

Penentuan Penempatan Kamera Pengawas Dalam Suatu Ruang dengan Menggunakan Algoritma *Divide and Conquer*

Ricardo Pramana Suranta / 13509014

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

13509014@std.stei.itb.ac.id

Kamera pengawas (*security camera*) adalah alat yang digunakan untuk menjaga keamanan berbagai tempat penting. Penempatan kamera pengawas didalam suatu daerah adalah hal yang penting, oleh karena mengikuti daerah yang hendak diawasi, spesifikasi kamera, serta berbagai hal lain. Penggunaan algoritma *divide and conquer*, dibantu dengan struktur data *quadtree* dapat membantu untuk menentukan penempatan kamera pengawas dalam suatu daerah tertentu, terutama ruang yang memiliki batas yang tetap dan *field of view* kamera yang tetap pula.

Kata Kunci — kamera pengawas, *field of view*, *divide and conquer*, *quadtree*

I. PENDAHULUAN

Kamera pengawas (*security camera*) adalah alat yang lazim kita temukan pada tempat-tempat yang memerlukan pengawasan khusus, seperti bank, bandara, atau mall. Kamera pengawas, yang umumnya dikenal sebagai CCTV (Closed-circuit television) berfungsi untuk untuk mengawasi daerah yang spesifik, dan mengirimkan sinyal tersebut ke daerah yang spesifik pula, misalnya ruang monitor keamanan di gedung-gedung. Oleh karena dapat mengawasi berbagai tempat secara *real-time* pada waktu yang bersamaan tanpa henti, kamera keamanan banyak digunakan di untuk mengawasi tempat-tempat penting, sehingga dapat menekan biaya untuk mempekerjakan penjaga khusus.

Oleh karena kamera pengawas berfungsi untuk mengawasi daerah tertentu, dan memiliki “jarak pandang” yang terbatas, maka penempatan kamera agar seluruh daerah, terutama daerah-daerah yang penting dapat terawasi dengan baik, dengan jumlah kamera pengawas sesedikit mungkin. Pada gedung tempat penulis biasa berkuliah, yakni Labtek V, sempat terjadi kebakaran didalam sebuah toilet, yang sempat dihentikan oleh mahasiswa sebelum api menjalar ke daerah lain. Pihak yang berwajib mengatakan, sang pelaku telah mencari jalur yang tidak terlihat oleh kamera pengawas terlebih dahulu sebelum bertindak, oleh karena itu, kejadian kebakaran tersebut tidak diketahui oleh petugas karena

daerah toilet tidak terekam oleh kamera pengawas, hingga akhirnya mahasiswa yang masih beraktifitas di sekitar Labtek V menemukan kejadian kebakaran tersebut dan memanggil pihak yang berwajib.

Dalam makalah ini, penulis hendak mencoba menentukan penempatan *security camera* dengan cara membagi daerah atau ruangan yang hendak diawasi dengan struktur data *quadtree*, yang umumnya digunakan sebagai struktur data untuk algoritma *divide and conquer* yang umum digunakan pada permasalahan-permasalahan besar yang dapat dipecah menjadi sub-permasalahan yang cukup kecil untuk dikerjakan, lalu digabungkan kembali menjadi sebuah solusi umum.

II. DASAR TEORI

2.1 Kamera Pengawas

Kamera pengawas yang dijual di pasaran pada saat ini memiliki berbagai macam jenis, spesifikasi, dan harga. Namun, tiap-tiap kamera memiliki satu hal utama, yakni *field of view* (ruang pandang), yaitu seberapa besar daerah yang dapat terlihat dengan jelas oleh sebuah kamera pengawas. *Field of view* sebuah kamera pengawas dipengaruhi oleh beberapa hal berikut :

- *Focal Length* (jarak fokus) : jarak dari titik tengah lensa kepada sebuah kepada *focal point* (titik fokus). Semakin jauh *focal length*, maka semakin sempit sudut pandang daerah yang dapat terlihat (*field of view*) oleh kamera pengawas.
- *Camera Installation Height*: tinggi pemasangan kamera.
- *Camera Tilt*: kemiringan kamera terhadap tanah.
- *Camera Sensor Format*: *format* dari sensor kamera yang digunakan, yakni berapa banyak sensor penangkap cahaya yang dimiliki oleh kamera. Beberapa *format* yang beredar di pasaran adalah 1/4", 1/3.6", 1/3" , 1/2.5", 1/2", 2/3", 1" dan 1.25". Semakin banyak sensor penangkap cahaya yang dimiliki suatu kamera, semakin sedikit *noise* yang akan dihasilkan pada

gambar yang terekam. Semakin banyak *noise* yang dihasilkan, maka jarak pandang dari kamera juga semakin terbatas.

Selain hal diatas, kualitas gambar yang dihasilkan juga dipengaruhi oleh resolusi gambar yang dihasilkan oleh kamera, yang menentukan seberapa besar daerah dari gambar yang dapat terlihat dengan jelas oleh pengguna.



Gambar 1. Salah satu contoh kamera pengawas yang dipasang di ujung gedung.

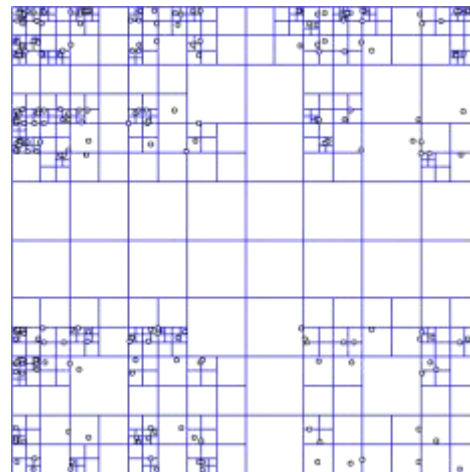
Oleh karena kamera pengawas berfungsi untuk memungkinkan pengawasan yang berlangsung secara terus-menerus, maka adalah hal yang wajar bila kamera pengawas ditemukan pada tempat-tempat penting seperti bandara dan bank, atau bahkan pada tempat-tempat kecil seperti *minimarket* atau *convenience store*. Tidak jarang, pada tempat-tempat seperti *minimarket*, dipasang kamera pengawas *dummy*, yang tidak dapat berfungsi sebagaimana kamera pengawas sesungguhnya, namun terlihat persis seperti yang asli. Biasanya, pemasangan kamera *dummy* ini diikuti dengan “peringatan” seperti “Smile! You are on CCTV”, hanya untuk mengurangi niat pencuri untuk melakukan aksinya. Dan memang, penggunaan kamera *dummy* terbukti efektif untuk mengurangi pencurian di tempat-tempat seperti *minimarket* dan *convenience store*.

2.2 Quadtree

Quadtree adalah struktur data berbentuk pohon yang tiap simpulnya memiliki tepat empat buah anak, kecuali daun (simpul terakhir dari pohon). *Quadtree* pada umumnya digunakan untuk membagi ruang dua dimensi secara rekursif menjadi empat kuadran atau daerah, dan akan kembali memecah daerah tersebut hingga mencapai suatu batas tertentu, yang merupakan luas daerah terkecil yang telah didefinisikan sebelumnya. Daerah tersebut dapat berbentuk bujursangkar, persegi panjang, atau bentuk-bentuk lain. Struktur data ini dinamakan sebagai *quadtree* oleh Raphael Finkel dan J.L. Bentley pada tahun 1974. Pada implementasinya, *quadtree* dapat memiliki berbagai macam bentuk, namun pada umumnya memiliki fitur-fitur berikut :

- Terbagi menjadi sel-sel yang dapat diadaptasi
- Tiap sel memiliki kapasitas maksimum, yang akan terbagi kembali apabila mencapai kapasitas tersebut.

- Hirarki direktori pohon akan mengikuti dekomposisi spasial dari *quadtree*, yakni bagaimana *quadtree* tersebut terlihat secara spasial atau kasat mata.



Gambar 2. Sebuah region *quadtree* dengan data berbentuk titik..

2.3 Algoritma Divide and Conquer

Awalnya, *divide and conquer* adalah strategi militer yang kerap digunakan dalam peperangan pada zaman dahulu. Jenderal-jenderal perang pada masa lampau mengamati bahwa lebih mudah mengalahkan musuh apabila telah dibagi-bagi menjadi bagian kecil dan menyerang masing-masing bagian tersebut satu per satu, daripada menyerang pasukan musuh secara langsung. Strategi militer ini dikenal dengan nama *divide ut imperes* (yang dikenal sebagai politik *divide et impera* pada zaman penjajahan Belanda), yang berarti “pecah belah dahulu kemudian kuasai.”

Algoritma *divide and conquer* membagi masalah (*problem*) menjadi beberapa upa-masalah (*subproblem*) yang lebih kecil, kemudian menyelesaikan masing-masing upa-masalah (*subproblem*) yang lebih kecil, kemudian menyelesaikan masing-masing upa-masalah secara independen, dan akhirnya menggabungkan solusi masing-masing upa-masalah tersebut menjadi solusi masalah semula. Secara rinci, proses algoritma *divide and conquer* disusun atas tiga proses utama :

1. *Divide*: membagi masalah menjadi beberapa upa-masalah yang memiliki kemiripan dengan masalah semula, namun berukuran lebih kecil.
2. *Conquer*: menyelesaikan masing-masing upa-masalah (secara rekursif)
3. *Combine*: menggabungkan solusi masing-masing upa-masalah sehingga membentuk solusi masalah semula.

Metode *divide and conquer* memberikan dua keuntungan. Pertama, metode ini menyediakan pendekatan yang sederhana untuk memecahkan masalah yang secara konseptual sulit, dengan cara mereduksi ukuran masalah secara rekursif sampai ia menjadi kasus basis yang mudah diselesaikan secara trivial. Kedua, bahkan jika solusi masalah sudah diketahui secara pasti

(misalnya masalah pengurutan, perpangkatan bilangan secara *brute force*, dll), *divide and conquer* dapat secara substansial mengurangi biaya (*cost*) komputasi (kompleksitas). Sebagai contoh, bila masalah semula membutuhkan waktu sebesar $O(n^2)$ untuk masalah yang berukuran n , dan langkah penggabungan solusi dari dua buah upa-masalah membutuhkan waktu $O(n)$, maka algoritma *divide and conquer* dapat mengurangi kompleksitas waktu menjadi $O(n \log n)$.

III. PENENTUAN PENEMPATAN KAMERA PENGAWAS MENGGUNAKAN ALGORITMA *DIVIDE AND CONQUER*

Umumnya, penempatan kamera pengawas dirancang oleh manusia, dengan perkiraan-perkiraan tertentu, seperti biaya, lokasi, keamanan, dan berbagai hal lainnya. Berikut, penulis mencoba melakukan penentuan penempatan kamera pengawas dengan menggunakan algoritma *divide and conquer*. Namun, dalam melakukan hal ini, penulis memberikan beberapa batasan dan prasyarat, antara lain :

- Penentuan penempatan dilakukan dengan cara mengambil luas ruang lingkup, dan membaginya kedalam struktur data *quadtree*, yakni menjadi empat bagian, dan terus melakukannya secara rekursif hingga setiap bagian telah memenuhi luas tertentu yang telah ditentukan sebelumnya.
- *Field of view*, atau luas daerah yang dapat terekam dengan jelas oleh kamera pengawas telah ditetapkan terlebih dahulu. Hal ini berarti jenis, ketinggian pemasangan, serta kemiringan kamera terhadap tanah dalam pemasangannya haruslah sama, oleh karena *field of view* kamera akan menjadi luas “pembatas” dari pembagian ruang lingkup yang menjadi tempat penempatan kamera pengawas.
- Ruang lingkup dimana kamera pengawas akan diletakkan adalah ruang terbatas, yang luasnya telah ditentukan terlebih dahulu.
- Posisi benda-benda yang hendak diawasi, dan/atau jalur yang memungkinkan orang-orang untuk masuk dan keluar dari ruang lingkup tersebut telah ditentukan terlebih dahulu.
- Ruang dan benda direpresentasikan dalam bentuk segiempat.

Input dari proses penempatan kamera pengawas ini adalah :

1. Panjang dan lebar dari ruang lingkup tempat penempatan kamera, dalam satuan yang telah ditentukan (misalkan meter.)
2. Posisi dan luas masing-masing benda yang hendak diawasi, antara lain posisi titik tengah benda dalam koordinat x dan y dari tiap benda, dengan titik tengah ruang lingkup sebagai sumbu utama, serta panjang dan lebar benda.
3. Posisi dan lebar masing-masing jalur yang tersedia, antara lain posisi titik tengah benda dalam

koordinat x dan y dari tiap jalur, dan lebar jalur tersebut.

4. Luas dari *field of view* sebuah kamera pengawas, antara lain panjang dan lebar *field of view*.

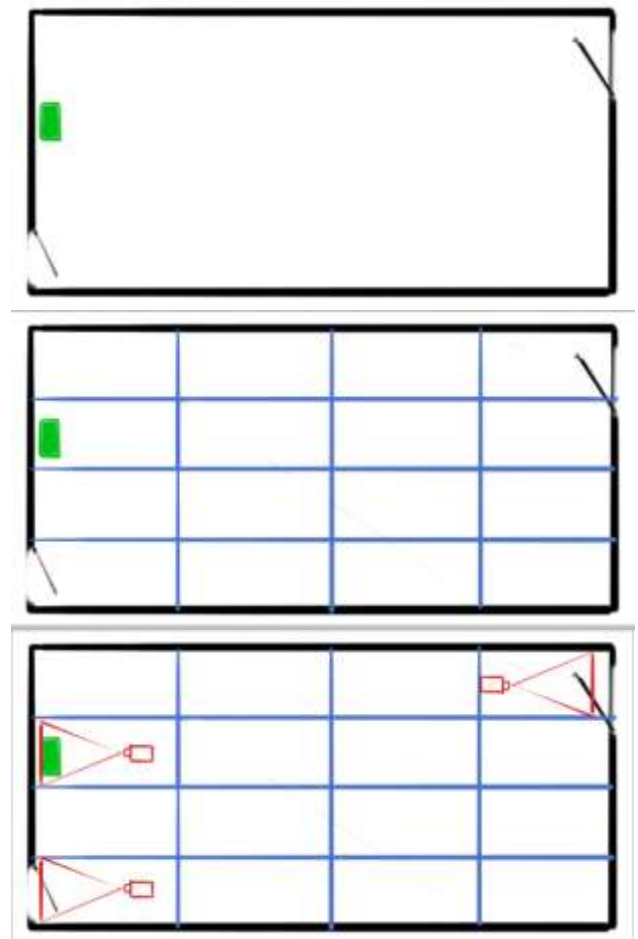
Output dari proses penempatan ini adalah representasi dari ruang lingkup masukan dalam bentuk *quadtree*. Masing-masing daun dari *quadtree* menyimpan hal berikut :

- Titik tengah ruang bagian tersebut (dalam koordinat x dan y), beserta panjang dan lebarnya.
- Posisi dan luas (untuk benda) atau lebar (untuk jalur) serta nomor identifikasi untuk tiap benda dan jalur yang ada pada ruang bagian ini, tepat seperti masukan yang diberikan.
- Posisi (dalam koordinat x dan y) ,arah hadap (kearah $y+$, $y-$, $x+$, atau $x-$, sesuai dengan sumbu), serta nomor identifikasi dari benda atau jalur yang diawasi oleh tiap kamera yang terdapat dalam ruang bagian tersebut.
- Penanda untuk kasus khusus yang bertipe *boolean*, yang menandakan apakah terdapat benda yang titik tengahnya berada pada garis terluar dari sebuah ruang bagian, yang menyebabkan adanya dua atau lebih kamera yang mengawasi benda tersebut, sehingga dapat ditangani lebih lanjut.

Berikut adalah tahap-tahap dari penentuan penempatan kamera pengawas dalam sebuah ruang lingkup :

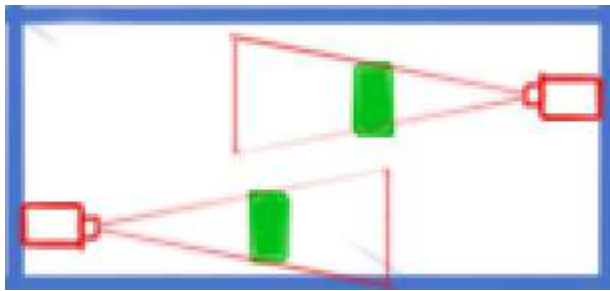
1. Luas pembatas, yakni luas *field of view* dari sebuah kamera pengawas ditentukan terlebih dahulu.
2. Ruang lingkup dimasukkan, lalu seluruh lokasi dari benda-benda penting yang hendak dijaga, serta lokasi dari pintu atau jalur dimana orang dapat keluar dan masuk dari ruang tersebut ditentukan terlebih dahulu.
3. Ruang lingkup dimasukkan, lalu diperiksa terlebih dahulu :
 - a) (*Basis*) Bila lebar ruang lingkup masukan kurang dari dua kali lebar luas pembatas, dan panjang ruang lingkup masukan kurang dari dua kali panjang luas pembatas, maka ruang lingkup tersebut akan dianggap sebuah ruang bagian (*region*), yaitu daun pada pohon *quadtree*. Setiap benda atau jalur yang berada dalam ruang lingkup tersebut dimasukkan kedalam daun tersebut, dan akan diproses pada tahap 4. Hasil “pembagian” dari tahap ini tidak selalu menghasilkan ruang bagian yang “tepat ukuran” dengan *field of view* kamera, oleh karena pembagian dalam metode *quadtree* hanya memeriksa apakah suatu daerah dapat dibagi kembali atau tidak.
 - b) (*Rekursif*) Bila kondisi diatas tidak terpenuhi, maka ruang tersebut akan

- dibagi menjadi empat ruang sama rata, dengan titik tengah dari ruang tersebut sebagai titik temu keempat bagian. Masing-masing bagian tersebut kembali diproses dalam tahap ini.
4. Ruang bagian yang masuk di tahap ini diperiksa terlebih dahulu:
 - a) Apabila tidak terdapat benda yang hendak diawasi atau jalur masuk dan keluar, maka tidak perlu ada kamera pengawas yang diletakkan pada bagian ini.
 - b) Bila kondisi pada tahap (a) tidak terpenuhi, maka dilakukan pemeriksaan atas jumlah benda atau jalur pada ruang ini. Bila hanya terdapat satu benda atau jalur, maka posisi penempatan kamera dapat langsung diperkirakan dengan cara menaruh kamera pada titik tengah dari benda tersebut ditambah (atau dikurangi, tergantung pada posisi kamera terhadap sumbu 0) dengan panjang atau lebar *field of view* kamera, atau pada ujung dari ruang bagian apabila posisi kamera hasil operasi tersebut melebihi batas dari ruang bagian.
 - c) Bila hasil pemeriksaan benda atau jalur pada tahap (b) menghasilkan lebih dari satu buah benda atau jalur, maka titik tengah dari “gabungan” posisi setiap benda dan jalur harus dicari, lalu dilakukan pemeriksaan apakah ada titik tempat penempatan kamera, dimana lebar *field of view* dari kamera dapat melingkupi seluruh benda tersebut, dengan titik tengah hasil “gabungan” tiap benda dan jalur tersebut sebagai *focal point* (titik fokus) dari kamera. Bila cara ini berhasil, maka kamera pengawas diletakkan pada titik tersebut.
 - d) Apabila tahap (c) tidak berhasil, maka harus dilakukan pemeriksaan seperti tahap (c), hanya saja dengan kombinasi lain dari seluruh benda atau jalur yang menjadi masukan (misalkan awalnya benda A-B-C, diulangi lagi dengan A-B dan C secara terpisah, A-C dan B, serta B-C dan A). Kombinasi yang berhasil akan disimpan, dan tidak akan digabung lagi dengan sisa benda atau jalur yang belum ditangani. *Worst-case* yang mungkin terjadi adalah tidak ada kombinasi yang dapat ditangani, selain mengawasi masing-masing benda dengan satu kamera pengawas.
 - e) Apabila terdapat benda (bukan jalur) yang titik tengahnya berada pada ujung ruang bagian, maka penanda untuk kasus khusus diberi nilai *true*, sehingga dapat diproses nanti.
 5. Apabila seluruh ruang bagian telah diproses pada tahap 4, maka dilakukan pemeriksaan pada masing-masing ruang bagian, apakah terdapat kasus khusus. Apabila benar, maka dilakukan pemeriksaan kepada ruang bagian tersebut dan ruang bagian yang menjadi tetangganya. Setelah itu, dilakukan reduksi atas kamera-kamera yang mengawasi benda yang sama. Kamera yang dipertahankan adalah kamera yang juga memiliki benda atau jalur lain yang ditangani pada saat yang bersamaan, sedangkan kamera yang hanya mengawasi benda yang berada di irisan ruang bagian tersebut disingkirkan. Apabila seluruh kamera yang mengawasi benda tersebut tidak memiliki benda atau jalur lain yang sedang diawasi juga, maka akan dipilih satu kamera secara acak untuk dipertahankan, sedangkan yang lainnya disingkirkan.
 6. Setelah semua tahap diatas selesai, maka setiap daun dari *quadtree* yang menjadi representasi ruang “disusun” hingga merepresentasikan ruang lingkup keseluruhan, berikut dengan posisi penempatan kamera pengawas dalam ruang tersebut.

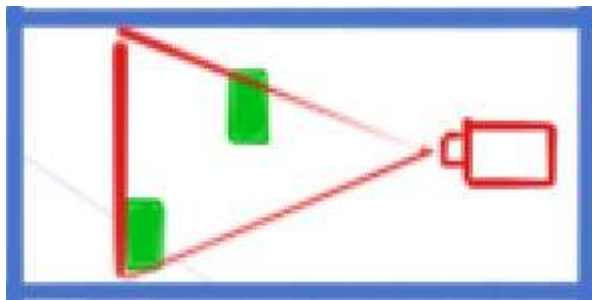


Gambar 3. Proses penempatan kamera pengawas(warna merah) pada ruangan dengan dua pintu dan sebuah benda

(kotak warna hijau). Pertama-tama ruangan dibagi menjadi beberapa ruang bagian (garis warna biru), lalu pada masing-masing ruang bagian yang memerlukan pemantauan, diletakkan kamera pengawas dengan pengaturan yang sesuai.

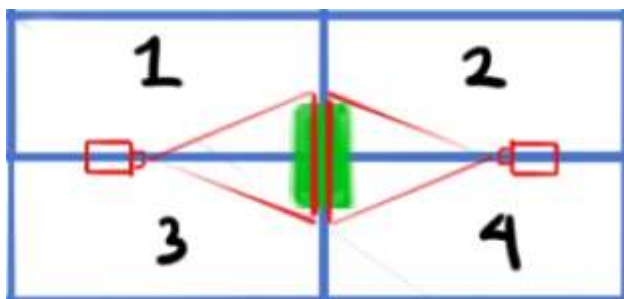


(a)

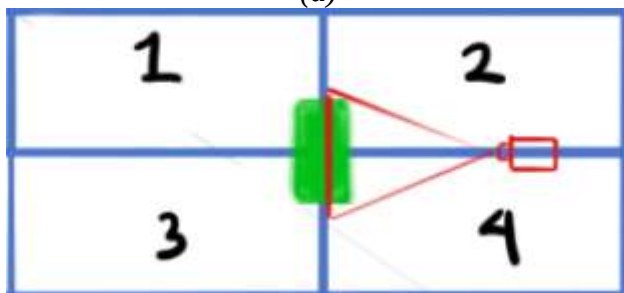


(b)

Gambar 4. Perbandingan pengaruh pengaturan field of view kamera terhadap besarnya ruang bagian dan efisiensi jumlah kamera. Gambar (a) menunjukkan bahwa field of view yang terlalu panjang akan membentuk region yang cukup besar, namun bila terdapat dua atau lebih benda yang penempatannya kurang “baik”, yang terjadi adalah pemborosan resource. Sebaliknya, gambar (b) menunjukkan bahwa field of view yang lebih lebar akan menghasilkan ruang bagian yang hampir sama besar, namun dapat mengawasi lebih banyak benda



(a)



(b)

Gambar 5. Penanganan kasus benda yang terletak pada

lebih dari satu region. Pada gambar (a), terlihat dua buah kamera pengawas, namun sebenarnya itu adalah tumpukan dari kamera region 1 dan 3 (sebelah kiri), dan tumpukan dari kamera region 2 dan 4 (sebelah kanan). Gambar (b) menunjukkan hasil reduksi, yaitu hanya tersisa satu kamera pengawas, yang mungkin berasal dari region 2 atau 4.

IV. KESIMPULAN

Penggunaan penempatan kamera pengawas dalam suatu ruangan dengan algoritma *divide and conquer* dapat berguna banyak apabila dilakukan dengan pengaturan yang tepat. Apabila pengguna dapat menentukan perbandingan *field of view* dari kamera pengawas terhadap luas ruang lingkup, maka kita dapat menentukan penempatan kamera pengawas dalam suatu ruangan secara efektif, dan menggunakannya untuk tindakan-tindakan yang repetitif, misalnya pemasangan kamera pengawas pada ruangan-ruangan dalam gedung bertingkat,

Yang menjadi masalah pada metode ini adalah harusnya pengguna menentukan standar *field of view* terlebih dahulu, yang umumnya berbeda untuk tiap ruangan. Mungkin, metode ini dapat disempurnakan lagi bila menggunakan *octtree* (pemetaan tiga dimensi terhadap ruangan), dan dengan algoritma yang lebih mangkus, yang dapat mengatur *field of view* secara otomatis, sesuai dengan keadaan ruangan pada saat itu.

REFERENSI

- [1] Munir, Rinaldi, *Diktat Kuliah IF3051Strategi Algoritma*, Penerbit Informatika : Bandung, 2009
- [2] <http://www.jvsg.com/cctv-camera-lens-calculations/>, diakses pada tanggal 8 Desember 2011, pukul 20.00 WIB
- [3] http://en.wikipedia.org/wiki/Security_camera, diakses pada tanggal 8 Desember 2011, pukul 20.00 WIB
- [4] <http://en.wikipedia.org/wiki/Quadtree>, diakses pada tanggal 8 Desember 2011, pukul 22.00 WIB

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 9 Desember 2011

Ricardo Pramana Suranta / 13509014