

BAB II

TINJAUAN PUSTAKA

2.1 Manual-Based Quantity Takeoff

Quantity takeoff merupakan tahapan pertama dan paling penting dalam proses estimasi biaya maupun penjadwalan rencana proyek (Liu, Lu, & Al-Hussein, 2014). Proses *quantity takeoff* diawali dengan perancang bangunan mengirim gambar rencana proyek kepada *quantity surveyor*. Umumnya gambar rencana yang diberikan berupa gambar dua dimensi atau 2D dalam kertas dan file CAD. Lalu kemudian *quantity takeoff* dilakukan secara manual yaitu menghitung volume geometrik dari komponen bangunan berdasarkan gambar yang diterima.

Dikarenakan estimasi menggunakan gambar 2D secara manual, maka perhitungan akan sangat rawan terjadi kesalahan hitung dan ketidak telitian karena *human-error* (Shick Alshabab, Vysotskiy, Khalil, & Petrochenko, 2017). Walaupun setelah melakukan pengecekan ulang setelah dilakukan estimasi, sulit dipastikan bahwa bagian yang telah revisi sudah benar atau tidak. Ditambah karena estimasi secara manual bergantung pada interpretasi manusia sehingga meningkatkan kemungkinan hasil perhitungan yang kurang akurat. Kesalahan perhitungan dalam proses *quantity takeoff* akan mengakibatkan komponen pekerjaan yang tidak terhitung kedalam estimasi biaya tetapi dalam pembangunan tetap menggunakan biaya, sehingga menimbulkan pengeluaran yang tidak sesuai dengan estimasi (Tong, 2005).

2.2 BIM-based quantity takeoff

Whang, Park, & Min (2016) menyimpulkan bahwa rata - rata persentase akurasi hasil *quantity takeoff* pekerjaan beton dengan BIM adalah 95% sedangkan dengan cara manual sebesar 89%. Kelebihan utama dalam pekerjaan estimasi dengan penggunaan BIM adalah adanya visualisasi. Dikarenakan model 3D proyek dapat dilihat dan diinteraksikan secara langsung, maka estimator dapat lebih memahami proyek yang dikerjakan (Samphaongoen, 2010; Stanley, Thurnell, & Zealand, 2013).

Pemodelan gambar 2D kedalam 3D masih dilakukan secara manual pada BIM, maka masih terdapat kemungkinan terjadinya kesalahan dalam pemodelan. Revisi pada model BIM berasal dari kesalahan pemodelan atau perubahan design akan tetap dikalkulasi secara otomatis pada hasil estimasi. Hal tersebut akan mengurangi tambahan pekerjaan untuk melakukan estimasi dan pengecekan ulang setiap kali ada perubahan pada model proyek (Whang et al., 2016).

2.2.1 Framework Pengambilan Informasi

Model 3D dalam BIM terdiri dari susunan objek atau komponen individu parametrik (Whang et al., 2016). Maka dalam model BIM tiap komponen individu memiliki informasi tentang detail dari komponen tersebut. Informasi yang didapat dalam analisa akan sangat bervariasi seperti dimensi, posisi, material, dll. Sehingga diperlukan perencanaan

framework pengambilan informasi untuk membatasi informasi yang akan diambil dari model BIM.

Framework yang dirancang akan mengandung informasi yang dibutuhkan dalam *quantity takeoff* berupa parameter dari komponen struktur yang ditinjau. Parameter dari komponen sendiri sudah secara langsung terdapat dalam program BIM, terkecuali *formwork* atau bekisting (Liu et al., 2014).

Ma & Liu (2014) merancang *framework* pengambilan informasi dalam proses estimasi dengan basis pembuatan BQ (Bill of Quantity). Maka dapat ditentukan informasi apa yang dibutuhkan dalam *quantity takeoff* berdasarkan pembuatan BQ. Dalam penyusunan data, Liu et al.(2014) memanfaatkan RSMeans berdasarkan MasterFormat sebagai format penyusunan data. Sedangkan Aram, Eastman, & Sacks (2014) merancang *framework* berdasarkan ontology untuk penyusunan data.

2.2.2 LOD (*Level of Development*)

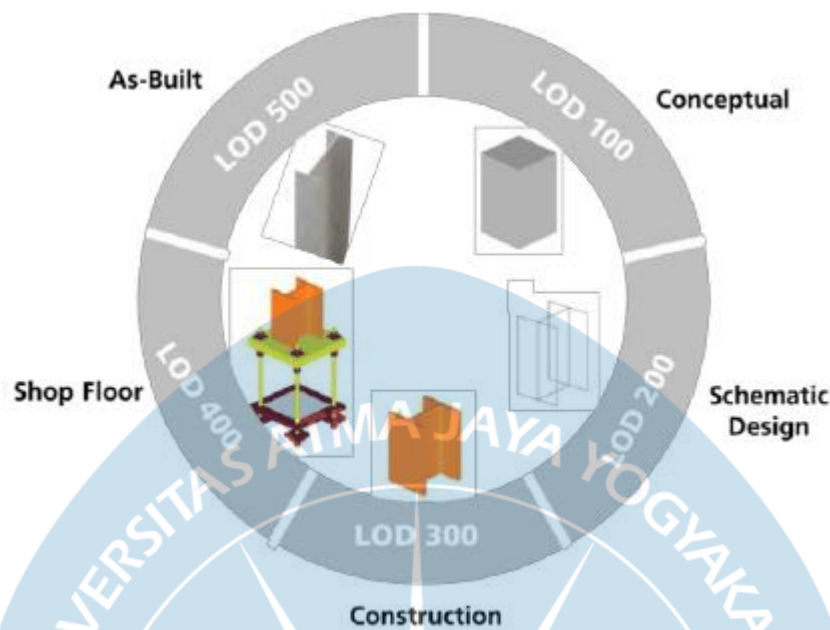
Untuk melakukan estimasi proyek secara detail dengan BIM maka perlu diperhatikan pada tahap design dari gambar rencana proyek apa penggunaan BIM digunakan (Wood, Panuwatwanich, & Doh, 2014). LOD (*Level of Development*) merupakan standard yang menentukan seberapa detail informasi yang sudah terdapat dalam rencana gambar proyek (Clark, 2019). LOD dikembangkan oleh *The Americans Institute of Architects* (AIA) dalam dokumen AIA E202 – 2008. LOD penting diperhatikan sebab

beberapa elemen spesifik dalam model BIM dapat sulit untuk dibedakan bila detail dari gambar masih kurang.

Berdasarkan dokumen E202 AIA, tingkatan LOD dibagi menjadi 5 tingkatan dengan spesifikasi yang berbeda. Tingkatan LOD adalah sebagai berikut :

Tabel 2.1 Tingkatan LOD berdasarkan dokumen E202 AIA

<i>Level of Developement (LOD)</i>	<i>Deskripsi</i>
LOD 100	Tahap awal dimana model elemen hanyalah konseptual tanpa banyak detail. Detail yang diketahui adalah luas, tinggi, volume dan lokasi dari elemen.
LOD 200	Bentuk model dari elemen secara umum tanpa banyak detail, serta dimensi dan jumlah elemen masih dalam perkiraan.
LOD 300	Tiap elemen memiliki bentuk, dimensi, lokasi, dan jumlah yang spesifik. Informasi selain data geometris juga dapat disertakan.
LOD 400	Tiap model elemen sudah spesifik dengan fabrikasi sempurna serta bentuk, dimensi, lokasi, dan jumlah yang akurat.
LOD 500	Model sesuai dengan bentuk asli bangunan secara akurat



Gambar 2.1 LOD dalam BIM (Dallasega et al., 2015)

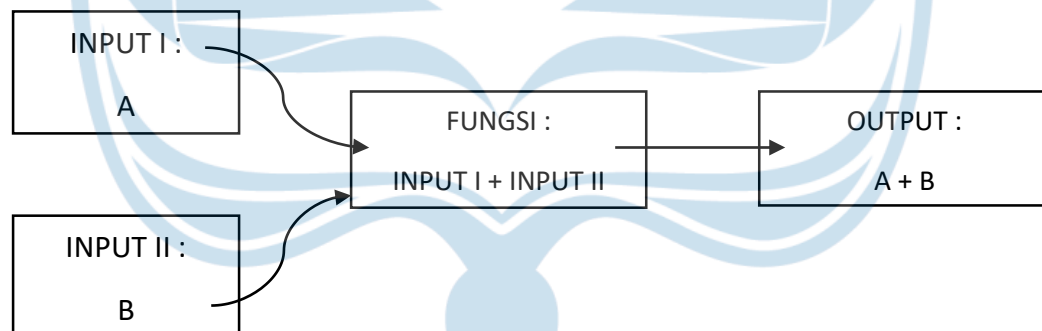
LOD yang digunakan untuk pembuatan model BIM harus memenuhi seberapa banyak kebutuhan informasi dari estimator. Semakin detail pekerjaan estimasi maka akan LOD yang dibutuhkan juga akan semakin tinggi dan detail.

2.3 Komputasi BIM dan *Visual Programming*

Komputasi dalam BIM menggunakan sistem algoritma yang terintegrasi dengan program BIM itu sendiri. Komputasi biasa dilakukan dalam pengambilan data dan manajemen sesuai kebutuhan pengguna. Komputasi BIM umumnya dilakukan menggunakan *visual programming language* (VPL) seperti Dynamo. Dalam penggunaan Dynamo, skrip program dibuat tidak dengan cara coding

manual, tetapi dibuat dengan sistem *flow chart* (Kensek, 2015). Dalam pembuatan skript Dynamo, telah tersedia fungsi-fungsi yang dapat digunakan untuk membuat *workflow* untuk menghasilkan output yang dibutuhkan secara otomatis (Seghier, Ahmad, & Lim, 2019). Contoh output dapat dari komputasi berupa suatu rangkaian pekerjaan atau pengambilan informasi dari elemen yang berjumlah banyak pada model BIM.

Dalam *flow chart* skrip tersebut, terdapat fungsi pekerjaan berbentuk grafis yang disebut dengan *nodes* yang dihubungkan satu sama lain (Ari Monteiro, 2016). Rangkaian dari *nodes* tersebut akan menghasilkan satu atau lebih output pekerjaan. Selain menggunakan *nodes* yang tersedia pada program, pengguna juga dapat merancang *custom nodes* mereka sendiri dengan melakukan coding.



Gambar 2.2 Konsep sederhana dari *nodes* dalam VPL