Peletakan Kamera pada Bangunan *Convex Polygon*

IF-37-05 Kelompok 01

Ananda Faisal Faritz 1301144324

Diska Andini 1301144064

Satrio Adi Prabowo 1301140154

Widodo Setyo Y 1301140194

Bangunan *convex polygon* merupakan bangunan yang memiliki sudut kurang dari atau sama dengan 180o di setiap titik sudutnya. Peletakan kamera pada bangunan yang berbentuk *convex polygon* dengan jumlah kamera seminimal mungkin adalah permasalahan yang akan diselesaikan dalam laporan ini. Asumsi yang digunakan dalam permasalahan ini adalah mengabaikan peletakan kamera, mengabaikan ketinggian kamera dan mengabaikan ukuran setiap sisi dinding ruangan. Jangkauan kamera di definisikan dan kamera tidak boleh diletakkan dititik sudut ruangan. Adapun solusi yang kami tawarkan untuk menyelesaikan permasalahan ini adalah dengan mencoba semua kemungkinan dengan menggunakan segala kombinasi dari jangkauan kamera yang didefinisikan.

# Pendahuluan

Salah satu alat penunjang keamanan yang sering digunakan dalam sebuah ruangan adalah kamera pengawas, dengan peletakan yang benar dapat membuat pengawasan yang maksimal. Biasanya kamera pengawas diletakkan di sudut ruangan, akan tetapi pada permodelan sistem *Surveillance* ini mengharuskan kamera diletakkan di sisi dinding bukan di sudut ruangan dengan jumlah kamera seminimal mungkin. Hal ini membuat kelompok kami tertarik untuk menyelesaikan masalah pada topik ini, yaitu mencari banyak kamera seminimal mungkin untuk pengawasan yang maksimal dengan mendeskripsikan algoritma dan rumusan yang ada.

Untuk menyelesaikan permasalahan ini awalnya akan menggunakan metode *Computational Geometry*, akan tetapi metode ini tidak jadi digunakan karena terdapat metode yang lebih mudah dan lebih menjawab pertanyaan dari permasalahan tersebut. Metode yang akan digunakan adalah mencoba semua kemungkinan dengan menggunakan semua kombinasi dari jangkauan kamera yang didefinisikan.

Program yang dibuat dengan metode ini akan menampilkan jumlah minimal kamera yang didefinisikan dapat menutupi semua dinding, namun jika kamera yang di definisikan tidak dapat menutupi semua sisi dinding maka program akan menampilkan “*impossible*”.

# Deskripsi Masalah dan Asumsi dalam Pemodelan

*The International Corporation for Protection And Control (ICPC)* mengembangkan teknologi yang efisien untuk kontrol dan proteksi yang baik. Tentunya, mereka ingin markas mereka terkontrol dan terproteksi. Dilihat dari atas, bangunan markas mereka memiliki bentuk *convex polygon.* Ada beberapa tempat yang cocok untuk instalasi kamera untuk memonitor bangunan tersebut. Setiap kamera mencakup jarak tertentu dari sisi *polygon* (dinding bangunan), bergantung pada letak kamera tersebut. *ICPC* hendak meminimalkan jumlah kamera yang dibutuhkan untuk mencakup semua bagian dari bangunan.

1. **Masukan**

Masukanterdiri dari uji kasus tunggal. Baris pertamanya terdiri dari 2 *integer* n dan k (3 ≤ n ≤ 106 dan 1 ≤ k ≤ 106). Di mana n adalah jumlah dinding dan k adalah jumlah dinding yang memungkinkan untuk menginstalasi kamera. Dari setiap baris k berisi dua buah *integer* ai dan bi (1 ≤ ai, bi ≤ n). *Integer* tersebut mendefinisikan dinding ke-i yang akan di awasi. Jika ai ≤ bi maka kamera akan mengawasi setiap dinding j di mana ai ≤ j ≤ bi. Jika ai > bi maka kamera akan mengawasi setiap dinding j di mana ai ≤ j ≤ n atau 1 ≤ j ≤ bi.

1. **Keluaran**

Menampilkan jumlah minimal kamera yang cukup untuk mengawasi setiap dinding bangunan. Jarak awas antar dua kamera mungkin saling tumpang tindih. Jika bangunan tidak dapat di awasi, tampilkan *impossible*.

|  |  |
| --- | --- |
| Contoh Masukan 1 | Contoh Keluaran 1 |
| 100 7  1 50  40 70  60 90  0 40  10 60  50 80  0 20 | 3 |

|  |  |
| --- | --- |
| Contoh Masukan 2 | Contoh Keluaran 2 |
| 8 2  8 3  5 7 | Impossible |

|  |  |
| --- | --- |
| Contoh Masukan 3 | Contoh Keluaran 3 |
| 8 2  8 4  5 7 | 2 |

|  |  |
| --- | --- |
| Contoh Masukan 4 | Contoh Keluaran 4 |
| 20 4  1 5  7 13  11 17  15 2 | Impossible  (Karena dinding ke 6 tidak terawasi kamera) |

|  |  |
| --- | --- |
| Contoh Masukan 5 | Contoh Keluaran 5 |
| 50 5  1 15  10 20  20 35  40 10  15 30 | Impossible  (Karena dinding ke 36,37,38,39 tidak terawasi kamera) |

|  |  |
| --- | --- |
| Contoh Masukan 6 | Contoh Keluaran 6 |
| 70 6  1 20  20 35  30 45  45 60  50 65  60 5 | 5  (Semua dinding terawasi kamera) |

|  |  |
| --- | --- |
| Contoh Masukan 7 | Contoh Keluaran 7 |
| 3 2  1 2  2 3 | 2  (Semua dinding terawasi kamera) |

|  |  |
| --- | --- |
| Contoh Masukan 8 | Contoh Keluaran 8 |
| 100 7  1 40  60 80  80 20  30 60  90 30  80 10  70 90 | 4  (Semua dinding terawasi kamera) |

Asumsi pemodelan yang akan dibuat :

1. Ruangan yang akan dipasang kamera pengawas berbentuk *convex polygon.*
2. Ketinggian dan ukuran setiap sisi dinding ruangan diabaikan.
3. Kamera tidak boleh diletakkan di titik sudut ruangan
4. Tidak di definisikan letak kamera namun jangkauan kamera di definisikan.

# Dasar Teori dan Studi Literatur

1. ***Convex Polygon***

Convex Polygon adalah sebuah polygon yang memiliki sudut interior kurang dari 180o. Hal tersebut mengakibatkankan semua titik sudut menjorok keluar menjahui titik sudut.[1]

1. **Representasi Contoh**
2. **Representasi Contoh 2**

Contoh masukan 2 :

n : jumlah dinding

k : jumlah tempat yang memungkinkan dipasang kamera

n k

8 2

Spesifikasi letak kamera

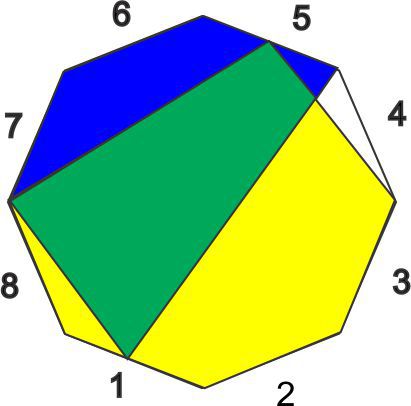
ai bi (Spesifikasi dinding ke-**i** yang akan di awasi.)

k1 8 3

k2 5 7

k1 : karena nilai 8lebih besar dari 3maka kamera akan menutupi setiap dinding j di mana 8≤ j ≤ 8 atau 1 ≤ j ≤ 3.

k2 : karena 7lebih besar dari 5maka kamera akan menutupi setiap dinding j di mana 7 ≤ j ≤ 5.

****

Keterangan :

1. Daerah berwarna biru mewakili daerah yang tertutupi kamera yang ada di dinding 4 .
2. Daerah berwarna kuning mewakili daerah yang tertutupi kamera yang ada di dinding 8 .
3. Daerah berwarna hijau mewakili daerah yang tertutupi dua kamera yang ada di dinding 8 dan 4 .
4. Daerah berwarna putih mewakili daerah yang tidak tertutupi kamera manapun.

Kesimpulan :

Sehingga bangunan dengan dinding (n=8) dan tempat yang memungkinkan untuk instalasi kamera (k=2) dengan spesifikasi k1 (ai = 8 dan bi = 3) dan k2(ai = 5 dan bi = 7) tidak memungkinkan menutupi semua sisi ruangan karena ada sisi yang tidak tertutupi kamera .

1. **Representasi Contoh 3**

Contoh masukan 3 :

n : jumlah dinding

k : jumlah tempat yang memungkinkan dipasang kamera

n k

8 2

Spesifikasi letak kamera

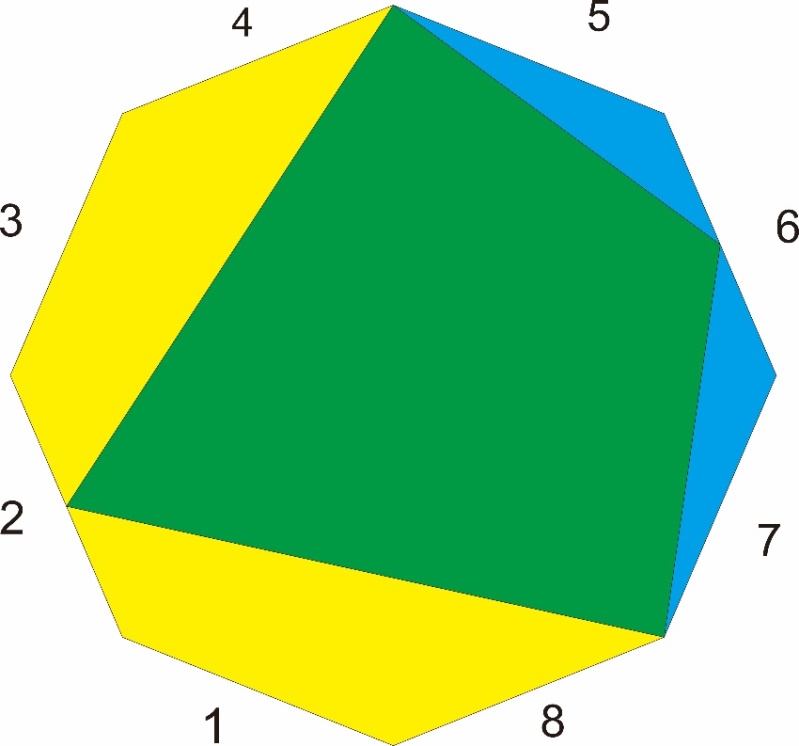
ai bi (Spesifikasi dinding ke-**i** yang akan di awasi.)

k1 8 4

k2 5 7

k1 : karena nilai 8lebih besar dari 4maka kamera akan menutupi setiap dinding j di mana 8 ≤ j ≤ 8 atau 1 ≤ j ≤ 4.

k2 : karena 7lebih besar dari 5maka kamera akan menutupi setiap dinding j di mana 7 ≤ j ≤ 5.



Keterangan :

1. Daerah berwarna biru mewakili daerah yang tertutupi kamera yang ada di dinding 2 .
2. Daerah berwarna kuning mewakili daerah yang tertutupikamera yang ada di dinding 6 .
3. Daerah berwarna hijau mewakili daerah yang tertutupi dua kamera yang ada di dinding 2 dan 6.

Kesimpulan :

Sehingga bangunan dengan dinding (n = 8) dan tempat yang memungkinkan untuk instalasi kamera (k = 2) dengan spesifikasi k1 (ai = 8 dan bi = 4) dan k2 (ai = 5 dan bi = 7) menutupi semua sisi ruangan.

1. **Representasi Contoh dengan Metode yang akan digunakan**

Metode yang akan digunakan berbeda dengan asumsi sebelumnya, metode yang digunakan tidak mempermasalahkan di mana letak kamera berada. Metode ini menganggap bahwa *convex polygon* berbentuk *circle*, sehingga dapat di representasikan dengan *node* yang saling berhubungan dan membentuk *list singular*.

1. **Representasi Contoh 1**

Contoh masukan 1 :

n : jumlah dinding

k : jumlah tempat yang memungkinkan dipasang kamera

n k

100 7

Spesifikasi letak kamera

ai bi (Spesifikasi dinding ke-**i** yang akan di awasi.)

k1 1 50

k2 40 70

k3 60 90

k4  80 40 (kamera mentupi dinding 0-40 atau dinding 80-100)

k5  10 60

k6 50 80

k7  70 20 (kamera mentupi dinding 1-20 atau dinding 70-100)

k1 : karena nilai 50lebih besar dari 1maka kamera akan menutupi setiap dinding j di mana 1 ≤ j ≤ 50.

.k2 : karena 70lebih besar dari 40maka kamera akan menutupi setiap dinding j di mana 40 ≤ j ≤ 70.

k3 : karena 90lebih besar dari 60maka kamera akan menutupi setiap dinding j di mana 60 ≤ j ≤ 90

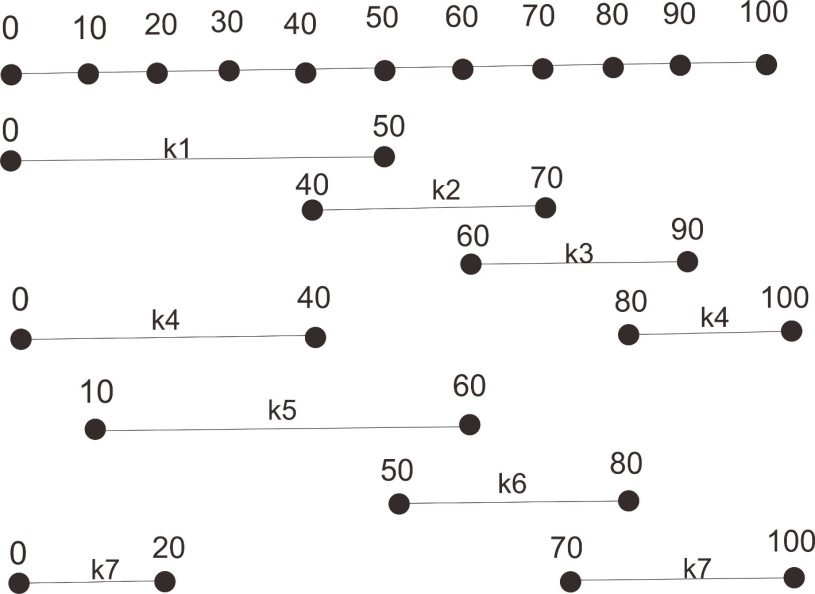
k4 : karena 40lebih besar dari 1maka kamera akan menutupi setiap dinding j di mana 1 ≤ j ≤ 40 atau 80 ≤ j ≤ 100

k5 : karena 60lebih besar dari 10maka kamera akan menutupi setiap dinding j di mana 10 ≤ j ≤ 60

k6 : karena 80lebih besar dari 50maka kamera akan menutupi setiap dinding j di mana 50 ≤ j ≤ 80

k7 : karena 20lebih besar dari 0maka kamera akan menutupi setiap dinding j di mana 0 ≤ j ≤ 20 atau 70 ≤ j ≤ 100

Dari keterangan di atas kita misalkan jumlah sisi dinding yang akan diawasi kamera dalam bentuk *graph singular list* sebagai berikut :



Pada gambar di atas *node* dimisalkan sebagai sudut dinding, di antara *node* 0 dan 10 merupakan sisi dinding 1 , di antara *node* 10 dan 20 merupakan sisi dinding 2 dan seterusnya. Untuk kamera pertama dimisalkan sebagai k1, terlihat pada gambar kamera 1 akan menutupi sisi dinding 1-50. Kamera 2 dimisalkan sebagai k2 yang menutupi sisi dinding 40-70.untuk kamera selanjutnya sama sperti contoh kamera 1 dan 2. Karena tidak ada sisi dinding yang tidak tertutupi maka program akan menampilkan jumlah kamera yang digunakan tersebut, akan tetapi pada representasi contoh kali ini kita akan menentukan jumlah minimal kamera yang dapat digunakan. Dapat dilihat dari representasi contoh kita hanya membutuhkan 3 kamera, yaitu kamera 1, 4 dan 6.

1. **Representasi Contoh 2**

Contoh masukan 2 :

n : jumlah dinding

k : jumlah tempat yang memungkinkan dipasang kamera

n k

8 2

Spesifikasi letak kamera

ai bi (Spesifikasi dinding ke-**i** yang akan di awasi.)

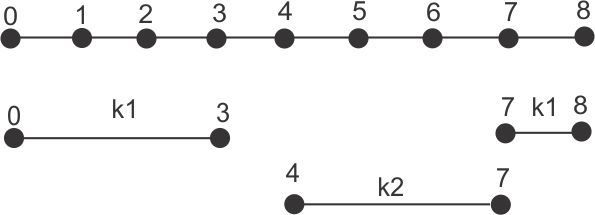
k1 8 3

k2 5 7

k1 : karena nilai 8lebih besar dari 3maka kamera akan menutupi setiap dinding j di mana 8≤ j ≤ 8 atau 1 ≤ j ≤ 3.

.k2 : karena 7lebih besar dari 5maka kamera akan menutupi setiap dinding j di mana 7 ≤ j ≤ 5.

Dari keterangan di atas kita misalkan jumlah sisi dinding yang akan diawasikamera dalam bentuk *graph singular list* sebagai berikut :



Pada gambar di atas *node* dimisalkan sebagai sudut dinding, di antara *node* 0 dan 1 merupakan sisi dinding 1 , di antara *node* 1 dan 2 merupakan sisi dinding 2 dan seterusnya. Untuk kamera pertama dimisalkan sebagai k1, terlihat pada gambar kamera 1 akan menutupisisi dinding 1 ,2,3 dan 8. Kamera 2 dimisalkan sebagai k2 yang menutupi sisi dinding 5, 6 dan 7. Karena terdapat sisi dinding 4 yang tidak tertutupi, maka program akan menampilkan “*impossible*”.

1. **Representasi Contoh 3**

Contoh masukan 3 :

n : jumlah dinding

k : jumlah tempat yang memungkinkan dipasang kamera

n k

8 2

Spesifikasi letak kamera

ai bi (Spesifikasi dinding ke-**i** yang akan di awasi.)

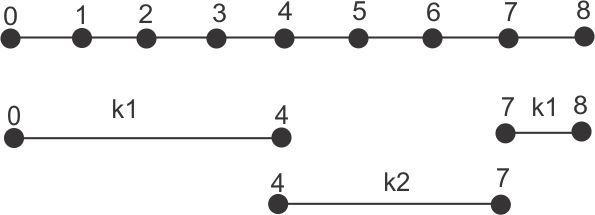
k1 8 4

k2 5 7

k1 : karena nilai 8lebih besar dari 4maka kamera akan menutupi setiap dinding j di mana 8 ≤ j ≤ 8 atau 1 ≤ j ≤ 4.

.k2 : karena 7lebih besar dari 5maka kamera akan menutupi setiap dinding j di mana 7 ≤ j ≤ 5.

Dari keterangan di atas kita misalkan jumlah sisi dinding yang akan diawasi kamera dalam bentuk *graph singular list* sebagai berikut :



Pada gambar di atas *node* dimisalkan sebagai sudut dinding, di antara *node* 0 dan 1 merupakan sisi dinding 1 , di antara *node* 1 dan 2 merupakan sisi dinding 2 dan seterusnya. Untuk kamera pertama dimisalkan sebagai k1, terlihat pada gambar kamera 1 akan menutupi sisi dinding 1 ,2,3,4 dan 8. Kamera 2 dimisalkan sebagai k2 yang menutupi sisi dinding 5, 6 dan 7. Karena tidak ada sisi dinding yang tidak tertutupi maka program akan menampilkan jumlah kamera yang digunakan tersebut yaitu 2.

1. **Representasi Contoh 4**

Contoh masukan 4 :

n : jumlah dinding

k : jumlah tempat yang memungkinkan dipasang kamera

n k

20 4

Spesifikasi letak kamera

ai bi (Spesifikasi dinding ke-**i** yang akan di awasi.)

k1 20 4

k2 1 5

k3 7 13

k4  11 17 (kamera mentupi dinding 0-40 atau dinding 80-100)

k5  15 2

k1 : karena nilai 5lebih besar dari 1maka kamera akan menutupi setiap dinding j di mana 1 ≤ j ≤ 5.

k2 : karena 13lebih besar dari 7maka kamera akan menutupi setiap dinding j di mana 7 ≤ j ≤ 17.

k3 : karena 17 lebih besar dari 11 maka kamera akan menutupi setiap dinding j dimana 11≤ j ≤ 17.

k4  :karena 2 lebih kecil 15 maka kamera akan menutupi dinding 15 sampai dengan 20 dan dinding 1 sampai dengan 2

1. **Representasi Contoh 5**

Contoh masukan 5 :

n : jumlah dinding

k : jumlah tempat yang memungkinkan dipasang kamera

n k

50 5

Spesifikasi letak kamera

ai bi (Spesifikasi dinding ke-**i** yang akan di awasi.)

k1 1 15

k2 10 20

k3 20 35

k4  40 10

k5  15 30

k1 : karena nilai 15lebih besar dari 1maka kamera akan menutupi setiap dinding j di mana 1 ≤ j ≤ 5.

k2 : karena 20lebih besar dari 10maka kamera akan menutupi setiap dinding j di mana 10 ≤ j ≤ 20.

k3 : karena 35 lebih besar dari 20 maka kamera akan menutupi setiap dinding j dimana 20≤ j ≤ 35.

k4 :karena 10 lebih kecil 40 maka kamera akan menutupi dinding 40 sampai dengan 50 dan dinding 1 sampai dengan 10

k5 :karena 30 lebih besar dari 15 maka kamera akan menutupi setiap dinding j dimana 15 ≤ j ≤ 30.

1. **Representasi Contoh 6**

Contoh masukan 6 :

n : jumlah dinding

k : jumlah tempat yang memungkinkan dipasang kamera

n k

70 6

Spesifikasi letak kamera

ai bi (Spesifikasi dinding ke-**i** yang akan di awasi.)

k1 1 20

k2 20 35

k3  30 45

k4  45 60

k5  50 65

k6  60 5

k1 : karena nilai 20lebih besar dari 1maka kamera akan menutupi setiap dinding j di mana 1 ≤ j ≤ 20.

k2 : karena 35lebih besar dari 20maka kamera akan menutupi setiap dinding j di mana 20 ≤ j ≤ 35.

k3 : karena 45 lebih besar dari 30 maka kamera akan menutupi setiap dinding j dimana 30≤ j ≤45.

k4 :karena 60 lebih besar dari 45 maka kamera akan menutupi setiap dinding j dimana 45 ≤ j ≤ 60.

k5 :karena 65 lebih besar dari 40 maka kamera akan menutupi setiap dinding j dimana 40 ≤ j ≤ 65

k6  : karena 5 lebih keil dari 60 maka kamera akan menutupi dinding 60 dan dinding 1 sampai dengan 5

1. **Representasi Contoh 7**

Contoh masukan 7 :

n : jumlah dinding

k : jumlah tempat yang memungkinkan dipasang kamera

n k

3 2

Spesifikasi letak kamera

ai bi (Spesifikasi dinding ke-**i** yang akan di awasi.)

k1 1 2

k2 2 3

k1 : karena nilai 2lebih besar dari 1maka kamera akan menutupi setiap dinding j di mana 1 ≤ j ≤ 2.

k2 : karena 3lebih besar dari 2maka kamera akan menutupi setiap dinding j di mana 2 ≤ j ≤ 3.

1. **Representasi Contoh 8**

Contoh masukan 8 :

n : jumlah dinding

k : jumlah tempat yang memungkinkan dipasang kamera

n k

100 7

Spesifikasi letak kamera

ai bi (Spesifikasi dinding ke-**i** yang akan di awasi.)

k1 1 40

k2 60 80

k3  80 20

k4  30 60

k5  90 30

k6  80 10

k7 70 90

k1 : karena nilai 40lebih besar dari 1maka kamera akan menutupi setiap dinding j di mana 1 ≤ j ≤ 40.

k2 : karena 80lebih besar dari 60maka kamera akan menutupi setiap dinding j di mana 60 ≤ j ≤ 80.

k3 : karena 20 lebih kecil dari 80 maka kamera akan menutupi dinding 80 sampai dengan 100 dan 1 sampai dengan 20.

k4 :karena 60 lebih besar dari 30 maka kamera akan menutupi setiap dinding j dimana 30 ≤ j ≤ 60.

k5 :karena 30 lebih kecil dari 90 maka kamera akan menutupi dinding 90 sampai dengan 100 dan 1 sampai dengan 30

k6  : karena 10 lebih keil dari 80 maka kamera akan menutupi dinding 80 sampai dengan 100 dan dinding 1 sampai dengan 10

k7  : karena 90 lebih besar dari 70 maka kamera akan menutupi dinding 70 sampai dengan 90

1. **Representasi Contoh 9**

Contoh masukan 9 :

n : jumlah dinding

k : jumlah tempat yang memungkinkan dipasang kamera

n k

150 8

Spesifikasi letak kamera

ai bi (Spesifikasi dinding ke-**i** yang akan di awasi.)

k1 1 30

k2 50 80

k3  90 120

k4  100 140

k5  30 80

k6  130 20

k7  110 10

k8 80 100

k1 : karena nilai 30lebih besar dari 1maka kamera akan menutupi setiap dinding j di mana 1 ≤ j ≤ 30.

k2 : karena 80lebih besar dari 50maka kamera akan menutupi setiap dinding j di mana 50 ≤ j ≤ 80.

k3 : karena 120 lebih besar dari 90 maka kamera akan menutupi setiap dinding j dimana 90 ≤ j ≤120.

k4 :karena 140 lebih besar dari 100 maka kamera akan menutupi setiap dinding j dimana 100 ≤ j ≤ 140.

k5 :karena 80 lebih besar dari 30 maka kamera akan menutupi

setiap dinding j dimana 30 ≤ j ≤ 80.

k6  : karena 20 lebih kecil dari 130 maka kamera akan menutupi dinding 130 sampai dengan 150 dan dinding 1 sampai dengan 20

k7  : karena 10 lebih kecil dari 110 maka kamera akan menutupi dinding 110 sampai dengan 150 dan 1 sampai dengan 10

k8 : karena 100 lebih besar dari 80 maka kamera akan menutupi setiap dinding j dimana 80 ≤ j ≤100.

1. **Representasi Contoh 10**

Contoh masukan 10 :

n : jumlah dinding

k : jumlah tempat yang memungkinkan dipasang kamera

n k

13 3

Spesifikasi letak kamera

ai bi (Spesifikasi dinding ke-**i** yang akan di awasi.)

k1 8 3

k2 6 10

k3  1 4

k1 : karena nilai 3lebih kecil dari 8maka kamera akan menutupi dinding 8 sampai dengan 13 dan 1 sampai dengan 3

k2 : karena 10lebih besar dari 6maka kamera akan menutupi setiap dinding j di mana 6 ≤ j ≤ 10.

k3 : karena 4 lebih besar dari 1 maka kamera akan menutupi setiap dinding j dimana 1 ≤ j ≤4.

# Pemodelan Sistem dan Analisisnya

**Tahapan-tahapan Pemodelan Sistem** :

1. Mendefinisikan jumlah dinding yang terdapat pada sebuah bangunan berbentuk *convex* *polygon*
2. Menentukan jumlah tempat yang memungkinkan untuk menginstalasi kamera
3. Menentukan jangkauan setiap kamera.
4. Menentukan jumlah minimal kamera yang dapat men-*cover* semua sisi dinding pada bangunan tersebut

**Spesifikasi Metode atau Algoritma yang digunakan**

Untuk merealisasikan metode yang akan digunakan diperlukan tahapan-tahapan sebagai berikut :

1. Masukkan dua buah *integer* (n dan k)

n : jumlah dinding

k : jumlah dinding yang memungkinkan untuk menginstalasi kamera

1. Masukkan ai dan bi sebanyak k kali (k adalah *array*, ai dan bi adalah elemen dari *array* k, (i) menyatakan indeks).

Jika ai ≤ bi maka kamera akan mengawasi setiap dinding j di mana ai ≤ j ≤ bi. Jika ai > bi maka kamera akan mengawasi setiap dinding j di mana ai ≤ j ≤ n atau 1 ≤ j ≤ bi.

1. Melakukan gabungan untuk setiap elemen dalam *array* k. Gabungan dilakukan dengan mengombinasikan elemen dari *array* k. Sehingga akan dihasilkan banyaknya kombinasi sebanyak:

n\_Combination =

Kombinasi yang dihasilkan merupakan sebuah *array* sebagai berikut:

1. Memeriksa apakah hasil dari kombinasi tersebut dapat mendefinisikan bahwa semua dinding tertutupi atau tidak. Pemeriksaan dilakukan menggunakan pencarian sekuensial terhadap *array* hasil kombinasi. Apabila elemen dari *array* tersebut mencakup seluruh dinding, maka pencarian akan berhenti.
2. Menampilkan jumlah minimal kamera yang dapat menutupi semua sisi dinding bangunan atau menampilkan *impossible* jika ada sisi dinding bangunan yang tidak tertutupi.

Berikut adalah *pseudocode* yang digunakan untuk metode yang digunakan :

Program Surveillance

1 {I.S. : -}

2 {F.S. : Program mengeluarkan jumlah kamera minimal dari jumlah dinding, jumlah dinding yang mungkin di letakan kamera dan jangkauan kamera yang telah dimasukan}

4 function combinations(input iterable: list of integer, input integer: r): list of integer

5 {I.S. : -}

6` {F.S. : Fungsi mengembalikan List hasil kombinasi}

7 {contoh kombinasi combinations ('ABC', 2) --> AB AC BC}

8 kamus

9 pool, indices: list of integer

10 n: integer;

11 algoritma

12 {memindahkan isi parameter iterable ke variabel pool}

13 pool = tuple(iterable)

14 {menginisialisasi n dengan panjang dari pool}

15 n = pool.length

16 if r > n then {bila jumlah r > n return nul}

17 return null

18 end if;

19 {menginisialisasi indices dengan list of integer dari 1 hingga r}

20 indices = range(r)

21 {menyimpan sementara elemen pada pool[i] sesuai dengan indeks pada indices}

22 yield tuple(pool[i] for i in indices)

23 while True: {selama indices[i] != i + n - r}

24 for i in reversed(range(r)):

25 {iterasi dari r downto i}

26 if indices[i] != i + n - r:

27 break

28 }

29 else:

30 return

31 indices[i] += 1

32 for j in range(i+1, r):

33 indices[j] = indices[j-1] + 1

34 yield tuple(pool[i] for i in indices)

35 {melanjutkan isi elemen dari pool[i]}

36 end function;

37 kamus

38 i, n, k: integer

39 cam\_posbl, comb\_list, answer: list of integer

40 algoritma

41 begin

42 {inisialisasi n dan k}

43 input(n,k)

44 {inisialisasi jawaban}

45 for i:=1 to n do

46 begin

47 answer[i] := i

48 end for;

49 {mengisi array k beserta elemennya}

50 for i:=1 to k do {iterasi dari i hingga k}

51 begin

52 {inisialisasi ai dan bi}

53 input(a, b)

54 if a <= b then {jika nilai ai <= bi}

55 for j:=a to b do {iterasi dari a hingga b}

56 begin

57 isi elemen terakhir dari camp\_posbl[i] dengan j}

58 cam\_posbl[i].append(j)

59 end for;

60 else

61 {jika nilai ai > bi}

62 for j:=a to n do {iterasi dari a hingga n}

63 begin

64 if j == n then

65 {jika nilai j = n}

66 for j:=1 to b do iterasi dari 1 hingga b}

67 begin

68 {isi elemen terakhir dari camp\_posbl[i] dengan j}

69 cam\_posbl[i].append(j)

70 end for;

71 else

72 {jika nilai j != n}

73 {isi elemen terakhir dari camp\_posbl[i] dengan j}

74 cam\_posbl[i].append(j)

75 end if;

76 end for;

77 end if;

78 end for;

79 {membuat kombinasi dari comb\_posbl dan disimpan ke dalam list}

80 for i:=1 to k do

81 comb\_list[i]+=combinations(cam\_posbl, i)

82 {mencari dari elemen pertama hingga elemen terakhir}

83 for i:=1 to comb.length do

84 {iterasi dari elemen pertama hingga banyaknya kombinasi}

85 begin

86 if comb\_list[i] == answer then

87 {jika comb\_list[i] = answer}

88 output(comb\_list[i].length) {outputkan banyaknya elemen dalam comb\_list[i]}

89 end if;

90 if comb\_list[i] != answer and i == comb.length then

91 {jika comb\_list[i] != anser dan i = banyaknya kombinasi}

92 output('impossible') {outputkan string impossible}

93 end if;

94 end for;

95 end.

**Kompleksitas Algoritma**

Algoritma untuk menghasilkan semua kombinasi dilakukan dengan teknik *Brute Force.* Algoritma akan melakukan kombinasi i dari k. Dimana i adalah bilangan bulat (1,2,3..k). kompleksitas dari algoritma dapat di representasikan sebagai berikut :

Atau

Dapat disimpulkan bahwa kompleksitas dari algoritma tersebut merupakan eksponensial

# Kesimpulan

# Peletakan kamera pengawas dengan jumlah minimal pada bangunan berbentuk *convex polygon* dengan asumsi yang digunakan dalam permasalahan ini adalah mengabaikan peletakan kamera, mengabaikan ketinggian kamera dan mengabaikan ukuran setiap sisi dinding ruangan. Jangkauan kamera di definisikan dan kamera tidak boleh diletakkan di titik sudut ruangan. Metode yang diajukan pada permasalahan ini yaitu mencoba semua kemungkinan dengan menggunakan segala kombinasi dari jangkauan kamera yang didefinisikan.

# Kelemahan dalam Pemodelan Sistem

# Teknik yang diterapkan dalam pemodelan sistem ini adalah *Brute Force,* artinya program harus mencoba semua kemungkinan yang ada untuk mendapatkan solusi. Karena hal tersebut, solusi yang kami tawarkan untuk pemodelan sistem ini belum efisien.

# Program dapat menentukan jangkauan setiap kamera yang digunakan untuk menutupi seluruh sisi dinding, tetapi tidak dapat menentukan kamera yang digunakan untuk menutupi seluruh sisi dinding.

# Referensi

1. *polygonconvex*. (2011). Retrieved 05 2016, from mathopenref: http://www.mathopenref.com/polygonconvex.html

**Lampiran**