

ALGOTECTURE (*ALGORITHMIC ARCHITECTURE*)

Daniel W. B. Pieter¹ dan Johansen C. Mandey²

¹ Mahasiswa PS S1 Arsitektur Unsrat

² Staf Pengajar Jurusan Arsitektur Unsrat

ABSTRAK

Pada dasarnya Arsitektur merupakan gabungan dari seni dan teknik, tapi saat ini dengan perkembangan digital, peranan komputer dalam proses desain menjadi sangat penting. Arsitektur Algoritmik mengambil peranan penting dalam hal ini, dengan kerumitannya dan pola desainnya yang benar-benar mempesona. Karya tulis ini bertujuan untuk mempelajari tentang Algoritma yang dapat memberikan solusi pada permasalahan desain dalam Arsitektur seperti perletakan ruang, pola struktur, bahkan nilai estetika.

Masalah dari penggunaan Algoritmik dalam desain adalah kepastiannya antara angka dan konsep, untuk beberapa desainer terasa terlalu mengatur. Banyak desainer tidak tertarik pada komposisi Matematis Desain, melainkan hanya pada Komposisi itu sendiri.

Pembahasan ini menyajikan mengenai Pengertian Algoritma, Algoritma dalam Arsitektur, serta beberapa Studi Kasus yang diharapkan dapat membantu para desainer untuk keluar dari permasalahan Matematis Desain.

Dari keseluruhan penulisan diperoleh hasil bahwa, Arsitektur Algoritmik merupakan suatu penerapan Algoritma ke dalam Arsitektur, yang bisa memperluas batas pemikiran manusia dalam menciptakan bentuk geometri non-linear, membantu memecahkan masalah perletakan ruang, bahkan masalah struktur, dengan solusi-solusi alternatif yang berkelanjutan.

Kata kunci: Arsitektur, Algoritmik

1. PENDAHULUAN

Dalam dua dekade terakhir, Arsitektur telah berubah dari desain dengan alat manual menjadi desain bentuk berbasis komputer. Transformasi ini, walaupun mengesankan, belum mencapai potensi penuh. Sebagian karena kurangnya pendidikan komputasi dari Arsitek atau kebanyakan literatur yang membingungkan tentang desain digital, serta hampir tidak ada contoh jelas penggunaan komputer dalam potensi penuh sebagai alat desain.

Perusahaan firma Arsitektur terkenal, seperti *SOM*, *NBBJ*, atau *RTKL*, menggunakan komputer hanya sebagai efisiensi alat sambil terus mengembangkan desain secara tradisional dalam arti manual, dan praktisi-praktisi menonjol, seperti Gehry, Morphosis, atau Zaha Hadid, menggunakan komputer sebagai sarana pemasaran dan presentasi. Tapi ada beberapa Arsitek muda segar yang mungkin dapat menggunakan metode komputasi dalam desain.

Lalu apa yang harus menjadi istilah yang tepat dari keterlibatan komputer dengan desain Arsitektur? Pertanyaan ini telah ada sejak awal dari Arsitektur berbantuan komputer. Dasarnya yaitu saat percobaan pertama mengenai kemungkinan komputasi atau *Cybernetic* Arsitektur di tahun 1950-an dan 1960-an. Namun akhirnya menghilang dan kemudian kembali lagi pada dekade terakhir ini, dengan pengembangan grafis komputer yang spektakuler dan mempesona yang diberikan oleh bentuk-bentuk aneh.

Kostas Terzidis seorang Arsitek sekaligus professor pengajar di Harvard University, mengatakan dalam bukunya bahwa tak ada istilah yang lebih baik dari *Algorithmic Architecture* untuk keterlibatan komputer dalam desain Arsitektur.

Pada dasarnya, posisi mengenai peran komputer dalam desain Arsitektur terbagi menjadi dua kategori besar. Bagi kebanyakan desainer, komputer hanyalah sebuah alat canggih dengan program yang memungkinkan mereka untuk menghasilkan bentuk lanjutan dan kontrol yang lebih baik dalam realisasinya. Bagi mereka yang lain, meskipun mesin mengubah secara signifikan sifat alami dari Arsitektur yang dihasilkan, mereka merasa tidak diperlukan bahkan tidak diinginkan untuk masuk ke dalam rincian prosesnya. Namun bagi para Arsitek seperti *Kostas Terzidis*, tidak dapat dihindari untuk masuk ke dalam kotak hitam pemrograman untuk membuat hal yang benar-benar kreatif dari komputer.

2. PEMBAHASAN

2.1. Pengertian

Menurut Kostas Terzidis, Arsitektur Algoritmik adalah istilah yang diciptakan di sini untuk menunjukkan penggunaan Algoritma dalam Arsitektur. Hal ini dibedakan dari *CAD* atau komputer grafis dalam arti bahwa proses Algoritmik tidak selalu berdasarkan Komputer. Tapi ada beberapa masalah seperti kompleksitas, ketidakpastian, ambiguitas, atau berbagai solusi yang mungkin dibutuhkan hubungan sinergis antara pikiran manusia dan sistem komputer. Sinergi ini hanya mungkin melalui penggunaan strategi Algoritmik yang menjamin komplementer dan dialektika hubungan antara pikiran manusia dan mesin.

Arsitektur Algoritmik adalah suatu area penelitian yang mempelajari tentang aplikasi Algoritma (Ilmu Komputer) di (dalam) Arsitektur. Menurut Kamus Oxford, Algoritma adalah ‘suatu proses atau seperangkat aturan yang harus diikuti dalam operasi perhitungan atau pemecahan masalah’.

Kostas Terzidis dalam bukunya, menjelaskan bahwa Algoritma adalah suatu prosedur komputasi untuk mengatasi masalah dalam jumlah langkah yang terbatas (*finite*). Hal ini melibatkan pemotongan, induksi, abstraksi, generalisasi, dan logika terstruktur. Strategi Algoritmik memanfaatkan pencarian pola yang berulang, prinsip-prinsip universal, modul yang dapat dipertukarkan, dan hubungan induktif. Kekuatan dari sebuah Algoritma terletak pada kemampuannya untuk menyimpulkan dan untuk memperluas batas-batas tertentu dari pemikiran manusia.

Secara teoritis, Algoritma adalah abstraksi dari suatu proses dan berfungsi sebagai pola sekuensial yang mengarah ke pemenuhan tugas yang diinginkan. Misalnya, Algoritma untuk memasak kentang dapat terdiri dari langkah-langkah berikut; Mengupas, Merebus, Memotong, Menyajikan. Jika ada langkah yang terbalik atau ditambahkan atau dihapus, akan memiliki hasil yang berbeda. Hasil ini mungkin lebih baik, sama, atau lebih buruk dari target awal. Namun, seperti dalam memasak, perubahan, keacakan, atau kecelakaan dalam proses dapat menyebabkan solusi baru, dan solusi itu seringkali berbeda secara signifikan dari target awalnya. Dalam kasus ini, Algoritma berfungsi sebagai pola pikir yang membantu dalam memahami masalah, mencari kemungkinan solusi, dan atau sebagai kendaraan untuk mendefinisikan masalah baru.

Meskipun definisi umum dari Algoritma melibatkan kata terbatas, namun tidak berarti bahwa masalah harus dibatasi atau ditentukan. Misalnya dalam dunia Algoritma ada yang disebut sebagai ‘*infinite loop*’ dimana ini merupakan situasi yang dianggap sebagai sebuah kemalangan dan mengakibatkan terhentinya proses pekerjaan. Walaupun langkah-langkah yang menggambarkan ‘*infinite loop*’ mungkin terbatas dan spesifik, tapi situasi yang dihasilkan tak tentu dan tak terbatas.

Misalnya, Algoritma sederhana didefinisikan melalui pernyataan berikut:

```
A = salah; start;
jika A adalah salah maka A = benar;
jika A adalah benar maka A = salah;
go to start;
```

Pernyataan tersebut mengarah ke sebuah argumen siklus yang tak terbatas (*infinite loop*) di mana A berubah antara benar dan salah tanpa akhir. Namun, serangkaian pernyataan memang terbatas, didefinisikan dengan baik, dan akurat.

Perhatikan sekarang Algoritma sederhana berikut:

```
start;
A= nomor acak antara 0 dan 10;
Jika A lebih besar dari 5 maka
berhenti
```

Dalam kasus ini, ada ketidakpastian sementara tentang penggenerasian dan terjadinya angka yang lebih besar dari 5 untuk mengakhiri *loop*. Walaupun pada akhirnya kemungkinan seperti ini hampir pasti, waktu terjadinya tidak selalu demikian, terbatas namun berujung pada perilaku tidak menentu, dan tak terduga.

2.2. Algoritma dalam Desain Arsitektur

Hubungan antara Algoritma, parameter dan output dalam komputasi juga terdapat dalam Arsitektur, misalnya urutan yang menciptakan metode dalam Arsitektur. Ada sifat formalistik tak terlihat dalam Arsitektur yang sesuai dengan Algoritma dan hasil fisik atau bentuk seperti output dalam komputasi. Jadi, kita bisa mengatakan bahwa teori di balik bentuk yang berbeda adalah Algoritma dalam Arsitektur.

Dengan perkembangan teknologi komputer yang drastis, banyak hal tidak mungkin yang dimungkinkan. Perkembangan teknologi menyadari analisis fisika non-linear yang memperluas daerah prediksi dari sistem yang kompleks seperti alam semesta atau fenomena iklim, mengakibatkan pergeseran paradigma akan pemahaman kita tentang ilmu pengetahuan alam. Ada sebuah super komputer yang disebut 'earth simulator' di Jepang merupakan salah satu contoh utama dari kemajuan teknologi komputer tersebut. Super komputer ini dikonfigurasi untuk menemukan dan memprediksi perubahan iklim pada skala global dan digunakan untuk mensimulasikan hal seperti pemanasan global, gerakan kerak bumi, dan sebagainya.

Dari contoh itu kita bisa melihat, kemajuan komputer memberikan kontribusi untuk memperluas pemahaman kita dalam bidang non-linearitas. Dan Arsitektur bukanlah suatu pengecualian karena penggunaan komputer meningkat dengan cepat di lapangan. Jelas dari tren terakhir, bahwa bentuk non-linear dengan Algoritma berkembang besar di daerah teknologi berbantuan komputer. Dalam desain, Algoritma dapat digunakan untuk memecahkan, mengatur, atau mengeksplorasi masalah dengan peningkatan visualisasi atau pengorganisasian kompleksitas.

Algoritma dalam Arsitektur juga berhubungan dengan *Parametric Design*, yang sekarang ini terkenal menghasilkan bentuk-bentukan yang aneh. Pada dasarnya *Parametric Design* adalah metode merancang objek Arsitektural yang berdasarkan pada hubungan dan aturan atau Algoritma dalam komputer. Dengan menggunakan perangkat lunak *Parametric*, kita dapat dengan cepat memanipulasi dan menghasilkan alternatif desain dalam 3D.

Beberapa contoh bangunan dengan Algoritma/parametric design yang terkenal, misalnya Walt Disney Concert Hall (Frank Gehry & Partners), BMW Welt (COOP HIMMELB(L)AU), Beijing National Stadium (Herzog & de Meuron), Guangzhou Opera House (Zaha Hadid), Water Cube (PTW Architects), Yas Hotel (Asymptote) dan lainnya. Parametric design pada dasarnya menggunakan Algoritma dalam urutan pola-pola bentukannya.



Dari contoh bangunan diatas juga, jika diperhatikan dengan baik dapat dilihat bahwa bentuk bangunan terdiri dari pola sama yang berulang, diputar, atau diperbesar sehingga menghasilkan pola yang kelihatannya acak, namun sebenarnya merupakan pola yang teratur. Bentuk tak teratur yang

sebenarnya teratur ini biasanya disebut sebagai Geometri Fraktal. Dilihat dari arti katanya, Fraktal (latin: *fraktus*) berarti patah, rusak, atau tak teratur. Jadi dapat diartikan Geometri Fraktal adalah geometri yang tidak teratur, non-linear, non-euclidean, dan bentuk bangunan yang dihasilkan akan terlihat 'tak teratur' dan susah untuk direalisasikan. Namun sekarang dengan perkembangan pemrograman komputer dan *software* yang semakin canggih, bukanlah tidak mungkin mendapatkan bentuk yang 'tak teratur' ini. Dan untuk itu, Algoritma akan sangat membantu untuk merealisasikannya.

Ada beberapa program lunak yang dapat digunakan untuk menghasilkan ruang dan bentuk dari aturan yang berbasis dan melekat dalam program-program Arsitektur, Tipologi, Kode Bangunan, dan Bahasa itu sendiri. Daripada dengan pemrograman langsung, kodifikasi desain menggunakan scripting tersedia dalam paket 3D (yaitu Maya Embedded Language (*MEL*), *3dMaxScript*, dan *FormZ*) dapat membangun konsistensi, struktur, koherensi, ketertelusuran, dan kecerdasan ke dalam bentuk 3D komputer. Dengan menggunakan bahasa *script*, desainer dapat melampaui 'mouse', melampaui keterbatasan pengaturan pabrik dari perangkat lunak 3D saat ini. Desain Algoritmik tidak menghilangkan perbedaan tapi menggabungkan kedua kompleksitas komputasi dan kreatifitas penggunaan komputer.

2.3. Contoh dasar Algoritma membuat Objek Geometri dan Transformasi (MEL)

Secara umum semua objek disituasikan pada 0,0,0 (x,y,z) origin pada titik tengah objek. Untuk mengganti titik tengah, orientasi, atau ukuran, kita menggunakan perintah *move*, *rotate*, and *scale*. setiap perintah mengambil 3 angka yang menghadirkan nilai x, y, dan z dari transformasi, sebagai contoh:

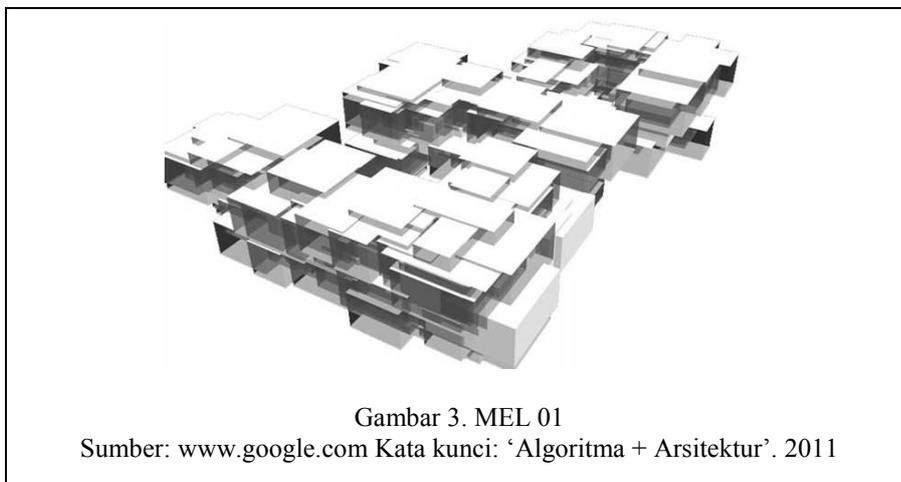
```
sphere -r 1; move 10 3 5.5; rotate 30 0 45; scale 1 2 1
```

Algoritma diatas menghasilkan bola dengan radius 1 kemudian memindahkannya pada lokasi 10, 3, 5.5,(dalam x,y,z) lalu memutarnya 30 derajat pada arah x, 0 derajat pada arah y dan 45 derajat pada arah z, dan kemudian memperbesar ukurannya 2 kali pada arah y, tapi membiarkannya pada arah yang lain.

Dengan *loops*, objek geometri, dan transformasi kita bisa menempatkan objek secara berurutan, membuat *ritme*, pengulangan, atau prosesi. Sebagai contoh, kode berikut akan menghasilkan 12 kubus berukuran 0.5 unit dan menempatkannya pada jarak yang sama 2 unit searah x:

```
for($x=0; $x<12; $x++) {polyCube -w 0.5 -d 0.5 -h 0.5; move ($x*2) 0 0;}
```

Proses Algoritma tersebut seterusnya dilanjutkan dengan bahasa atau kode-kode tertentu sehingga bisa menghasilkan beberapa posisi dan komposisi, dengan bentuk yang beragam, baik arah, skala, dan sebagainya.

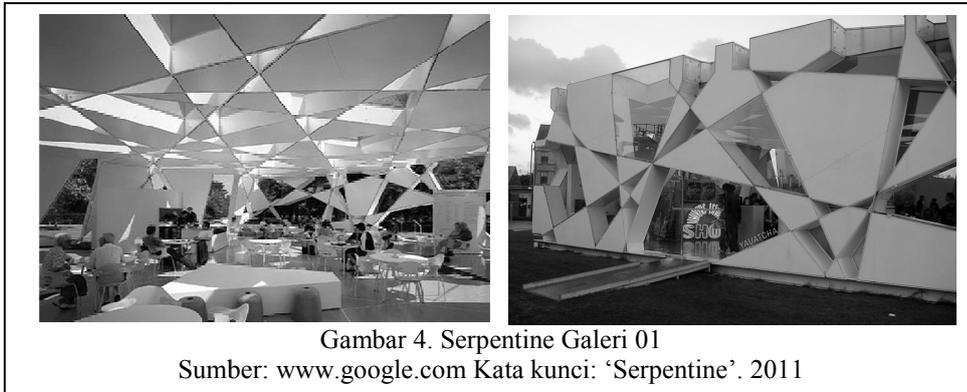


Gambar 3. MEL 01

Sumber: www.google.com Kata kunci: 'Algoritma + Arsitektur'. 2011

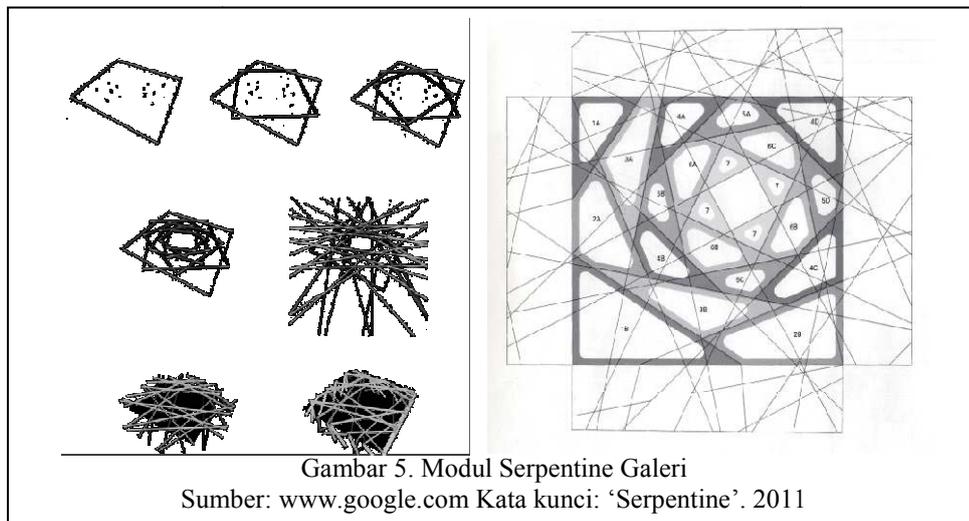
2.4. Studi Kasus

Pada tahun 2002, Toyo Ito diberikan kesempatan untuk membangun Paviliun dari Galeri Serpentine. Paviliun ini akan memiliki luas 300 m² ruang yang tertutup, memiliki kafe dan ruang acara. Ini merupakan proyek yang sangat singkat dimana mereka hanya memiliki tiga bulan untuk mendesain dan tiga bulan untuk membangun.



Dalam proses desain, Toyo Ito bertanya kepada Insinyur Struktur Cecil Balmond tentang 'bagaimana melayangkan atap pelat datar?' dan 'bagaimana membuat ruang tanpa kolom?' Balmond merespon Ito dengan memberikan jumlah sketsa yang sangat banyak. Salah satu contohnya adalah lantai yang akan dinaikan untuk dijadikan struktur yang menopang plat atap persegi, serta *polystyrene* dan panel dengan aluminium.

Kolaborasi antara Ito dan Balmond telah mengekspresikan konsep mereka dalam Serpentine Paviliun. Filosofi strukturnya yang fundamental adalah 'struktur merupakan episode atau momen spesifik dalam urutan proses, struktur sebagai jejak, struktur sebagai aplikasi Algoritma, serta struktur terhubung dengan geometri yang bergerak.

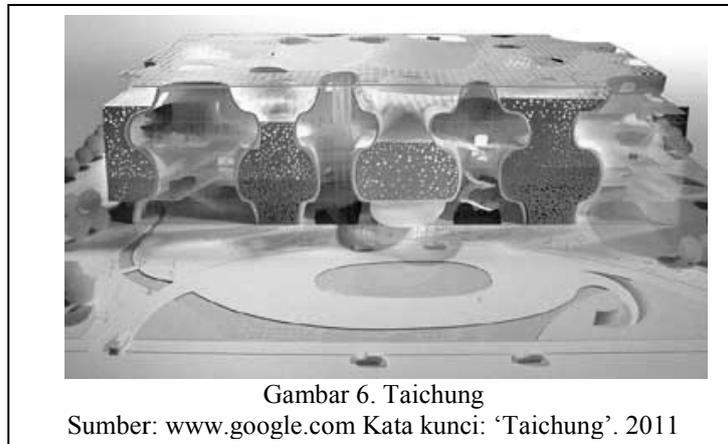


Balmond membuat Algoritma dari apa yang kelihatannya struktur acak. Ini merupakan aplikasi mekanisme Algoritma yang terbuat dari kotak berturut-turut yang ukurannya mengecil saat berputar, kotak yang satu didalam kotak yang lain, terjadi pergerakan spiral dari kotak. Garis struktural terbentuk dari hasil perpanjangan seluruh kotak di bidang ini berturut-turut dan perpanjangan mereka ke arah vertikal dari dinding. Segitiga dan Trapesium diciptakan sebagai hasilnya, dibuat dengan panel aluminium dan panel kaca bergantian. Melalui proses kolaborasi itu, ruang dengan fluiditas diwujudkan.

Banyak proyek terbaru Ito memiliki sistem yang diatur dari bentuk yang diciptakan menggunakan Algoritma yang dapat diperpanjang tanpa batas. Hal ini telah menjadi fitur dimana dapat melihat bagian bentuk berlanjutan seperti itu, dipotong oleh masalah realistis seperti batas-batas dalam tapak atau pembatasan dalam volumenya, yang ditentukan sebagai aturan dalam Algoritma.

Dalam membuat metode Arsitektur ini, ide di dalam dan luar dari ruang tidak ada lagi. Hal ini secara fisik hadir, tetapi secara ideologis tidak ada.

Seperti yang dapat dilihat di bawah, gambar Taichung Metropolitan Opera House ini mirip dengan sistem *grid* oleh Mies Van der Rohe, di mana *grid* menerus bersifat homogen. Namun, Ito berusaha untuk mencapai homogenitas melalui sistem yang sama sekali berbeda. Sebagai sarana untuk mencapai itu, Algoritma bekerja sangat efektif.



Taichung sedang dalam tahap pembangunan. Berlokasi di 7th metropolitan area of Taichung City, Taiwan. Memiliki luas kurang lebih 57,685 m², dengan jumlah 2011 tempat duduk. Didesain oleh Arsitek Jepang, Toyo Ito. Dimulai pada 11 november 2009, konstruksi selama 45 bulan, diprediksi selesai pada tahun 2013.

Ito berpendapat bahwa Arsitektur abad ke-20 adalah yang dicontohkan oleh Le Corbusier dan Mies Van der Rohe. Terutama Le Corbusier, percaya bahwa geometri murni merupakan keindahan terpenting dan bahwa geometri seperti lingkaran, bola, dan kubus adalah bentuk yang paling indah. Juga, Mies Van der Rohe meninggalkan kita dengan ide *grid* dimana prosesi sumbu ortogonal yang memiliki sifat homogen. Ini adalah salah satu dari dua hal yang dibentuk Arsitektur pada abad ke-20.

Di sisi lain, Cecil Balmond mendefinisikan lebih jauh dengan mengatakan 'Geometri hanyalah jalan yang diambil oleh sebuah titik yang bergerak'. Dia mengartikulasikan bahwa lingkaran dan persegi adalah hanya beberapa contoh dari jalan tersebut, dimana solusi tertentu didapat dari titik yang bergerak. Cecil Balmond mengatakan geometri yang dalam Arsitektur tidak hanya bisa dibentuk dengan geometri Euclidean, tetapi bisa direalisasikan melalui geometri Non-linear. Definisi Balmond seperti inilah yang dipercaya Ito telah benar-benar mengubah ide Geometri dalam Arsitektur. Selain itu, Cecil Balmond menganggap apa yang tampaknya bukan geometri adalah geometri.

Cecil Balmond membuat aturan atau prinsip-prinsip melalui Algoritma. Ia menyarankan bahwa ide-ide mewujudkan 'keacakan' yang dapat dipikirkan manusia sangat terbatas dan mengakibatkan ruang mudah menjadi konvensional. Ia percaya bahwa mendasarkan pada Algoritma pada awalnya tampak nyaman, namun sebenarnya sangat bebas dan memungkinkan penciptaan rumit dan hibrid. Situasi yang tak terduga bagi manusia.

Sikap seperti Cecil Balmond menunjukkan pikiran dan ekspresi terhadap Arsitektur, ruang dan matematika sebagai pemikir, atau seorang filsuf daripada Arsitek struktural. Meskipun struktur Arsitektur dan di atas adalah sangat terkait, proses pendekatan analisis strukturalnya adalah unik dan menarik yang membuat Balmond berdiri keluar dari jutaan insinyur struktural saat ini.

3. PENUTUP

Dari pembahasan diatas dapat disimpulkan bahwa, Arsitektur Algoritmik yang berawal dari penggunaan komputer dalam desain ini merupakan suatu penerapan Algoritma ke dalam Arsitektur, yang bisa memperluas batas pemikiran manusia dalam menciptakan bentuk Geometri Non-linear, membantu memecahkan masalah perletakan ruang, bahkan masalah struktur, dengan solusi-solusi alternatif yang berkelanjutan. Contoh percabangan dari *Algoteecture* ini adalah *Parametric Design* dan Geometri Fraktal yang sekarang ini sedang hangat-hangatnya dibicarakan oleh para desainer.

Dalam Arsitektur, Algoritma membuat seorang desainer melompat dan berpetualang ke dalam

dunia baru yang tak diketahui. Ini bukanlah sebuah produk akhir, namun sebagai kendaraan untuk eksplorasi dalam pemecahan masalah. Yang membedakan hal ini dari pemecahan masalah umum adalah sifatnya yang tak dapat diprediksi dan hasil yang mengagumkan bahkan bagi penciptanya sendiri.

Memang hal ini kelihatannya sangat susah, namun Arsitek harus berani mengambil langkah *out of the box*, langkah yang revolusioner, dan dengan lapang hati menerima rasionalisasi desain yang lebih baik.

DAFTAR PUSTAKA

- Mizushima, Naohiro. _____. **Algorithm and Architecture**. _____.
- Terzidis, Kostas. 2006. **Algorithmic Architecture**. Elsevier Ltd, United States.