 hingga 2013, dan Sekolah Menengah Atas Negeri 8 Pekanbaru pada tahun 2013 hingga 2015. 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 *artificial intelligence*, *competitive programming*, dan rancang bangun aplikasi sistem informasi. Keinginan penulis dalam mengajar juga mendorong penulis menjadi asisten dosen pada mata kuliah Dasar Pemrograman, Struktur Data, dan Sistem Operasi. Karya penulis semasa perkuliahan diantaranya adalah pembangunan SheNeedsLab dan LPencerdas untuk acara Hackathon. Selama menempuh perkuliahan penulis juga aktif mengikuti kompetisi pemrograman tingkat nasional dan menjadi finalis pada lomba pemrograman COMPFEST (2017, 2018, dan 2019), INC Bina Nusantara (2017, 2018, dan 2019), FINDIT 2019, Arkavidia (2017, 2018, dan 2019) dan menjadi Juara 3 HOLOGY 2018.

Di luar kesibukan akademik, penulis juga berkontribusi dalam berbagai kepanitiaan, baik dalam skala kecil (yaitu dalam kam-

pus) maupun skala nasional. Kepanitian yang penulis ikuti adalah Schematics (2017 dan 2018). Kegiatan terakhir penulis adalah membantu kegiatan pelatihan nasional bagi peserta Olimpiade Komputer Indonesia pada Februari dan Maret 2019 lalu. Penulis dapat dihubungi melalui surel di michaeljulian98@gmail.com.