

Parameterisasi dalam Penggabungan Program Arsitektur

Markus Parcoyo dan Endy Yudho Prasetyo

Departemen Arsitektur, Fakultas Arsitektur, Desain dan Perencanaan, Institut Teknologi Sepuluh Nopember (ITS)

e-mail: endy_yudho_prasetyo@arch.its.ac.id

Abstrak—Pada saat ini kebutuhan manusia dalam beraktivitas dalam suatu konteks telah menjadi semakin kompleks dan dinamis. Kompleks dan dinamis yang dimaksud adalah, aktivitas manusia semakin beragam dan berubah-ubah seiring waktu. Hal ini memicu perubahan dalam arsitektur yang mana respon manusia akan pemenuhan kebutuhannya dalam beraktivitas, untuk berkembang menjadi kompleks dan dinamis. Perkembangan ini selanjutnya ditandai dengan adanya fenomena menggabungkannya dua atau lebih program dalam sebuah arsitektur, *crossprogramming*, *transprogramming*, *disprogramming*. Dalam menggabungkan dua atau lebih program dalam sebuah arsitektur dibutuhkan suatu cara yang efektif sekaligus memberikan hasil yang optimal. Oleh sebab itu, teknik parameterisasi dipilih karena kemampuannya dalam memetakan karakteristik program arsitektur secara terperinci. Teknik ini memetakan setiap karakteristik program arsitektur dengan menggunakan beberapa parameter. Hasil pemetaan data selanjutnya akan digunakan sebagai acuan dasar dalam melakukan perancangan. Pemetaan ini bertujuan untuk memudahkan perancang dalam memberikan respon yang tepat dalam menggabungkan program arsitektur. Sehingga menghasilkan sebuah arsitektur yang kompleks dan dinamis.

Kata Kunci—Arsitektur, Dinamis, Kompleks, Parameterisasi, Program.

I. PENDAHULUAN

KEBUTUHAN manusia dalam beraktivitas dalam suatu konteks telah menjadi semakin kompleks dan dinamis (Gambar 1). Kompleks dan dinamis yang dimaksud adalah, aktivitas manusia semakin beragam dan berubah-ubah seiring waktu. Hal ini memicu perubahan dalam arsitektur yang mana respon manusia akan pemenuhan kebutuhannya dalam beraktivitas [1], untuk berkembang menjadi kompleks dan dinamis. Perkembangan ini selanjutnya ditandai dengan adanya fenomena menggabungkannya dua atau lebih program dalam sebuah arsitektur, *crossprogramming*, *transprogramming*, *disprogramming* [2].

Kebutuhan ruang manusia dalam beraktivitas memiliki peran utama dalam menentukan respon arsitektur. Dipopulerkan oleh Jean Nicolas Louis Durand (1760-1835), tipe ruang yang sejenis membentuk sebuah fungsi bangunan, dan bangunan yang memiliki fungsi serupa membentuk sebuah tipe bangunan [3]. Namun bagaimana jika dibandingkan dengan saat ini, manusia memiliki kebutuhan yang kompleks dan dinamis. Hal ini menyebabkan elemen program dari antara tipe bangunan saling berbenturan dan bergabung menjadi satu. Fenomena ini dinyatakan dengan jelas oleh Bernard Tschumi



Gambar 1. Ilustrasi kompleksitas kebutuhan aktivitas manusia.



Gambar 2. Ilustrasi penggabungan program arsitektur.

(1994) dalam bukunya yang berjudul “*Architecture and Disjunction*”, mengenai munculnya *crossprogramming*, *transprogramming*, dan *disprogramming* dalam program arsitektur [2]. Dengan adanya fenomena tersebut, arsitektur diharapkan mampu meresponnya dengan melakukan penggabungan pada elemen program (Gambar 2).

Dalam perancangannya terdapat tiga masalah utama yang diangkat yaitu mengenai pemetaan konteks perancangan, penggabungan dan penyusunan elemen program arsitektur. Pemetaan konteks perancangan berpengaruh pada seberapa majemuk potensi program yang akan diangkat dan digabungkan. Setelah memiliki data potensi program yang akan diangkat, data tersebut perlu untuk dipetakan sesuai dengan karakteristiknya masing-masing. Program yang telah terpetakan perlu untuk disusun kembali ke dalam sebuah kesatuan dalam arsitektur.

Hal ini memicu dibutuhkannya sebuah parameterisasi desain [4] pada karakteristik elemen program arsitektur. Parameterisasi ini bertujuan sebagai acuan dalam melakukan tahap perancangan arsitektur. Sehingga perancang mampu

Tabel 1.
Data Potensi Program

Komunitas Pengguna	Potensi Program	Area (m ²)
Pemukiman	Area Jogging	1200
	Galeri	200
	Taman	800
	Balkon	100
	Tenance	80
Perdagangan	Area Makan	200
	Bar	40
	Retail	800
	Gudang	200
	R. Muat Barang	50
	R. Penyimpanan Bahan	20
	Area Kerja <i>Indoor</i>	400
	Area Kerja <i>Outdoor</i>	400
	Kantor Rental	160
	R. Rapat	50
Perkantoran	R. Bermain & Bersantai	50
	R. Workshop	60
	Area Diskusi	40
	Auditorium	400
	Resepsionis	16
Pendidikan	Area Baca	120
	Parkir	1500
Transportasi	Pusat Informasi	200
	Area Tunggu	20
	Atm	20
Operasional dan Servis	Atrium	800
	R. Opetator	100
	R. Ahu	60
	Cooling Tower	100
	R. Genset	32
	R. Pompa	32
	Toilet	360

Luasan yang diberikan pada data potensi program di atas berdasarkan standard referensi yang terdapat pada buku "Neufert" dan sejumlah preseden.

dengan cermat memetakan dan menyusun program arsitektur secara optimal, memiliki tingkat guna yang tinggi.

II. METODE PERANCANGAN

A. Pemetaan Konteks Perancangan

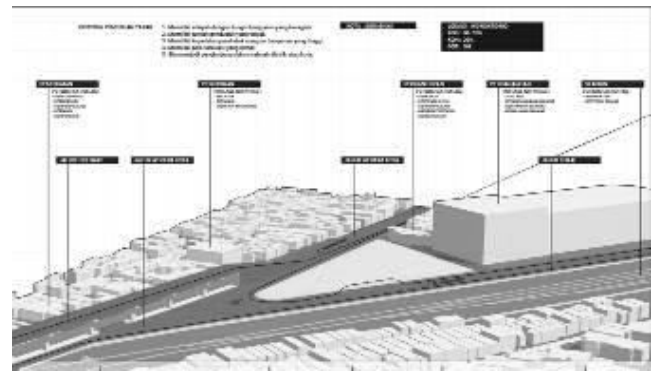
Sebagai ilustrasi dalam pemetaan konteks perancangan, studi konteks yang diangkat adalah kawasan Wonokromo, Surabaya (Gambar 3). Konteks ini dipilih karena memiliki komunitas pengguna yang majemuk, sehingga mampu untuk dijadikan ilustrasi akan kompleksitas potensi program.

Pada awal tahap pemetaan, pengamatan dan analisa mengenai potensi kebutuhan program dari konteks perancangan dipetakan ke dalam tabel potensi program (Gambar 4). Potensi program didapat dari adanya kekurangan fasilitas yang ditujukan untuk menunjang suatu aktivitas dari komunitas pengguna.

Selanjutnya, potensi-potensi program tersebut diberikan parameter nilai area minimumnya. Acuan dalam menentukan besaran nilai tersebut didapatkan dari buku standard dan



Gambar 3. Konteks Perancangan



Gambar 4. Pemetaan Potensi Program yang Dibutuhkan pada Konteks.

preseden yang ideal. Dengan adanya parameter area tersebut, perancang akan dimudahkan dalam mempertahankan efektifitas kegunaan tiap-tiap program (Tabel 1).

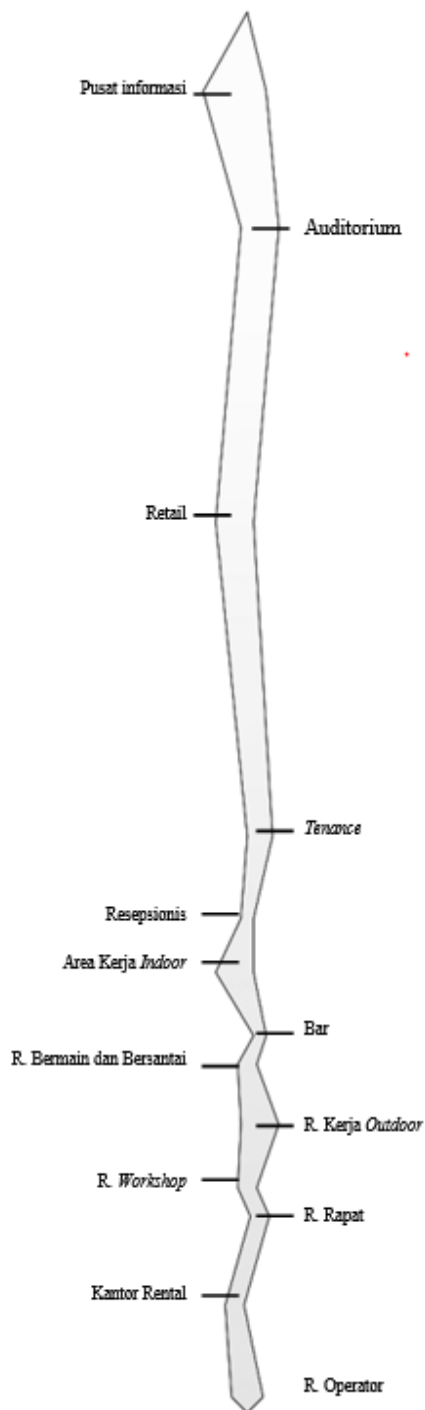
B. Parameterisasi Karakteristik Program

Dalam melakukan parameterisasi ini, perancang perlu untuk menentukan beberapa parameter tertentu untuk memetakan karakteristik setiap program. Pemilihan parameter tersebut dapat ditentukan berdasarkan peta data apa yang ingin kita peroleh. Semakin tepat pemilihan parameter yang diperlukan akan semakin efektif pula kinerja parameterisasi tersebut. Contohnya parameter tingkat privasi digunakan untuk memperoleh data program mana yang bersifat publik dan mana yang privat, waktu penggunaan untuk memperoleh data lama penggunaan dalam satuan jam per hari, dan lain sebagainya

Pada studi kasus ini, parameter yang diberikan antara lain adalah tingkat privasi, waktu penggunaan, ketetapan program terhadap posisinya untuk memperoleh data program mana yang tetap (*fix*) atau tidak tetap (*indeterminance*), dan kedalaman posisinya terhadap perimeter bangunan (*interioritas*) untuk memperoleh data jarak antara program dengan perimeter bangunan. Parameter-parameter ini selanjutnya digunakan untuk merancang gabungan dan susunan program dalam bangunan.

C. Penyusunan Program Tetap

Program tersebut kemudian dipetakan berdasarkan waktu penggunaan dan karakter ketetapanannya. Pemetaan ini digunakan



Gambar 5. Ilustrasi untaian program *fix*.

untuk memisahkan program yang umum, khusus, dan penunjang dalam pembentukan untaian program pada Gambar 5. Dengan adanya penggunaan terus-menerus ini, program-program tersebut membutuhkan aksesibilitas yang mudah, dekat dengan perimeter bangunan. Oleh sebab itu program-program tersebut akan lebih diposisikan pada elevasi dasar. Dan dikarenakan juga beberapa program tersebut merupakan program yang memiliki kategori operasional, aksesibilitas pun akan dibagi menjadi operator dan pengunjung.

Tabel 2.
Penggolongan Karakteristik Program

Digunakan 24/24 jam (umum)	Digunakan 12/24 jam	
	Tetap/Fix (khusus)	Tidak tetap (penunjang)
Area Tunggu	Pusat Informasi	Balkon
Atm	Auditorium	Area Baca
Atrium	Retail	Area Makan
Taman	Tenance	Area Diskusi
Jogging	Resepsionis	Galeri
Toilet	Area Kerja Indoor	R. Penyimpanan Bahan
R. Pompa	Bar	
R. Genset	Ruang Bermain Dan Bersantai	
Cooling Tower	Area Kerja Outdoor	
R. Ahu	R. Workshop	
R. Muat Barang	R. Rapat	
Gudang	Kantor Rental	
Parkir	R. Operator	

Tabel 3.
Parameterisasi Program *Fix*

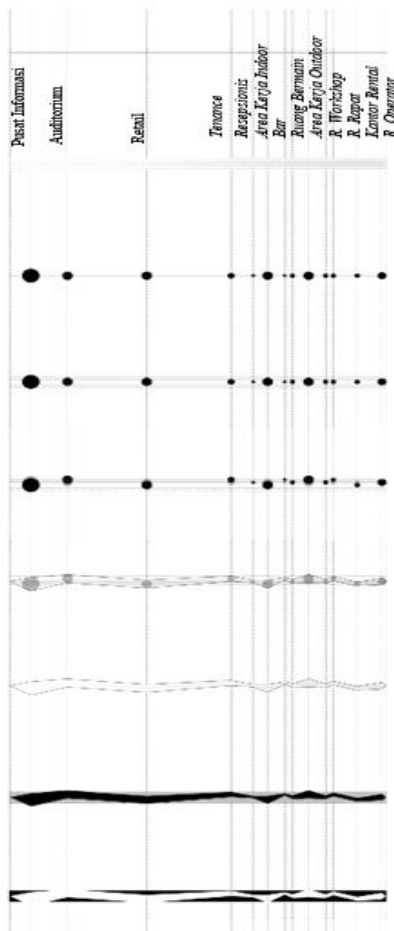
Tingkat Privasi	Program (<i>fix</i>)	Lebar (m)	Jarak (m)	Zona
1	Pusat Informasi	10	25	1
2	Auditorium	6	68	3
3	Retail	6	161	1
4	Tenance	4	260	3
5	Resepsionis	2	286	2
6	Area Kerja Indoor	8	303	1
7	Bar	2	323	3
8	Ruang Bermain Dan Bersantai	3	332	2
9	Area Kerja Outdoor	6	351	3
10	R. Workshop	3	371	2
11	R. Rapat	3	380	3
12	Kantor Rental	3	408	1
13	R. Operator	5	437	2

Data dari tabel ini akan digunakan untuk melakukan pengembangan program pada Gambar 6.

Program yang memiliki karakteristik menetap pada suatu posisi (*fix*) dikelompokkan ke dalam satu susunan untaian program dan disusun berurutan dari publik menuju ke privat. Urutan ini ditujukan untuk memberikan gradasi transisi program dari publik menuju ke privat. Setelah mendapati urutan tersebut, luasan program yang memiliki radius pendek (lebar) dan radius panjang (jarak) diposisikan dalam untaian program sesuai dengan besarnya masing-masing. Hasil dari tahapan ini adalah posisi titik dalam untaian program. Langkah berikutnya adalah menggeser titik-titik posisi program dalam untaian sesuai kode zonasi interioritasnya. Pola garis yang terbentuk kemudian dihubungkan, sehingga didapati perubahan dari untaian garis menjadi untaian bidang.

Sedangkan untuk program yang bersifat tidak menetap (*indeterminance*) akan diletakkan di samping dari hasil untaian program tetap. Titik peletakkannya program ini tetap disusun sesuai keterhubungannya dengan program tetap sebagai program pendukung.

Dalam menguji coba parameterisasi ini, perangkat lunak (*software*) grasshopper digunakan untuk memberikan visualisasi yang akurat mengenai ketergunaan parameterisasi dalam penggabungan program arsitektur. Data pada Tabel 3



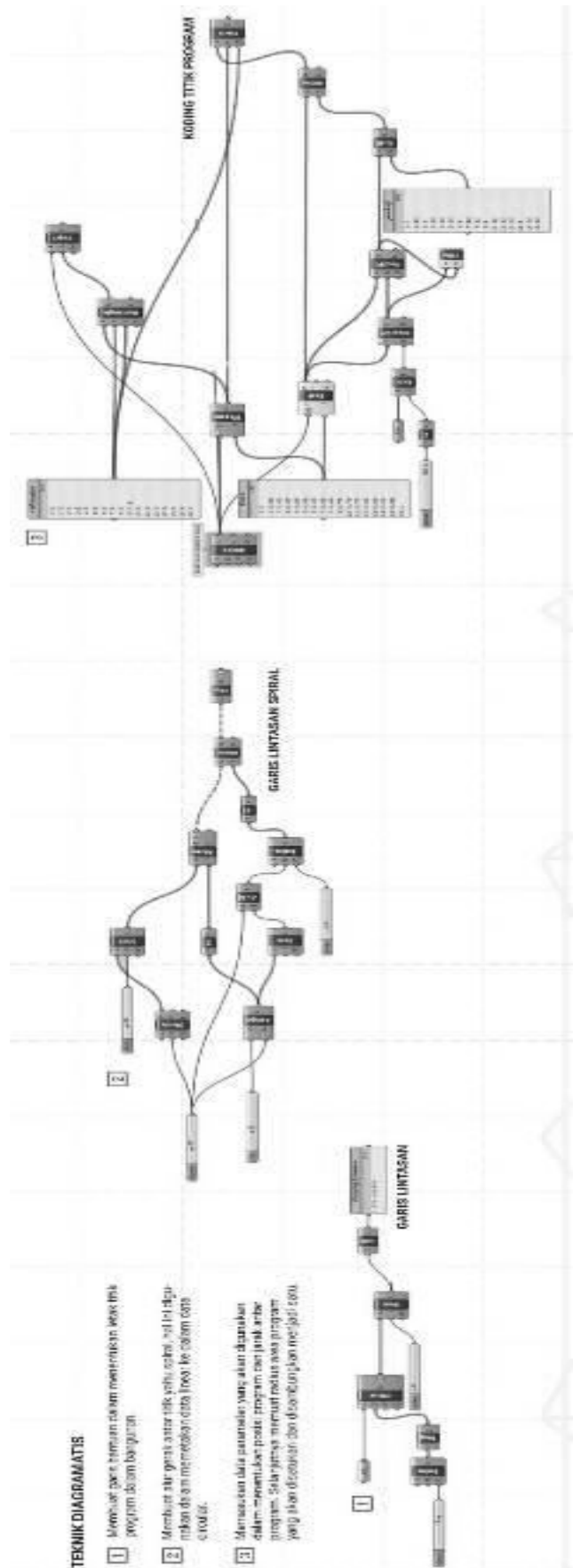
Gambar 6. Translasi parameter ke dalam untaian program.

dimasukan kedalam beberapa perintah di dalam *software* untuk menciptakan untaian program pada Gambar 7.

III. HASIL PARAMETERISASI

Dengan melakukan parameterisasi pada penggabungan program arsitektur terdapat beberapa kelebihan dan kekurangan di dalamnya. Parameterisasi ini memudahkan perancang dalam memberikan setiap respon programatis dengan memberikan acuan terukur berdasarkan karakteristik setiap program. Parameterisasi sendiri memberikan kesempatan pada perancang untuk mengembangkan parameter yang digunakan sekreatif mungkin. Dan ada nilai lebih parameterisasi lainnya yaitu, meskipun perancang dihadapkan pada sebuah permasalahan desain yang sama sekalipun, parameterisasi memberikan kesempatan pada perancang untuk memberikan respon semajemuk mungkin.

Sedangkan pada sisi sebaliknya, keterbatasan perancang dalam mengembangkan parameter yang diperlukan mampu menghambat proses desain itu sendiri. Oleh sebab itu, parameterisasi ini menuntut perancang untuk terus belajar dan berkembang dalam memecahkan setiap permasalahan desain yang ada. Sehingga parameterisasi dalam penggabungan program arsitektur mampu bekerja secara efektif dengan hasil yang optimal.



Gambar 7. Uji coba parameterisasi pada penggabungan program arsitektur.

IV. KESIMPULAN

Dalam memecahkan permasalahan kompleksitas kebutuhan aktivitas manusia, pemilihan respon parameterisasi dalam penggabungan program arsitektur mampu memberikan hasil yang optimal. Parameterisasi memungkinkan perancang untuk bereksplorasi sekreatif mungkin dalam perancangannya. Namun di dalam penerapannya, diperlukan kejelian perancang dalam memetakan permasalahan yang ada. Hal ini ditujukan agar perancang dapat memilih parameter yang tepat, sehingga dapat memunculkan hasil yang dikehendaki. Oleh sebab itu, dalam melakukan parameterisasi ini perancang diharapkan

untuk selalu mengembangkan wawasannya dan melakukan iterasi agar dapat menghasilkan respon yang optimal.

DAFTAR PUSTAKA

- [1] A. Laugier, "Sur L'architecture." Paris, 1753.
- [2] B. Tschumi, *Architecture and Disjunction*. London: MIT Press, 1994.
- [3] P. D. Plowright, *Revealing Architectural Design : Methods, Frameworks, and Tools*. New York: Routledge, 2014.
- [4] K. Jormakka, *Basics Design Methods*. Basel: Birkhäuser, 2008.