



Research Article

Perbandingan Waktu Proses Perencanaan Gedung Bertingkat Antara Menggunakan Dynamo dan Metode Konvensional pada Revit

Yoga Kencana Putra¹, Toriq Arif Ghuzdewan^{1,*}, and Akhmad Aminullah¹

¹Universitas Gadjah Mada, Sleman, Indonesia

Received: 11 March 2025, Accepted: 18 August 2025, Published: 9 September 2025

Abstract

Building Information Modeling (BIM) is a computer-based technology that facilitates planning activities across various disciplines, including structural, architectural, and mechanical–electrical works. The implementation of BIM in construction projects requires a high level of accuracy to ensure cost efficiency and smooth workflow. This study aims to examine the time differences in material calculation processes using two different methods: the conventional Material Take-Off feature in Autodesk Revit and a custom plug-in developed using Dynamo. The study employs a 3D model of a three-story building as the test object. Work duration was measured using a stopwatch at the beginning of each calculation process to obtain more precise time records. The results indicate that the use of Dynamo significantly accelerates the calculation process, achieving a fivefold speed improvement compared to the conventional Material Take-Off method. Furthermore, the developed Dynamo script can be applied to projects of the same type but with different geometric forms. Therefore, the proposed script is expected to serve as a practical solution for enhancing accuracy and productivity in concrete work planning for BIM-based construction projects.

© 2025 published by Sriwijaya University

Keywords: *Visual Programming, Quantity take-off, Volume, Building Information Modeling, Structural Design*

1. PENDAHULUAN

Perencanaan merupakan tahapan awal dalam konstruksi. Perencanaan sering mengalami kesalahan-kesalahan minor yang dapat mengakibatkan kerugian. Pada era industri 4.0, penggunaan beberapa aplikasi diharapkan dapat membantu perencana untuk meminimalisir kesalahan minor yang diakibatkan oleh *human error*. Program yang diciptakan dari beberapa aplikasi yang kemudian diintegrasikan agar mempermudah perencanaan disebut sebagai *Building Information Modeling* atau BIM. BIM menggunakan teknologi komputer yang dapat membantu pekerjaan atau tugas-tugas yang mengharuskan perencanaan dilakukan oleh berbagai disiplin ilmu, seperti struktural, arsitektural, ataupun mekanikal dan elektrikal. Hal penting dalam konsep ini adalah inovasi berdasarkan teknologi informasi dan analisis data. BIM memerlukan pembentukan teknologi pemantauan, pengumpulan data, pemrosesan dan pengendalian, serta mengintegrasikan semua sistem informasi [1]. Penggunaan teknologi pemodelan informasi memerlukan proses pembuatan elemen-

elemen bentuk yang kompleks dan penuh informasi [2]. Menurut Hazra [3], pengukuran kuantitas menggunakan CAD memerlukan waktu yang lama, selain itu untuk menghindari kesalahan yang disebabkan oleh perhitungan secara duplikat dan kelalaian, estimator harus menganalisis setiap gambar dan melakukan perhitungan secara presisi. Akan tetapi, BIM memungkinkan pengukuran yang lebih presisi dan lebih cepat sekaligus dapat menurunkan biaya.

Penggunaan BIM dalam perencanaan bangunan menjadi salah satu topik yang menarik dalam dunia konstruksi agar di masa depan BIM dapat digunakan oleh semua pihak dan mempermudah setiap perencanaan yang melibatkan banyak disiplin ilmu.

Pada saat ini, terdapat salah satu dari banyak aplikasi yang dapat melingkupi berbagai macam disiplin ilmu. Aplikasi tersebut bernama Revit yang diluncurkan oleh perusahaan Autodesk. Revit adalah *software* pemodelan 3D untuk memodelkan berbagai elemen bangunan secara realistis sehingga insinyur dapat memvisualisasikan perencanaan tersebut sebelum menjadi gambar [4]. Revit memungkinkan

berbagai macam disiplin ilmu seperti struktural, arsitektural, mekanikal dan elektrikal dikerjakan secara bersama-sama. Revit memiliki banyak fungsi yang dapat membantu perencana selama fase perencanaan seperti *quantity take-off* dan *clash detection*. Dua fitur ini merupakan fitur yang paling sering digunakan pada Revit.

Quantity take-off berfungsi untuk mengeluarkan estimasi kuantiti material yang dibutuhkan ketika proses pembangunan, sedangkan *clash detection* berfungsi untuk melihat benturan antar elemen seperti tulangan, perkabelan, dan pipa, sehingga elemen yang mengalami benturan harus dirubah secara desain agar tidak terjadi penumpukkan material ketika proses pembangunan. *Quantity take-off* merupakan proses peramalan biaya yang diperlukan selama fase desain konstruksi [5]. Dalam proses ekstraksi material atau yang biasa disebut *quantity take-off*, Revit dapat sekaligus mengeluarkan panjang, lebar, luas, bahkan volume dari elemen yang dipilih. Hal ini yang membuat Revit menjadi *powerful* dan menjadi pilihan perencana dalam melakukan perencanaan, dibandingkan harus kembali menggunakan AutoCAD yang penghitungan dilakukan secara manual. Revit dapat mengekstraksi informasi data tersebut ke format Microsoft Excel sehingga perencana tidak menghabiskan waktu dalam proses *quantity take-off*. Akan tetapi, proses *quantity take-off* pada Revit terdapat perbedaan penamaan, contohnya seperti penamaan *level* atau tingkat. Penamaan *level* pada tiap elemen berbeda, seperti pada kolom adalah *base level*, pada balok adalah *reference level*, dan pada plat lantai adalah *level*. Penamaan yang berbeda ini mengakibatkan informasi data level yang kosong pada *family* balok saat dilakukan *quantity take-off*. Oleh karena itu, perlu dilakukan *quantity take-off* secara terpisah agar informasi data yang dibutuhkan pada tiap elemen dapat dikeluarkan secara menyeluruh.

Untuk menutupi kelemahan Revit yang memerlukan perlakuan manual dalam beberapa prosesnya, Autodesk menambahkan *plug-in* bernama Dynamo pada Revit yang dapat membantu mengotomatisasi beberapa hal, termasuk *quantity take-off*. Menurut Thabet [6], alur kerja yang telah disesuaikan akan memungkinkan penghematan waktu, yang tadinya memerlukan waktu lama kemudian dapat dilakukan dalam hitungan menit dengan kombinasi pengumpulan data diseluruh siklus dan mengotomatiskan proses pengisian data. Dynamo merupakan *visual programming* yang dapat disesuaikan berdasarkan *workflow*. Pada Dynamo terdapat beberapa *nodes* yang telah tersedia pada *library* untuk memudahkan perencana dalam menyesuaikan jenis data atau tujuan dari ekstraksi pada Revit. Pengguna juga dapat membuat *node*

khusus menggunakan pemrograman langsung seperti Python dan digunakan pada *script* utama [7]. Terdapat juga beberapa *nodes* yang telah dibagikan oleh *publisher* lain untuk menyesuaikan kebutuhan perencana apabila *nodes* yang disediakan oleh Dynamo tidak mencakup kebutuhan dari perencana. Kelebihan dari Dynamo adalah dapat digunakan berulang kali pada jenis proyek yang sama dengan bentuk yang berbeda. Hal ini dikarenakan Dynamo dapat disimpan dalam bentuk *file* Dynamo dan dapat digunakan dengan memodifikasi *nodes* tertentu.

Penelitian dengan menggunakan Dynamo telah dilakukan Apinut [8] yang menyebutkan bahwa pengintegrasian Autodesk Revit, Dynamo, dan Microsoft Excel serta dengan menggabungkan teori-teori secara kolektif menghasilkan metodologi yang kuat dalam konstruksi dan rekayasa struktur. Khoshkar [9] berpendapat bahwa penggunaan Dynamo tidak hanya dapat mempercepat tahap desain, tetapi juga dapat membantu meningkatkan penyelesaian konstruksi dengan perintah kerja yang realistis.

Dalam penelitian yang telah dilakukan oleh Luz [10], menyebutkan bahwa penggunaan kombinasi pemodelan BIM, Dynamo, dan Python tidak hanya memberikan gambaran tentang sisa material, tapi juga dapat menjadi landasan sebagai inovasi dalam manajemen lingkungan konstruksi.

2. METODE

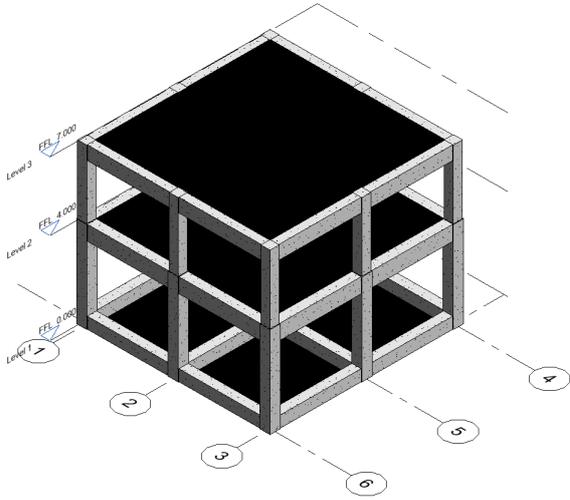
Penelitian ini menggunakan model yang dibuat peneliti sebagai model uji coba dalam penggunaan *plug-in* Dynamo di dalam aplikasi Revit. Pada penelitian sebelumnya, Akcamete [11] telah melakukan penelitian dengan menggunakan otomasi pada bekisting dengan membandingkan waktu pekerjaan antara Revit dan Dynamo.

Model dibuat dengan dua jenis kolom, tiga jenis balok, dan satu jenis plat lantai. Jenis kolom terdiri dari kolom K1 dan kolom K2, sedangkan jenis balok yaitu balok TB, balok B1, dan balok B2. Dimensi dari masing-masing jenis kolom dan balok dapat dilihat pada Tabel 1.

Table 1. Jenis kolom dan balok

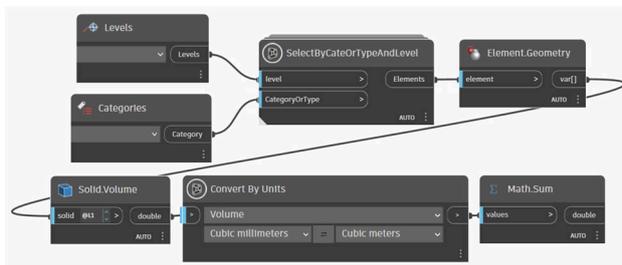
KOLOM		
Nomor	Tipe	Dimensi
1	K1	450X450
2	K2	400X400
BALOK		
1	TB	400X500
2	B1	400X650
3	B2	350X600

Model dibuat secara manual dengan jarak antar kolom yaitu sebesar 4 m. Elevasi tiap level yaitu level 1 pada titik +0.00, level 2 pada titik +4.00, dan level 3 pada titik +7.00. Bentuk model yang akan digunakan pada penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Model 3D

Setelah model telah selesai dibuat, kemudian tahap berikutnya adalah menciptakan script yang bertujuan untuk mengeluarkan jumlah volume dari model yang telah dibuat. Dynamo dapat dipilih melalui tabs *Visual Programming* di bagian *Manage*. Tahapan awal dalam pengembangan script melibatkan ekstraksi setiap jenis elemen konstruksi [10]. Dalam proses pembuatan *Dynamo's script* terdapat beberapa *nodes* dari *publisher* lain yang digunakan oleh peneliti sebagai *nodes* tambahan untuk mendukung pembuatan script. Contohnya adalah *nodes* yang dibuat oleh Genius Loci dengan fitur pemilihan elemen berdasarkan level yang dipilih. *Nodes* ini menyederhanakan bentuk script dalam pemilihan elemen tertentu berdasarkan level yang dipilih. Bentuk dari *dynamo's script* dapat dilihat pada Gambar 2.



Gambar 2. Program Dynamo

Pada penelitian ini terdapat alat dan bahan yang digunakan yaitu:

- a. Alat: satu unit laptop Acer Predator Helios 300 dengan processor: 12th Gen Intel Core i7-

12700H, RAM : DDR5 16GB, & VGA Nvidia RTX 3070Ti,

- b. Bahan: Autodesk Revit 2024, *Plug-In* Dynamo

3. HASIL DAN PEMBAHASAN

1. Perhitungan Volume menggunakan *Material Take-Off* Revit

Penggunaan material *take-off* dapat dilakukan secara *multi-categories* atau dengan dilakukannya pemilihan kategori tertentu. Penggunaan *multi-categories* memiliki beberapa kendala yang dapat mengakibatkan informasi data yang tidak sesuai menjadi kosong. Contohnya dalam penentuan level atau lantai, pada struktural kolom, ekstraksi informasi dapat dipilih melalui pilihan level, sedangkan pada struktural balok menggunakan pilihan *reference level*. Oleh karena itu, peneliti membagi masing-masing pekerjaan struktural untuk dilakukan *material take-off*. Gambar 3 menunjukkan dengan menggunakan *material take-off* yang terdapat pada Revit hasil perhitungan volume kolom pada lantai 1 sebesar 7,266 m³ dan lantai 2 sebesar 4,301 m³, sehingga total volume kolom sebesar 11,567 m³.

<Structural Column>

A	B	C
Base Level	Type	Material: Volume
Level 1	K1	0.810 m ³
Level 1	K1	0.810 m ³
Level 1	K1	0.810 m ³
Level 1	K1	0.810 m ³
Level 1	K1	0.786 m ³
Level 1	K1	0.810 m ³
Level 1	K1	0.810 m ³
Level 1	K1	0.810 m ³
Level 1	K1	0.810 m ³
Level 1: 9		7.266 m ³
Level 2	K2	0.480 m ³
Level 2	K2	0.480 m ³
Level 2	K2	0.480 m ³
Level 2	K2	0.480 m ³
Level 2	K2	0.461 m ³
Level 2	K2	0.480 m ³
Level 2	K2	0.480 m ³
Level 2	K2	0.480 m ³
Level 2	K2	0.480 m ³
Level 2: 9		4.301 m ³
Grand total: 18		11.567 m ³

q

Gambar 3. Hasil *quantity take-off* kolom Revit

Perhitungan volume balok yang terdapat pada Gambar 4 menunjukkan bahwa jenis balok TB pada level 1 memiliki volume sebesar 8,52 m³, level 2 dengan jenis balok B1 memiliki volume sebesar 10,3944 m³, sedangkan level 3 dengan jenis balok B2 memiliki volume sebesar 8,4672 m³. Volume total

pada balok dari ketiga lantai memiliki volume sebesar 27,3816 m³.

<Structural Framing>		
A	B	C
Reference Level	Type	Material: Volume
Level 1	TB	0.7100 m ³
Level 1	TB	0.7100 m ³
Level 1	TB	0.7100 m ³
Level 1	TB	0.7100 m ³
Level 1	TB	0.7100 m ³
Level 1	TB	0.7100 m ³
Level 1	TB	0.7100 m ³
Level 1	TB	0.7100 m ³
Level 1	TB	0.7100 m ³
Level 1	TB	0.7100 m ³
Level 1	TB	0.7100 m ³
Level 1: 12		8.5200 m ³
Level 2	B1	0.9230 m ³
Level 2	B1	0.9230 m ³
Level 2	B1	0.7526 m ³
Level 2	B1	0.7526 m ³
Level 2	B1	0.9230 m ³
Level 2	B1	0.9230 m ³
Level 2	B1	0.9230 m ³
Level 2	B1	0.7526 m ³
Level 2	B1	0.7526 m ³
Level 2	B1	0.9230 m ³
Level 2	B1	0.9230 m ³
Level 2: 12		10.3944 m ³
Level 3	B2	0.7560 m ³
Level 3	B2	0.7560 m ³
Level 3	B2	0.6048 m ³
Level 3	B2	0.6048 m ³
Level 3	B2	0.7560 m ³
Level 3	B2	0.7560 m ³
Level 3	B2	0.7560 m ³
Level 3	B2	0.7560 m ³
Level 3	B2	0.6048 m ³
Level 3	B2	0.6048 m ³
Level 3	B2	0.7560 m ³
Level 3	B2	0.7560 m ³
Level 3: 12		8.4672 m ³
Grand total: 36		27.3816 m ³

Gambar 4. Hasil quantity take-off balok Revit

Perhitungan plat lantai dapat dilihat pada Gambar 5. Perhitungan menunjukkan hasil yang berbeda pada level 3 terhadap level lainnya. Level 1 dan level 2 memiliki volume yang sama besar yaitu 6,9255 m³, sedangkan volume plat lantai level 3 memiliki volume sebesar 7,0176 m³. Oleh karena itu, volume total plat lantai adalah sebesar 20,8686 m³.

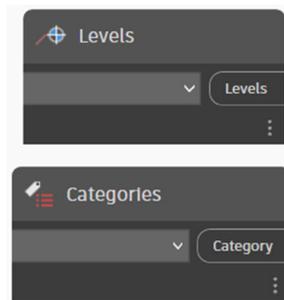
<Floor>		
A	B	C
Level	Type	Material: Volume
Level 1	Slab	6.9255 m ³
Level 1: 1		6.9255 m ³
Level 2	Slab	6.9255 m ³
Level 2: 1		6.9255 m ³
Level 3	Slab	7.0176 m ³
Level 3: 1		7.0176 m ³
Grand total: 3		20.8686 m ³

Gambar 5. Hasil quantity take-off plat lantai Revit

2. Perhitungan Volume menggunakan *Plug-In* Dynamo

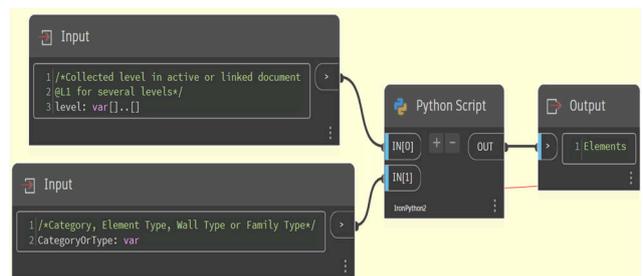
a. Pembuatan *Script* Dynamo

Hal yang perlu diperhatikan pada pembuatan *script* adalah *nodes-nodes* yang digunakan untuk tujuan yang ingin dicapai. Jika tujuan digunakannya Dynamo untuk mengetahui volume suatu elemen tanpa harus dilakukan seleksi *level*, maka *script* yang dibuat cukup dengan menggunakan *nodes categories* dan menghubungkan dengan *nodes* yang lainnya. Akan tetapi, jika ingin melakukan perhitungan volume dengan menentukan elemen serta *level* tertentu, maka dibutuhkan *nodes* tambahan untuk dapat melakukan perhitungan tersebut. Seperti yang terlihat pada Gambar 6 bahwa penelitian ini menggunakan kategori untuk *level* tertentu dalam melakukan perhitungan volume.



Gambar 6. Pemilihan levels and categories

Library dari Dynamo tidak memiliki *nodes* yang bisa langsung menghubungkan dua *nodes* diatas. Oleh karena itu, perlu dilakukan pembuatan *nodes* agar dapat mengubungkan dua *nodes* tersebut. Pembuatan *nodes* dilakukan dengan menggunakan Python Scripts pada *library* Dynamo. Pada Gambar 7, Python Scripts digunakan dengan menghubungkan dua variabel *input nodes* dan satu variabel *output nodes*.



Gambar 7. Proses pembuatan nodes

Proses penulisan *coding* pada Python Script terdiri dari tiga bagian yaitu poin 1, point 2 dan point 3. *Coding* dilakukan untuk memanggil parameter-parameter yang dibutuhkan dari elemen. Coding point 1 ditunjukkan oleh Gambar 8.

```
import clr
clr.AddReference('RevitNodes')
import Revit
clr.ImportExtensions(Revit.Elements)

clr.AddReference('RevitAPI')
from Autodesk.Revit.DB import Category, FamilySymbol,
FilteredElementCollector, Document, RevitLinkInstance,
ElementParameterFilter, ElementMulticlassFilter,
ElementMulticategoryFilter, FilterElementIdRule,
ParameterValueProvider, FilterNumericEquals, ElementId,
BuiltInCategory, BuiltInParameter, Level, HostObjAttributes
import System
from System.Collections.Generic import List

def tolist(obj1):
    if hasattr(obj1, "__iter__"): return obj1
    else: return [obj1]

SelectedLevel = tolist(UnwrapElement(IN[0]))[0]
cats = tolist(UnwrapElement(IN[1]))

doc = SelectedLevel.Document
elem = []
```

Gambar 8. Poin 1

Pada poin 2 ditunjukkan oleh Gambar 9 yang memperlihatkan coding untuk pemilihan kategori dari berbagai kategori yang ada pada Revit.

```
for cat in cats:
    if isinstance(cat, Category):
        catId = List[ElementId](x.Id for x in cats)
        filter = ElementMulticategoryFilter(catId)
        fec = FilteredElementCollector(doc).WhereElementIsNotElementType()
    elif isinstance(cat, System.Type):
        catId = List[System.Type](cats)
        filter = ElementMulticlassFilter(catId)
        fec = FilteredElementCollector(doc)
    elif isinstance(cat, FamilySymbol):
        filter = ElementParameterFilter(FilterElementIdRule(provider, evaluator, cat.Id))
        fec = FilteredElementCollector(doc).OfCategoryId(cat.Category.Id)
    elif isinstance(cat, HostObjAttributes):
        filter = ElementParameterFilter(FilterElementIdRule(provider2, evaluator, cat.Id))
        fec = FilteredElementCollector(doc).OfCategoryId(cat.Category.Id)
    collector = fec.WherePasses(filter)
```

Gambar 9. Poin 2

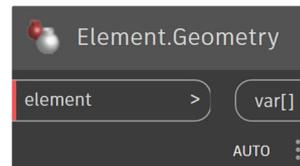
Pada poin 3 yang ditunjukkan oleh Gambar 10 merupakan proses pemilihan elemen pada level yang dipilih.

```
def GetLevel(item):
    val = None
    if hasattr(item, "LevelId"):
        val = item.Document.GetElement(item.LevelId)
        if val: return val
    if hasattr(item, "Level"):
        val = item.Level
        if val: return val
    if hasattr(item, "GenLevel"):
        val = item.GenLevel
        if val: return val
    if (item.GetType().ToString() in ("Autodesk.Revit.DB.Architecture.StairsRun",
    "Autodesk.Revit.DB.Architecture.StairsLanding")):
        item = item.GetStairs()
    if (item.GetType().ToString() == "Autodesk.Revit.DB.Architecture.Stairs" or item.Category.Id ==
    ElementId(BuiltInCategory.OST_Ramps)):
        try: return item.Document.GetElement(item.get_Parameter(BuiltInParameter.STAIRS_BASE_LEVEL_PARAM).AsElementId())
        except: pass
    if (item.GetType().ToString() == "Autodesk.Revit.DB.ExtrusionRoof"):
        try: return item.Document.GetElement(item.get_Parameter(BuiltInParameter.ROOF_CONSTRAINT_LEVEL_PARAM).AsElementId())
        except: pass
    if (item.GetType().ToString() == "Autodesk.Revit.DB.Mechanical.Duct" or "Autodesk.Revit.DB.Electrical.Conduit"):
        try: return item.Document.GetElement(item.get_Parameter(BuiltInParameter.RBS_START_LEVEL_PARAM).AsElementId())
        except: pass
    if hasattr(item, "OwnerViewId"):
        view = item.Document.GetElement(item.OwnerViewId)
        if hasattr(view, "GenLevel"):
            val = view.GenLevel
            if val: return val
```

Gambar 10. Poin 3

Hasil dari nodes baru untuk pemilihan yang membutuhkan kombinasi level dan categories adalah

elemen terpilih. *Nodes Element.Geometry* pada Gambar 11 digunakan untuk mengetahui bentuk dari elemen yang terpilih. Sehingga bisa menentukan *nodes* selanjutnya yang akan digunakan.

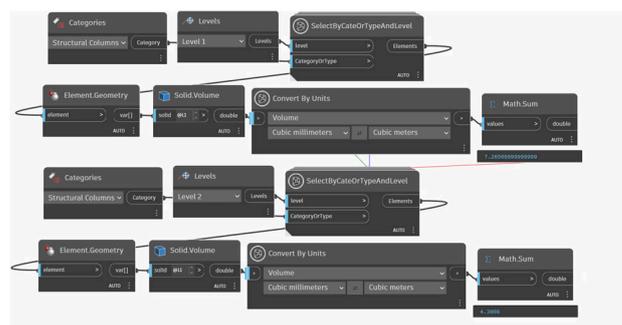


Gambar 11. *Nodes element.geometry*

Setelah *nodes Element.Geometry* mengeluarkan bentuk ruang dari elemen terpilih yaitu *Solid*, maka *nodes* selanjutnya adalah *Solid.Volume*. *Nodes* ini akan mengeluarkan volume dari elemen solid yang telah dipilih. Dynamo mengeluarkan volume dengan satuan mm^3 yang harus diubah menjadi satuan m^3 menggunakan *nodes Convert by Units*, sehingga hasil yang sebelumnya menggunakan satuan mm^3 akan berubah menjadi m^3 . Hasil yang dikeluarkan masih berupa volume tiap satu elemen yang mengharuskan penjumlahan agar menjadi jumlah volume elemen dalam satu *level*. Penjumlahan ini menggunakan *nodes Math.Sum*.

b. Perhitungan Dynamo

Pada penggunaan Dynamo, hal yang harus diperhatikan adalah penggunaan *nodes*. Semakin sedikit penggunaan *nodes*, maka semakin cepat waktu *running*. Pada penggunaan Dynamo untuk menghitung volume struktural kolom seperti pada Gambar 12, jumlah besarnya kubikasi volume sebesar $7,2657 m^3$ pada level 1 dan $4,3008 m^3$ pada level 2. Volume yang dihasilkan oleh *Plug-In* Dynamo sama seperti volume yang dihasilkan oleh *Material Take-Off* Revit.



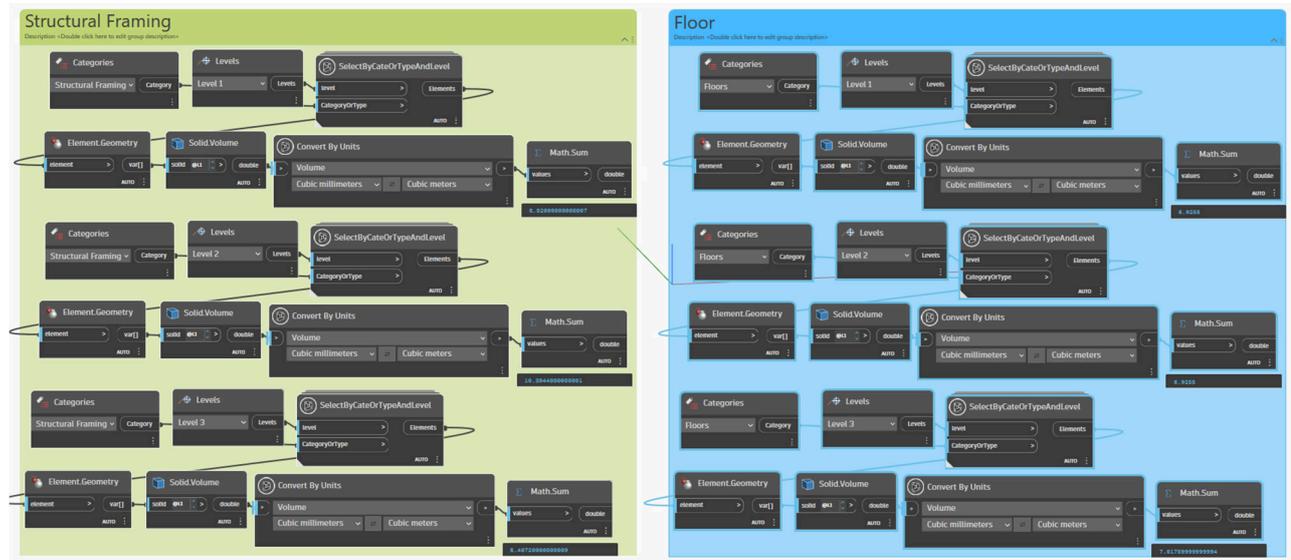
Gambar 12. Perhitungan volume kolom menggunakan Dynamo

Perhitungan volume pada struktural balok dan plat lantai juga bisa dihitung menggunakan *nodes* yang sama seperti pada perhitungan struktural kolom, akan tetapi terjadi modifikasi sedikit pada *nodes* kategori, yaitu *structural columns* diubah menjadi *structural framing* jika ingin menghitung struktural balok, dan diubah menjadi *floors* jika ingin



menghitung volume plat lantai. Terlihat pada Gambar 13, hasil perhitungan *Plug-In* Dynamo menunjukkan angka yang sama seperti perhitungan

Material Take-Off yang memudahkan perhitungan volume dengan hanya mengganti kategori yang akan dilakukan perhitungan.



Gambar 13. Perhitungan volume kolom dan plat lantai menggunakan Dynamo

3. Perbedaan Waktu dalam perhitungan Volume Struktural

Efisiensi waktu pekerjaan akan sangat diperlukan mengingat kecepatan dan ketepatan menjadi peranan utama BIM dalam dunia konstruksi. Perbedaan waktu dalam proses perhitungan akan menjadi salah satu opsi dalam penggunaan BIM dibandingkan perhitungan secara manual. Perhitungan volume yang dilakukan dengan dua metode yang berbeda yaitu *Material Take-Off* dan *Plug-In* Dynamo, menunjukkan bahwa hasil volume yang didapatkan dari dua metode tersebut adalah sama dan dapat dilihat pada Tabel 2.

Table 2. Perbandingan hasil volume antara *quantity take-off* dan Dynamo

No	Jenis Pekerjaan	Level	Volume (m ³)	
			QTO Revit	Dynamo
1	Structural Framing	Level 1	8,5200	8,52
		Level 2	10,3944	10,3944
		Level 3	8,4672	8,4672
2	Structural Colomn	Level 1	7,2657	7,2657
		Level 2	4,3008	4,3008
		Level 3		
3	Floor	Level 1	6,9255	6,9255
		Level 2	6,9255	6,9255
		Level 3	7,0176	7,0176
Total			59,8167	59,8167

Akan tetapi, perbedaan waktu dalam proses pengerjaan dua metode tersebut cenderung

signifikan. Proses pengerjaan Metode *Material Take-Off* banyak menghabiskan waktu karena proses harus dilakukan secara berulang pada setiap kategori, sedangkan penggunaan metode *Plug-In* Dynamo lebih cepat karena hanya perlu melakukan *copy-paste* pada *nodes* dan pemilihan kategori.

Table 3. Perbandingan waktu pengerjaan

No	Jenis Pekerjaan	Waktu	
		Dynamo	QTO Revit
1	Structural Framing	00:00:38	00:01:38
2	Structural Colomn	00:00:38	00:01:15
3	Floor	00:00:38	00:01:27
Total		00:00:38	0:04:20

Terlihat pada Tabel 3, penggunaan Dynamo hanya memerlukan waktu kurang lebih 38 detik dalam melakukan perhitungan, sedangkan pada penggunaan *material take-off* memerlukan waktu sekitar 4 menit. Hal ini disebabkan oleh pengulangan penggunaan *material take-off* dalam pemilihan kategori.

4. KESIMPULAN

Berdasarkan hasil dari penelitian dapat disimpulkan bahwa,

1. Perhitungan volume pada struktural seperti kolom, balok, dan plat lantai tidak bisa dilakukan bersamaan apabila menyematkan

- informasi data *level*. Hal ini dikarenakan oleh perbedaan penamaan *level* pada elemen tersebut.
2. Jumlah volume yang dihasilkan oleh QTO Revit dan penggunaan Dynamo adalah sama, sedangkan waktu yang diperlukan untuk menghitung volume berbeda. Dynamo lebih cepat dibandingkan QTO Revit yang dikarenakan oleh QTO Revit memerlukan pengulangan pada tiap elemen, sedangkan Dynamo hanya menyalin *script* yang sudah ada.
 3. Terdapat perbedaan waktu yang digunakan untuk perhitungan volume beton sampai 5 kali lipat. Dynamo hanya memerlukan waktu selama 38 detik, sedangkan perhitungan menggunakan *material take-off* memerlukan waktu selama 4 menit 20 detik. Hal ini dapat membuktikan bahwa penggunaan Dynamo akan mempersingkat waktu perencanaan.
 4. *Script* yang dikembangkan pada *Plug-In* Dynamo memiliki fleksibilitas penggunaan pada berbagai jenis model, sehingga tidak menghabiskan waktu untuk membuat ulang *script* pada Dynamo. Selain itu, perhitungan volume yang dihasilkan akan secara otomatis menyesuaikan dengan bentuk model 3D yang dibangun.

REFERENSI

- [1] E. Ignatova, S. Zotkin, and I. Zotkina, "The extraction and processing of BIM data," in *IOP Conference Series: Materials Science and Engineering*, Institute of Physics Publishing, 2018. doi: 10.1088/1757-899X/365/6/062033.
- [2] D. Shishina and P. Sergeev, "Revit dynamo: Designing objects of complex forms. Toolkit and process automation features," *Architecture and Engineering*, vol. 4, no. 3, pp. 30–38, 2019, doi: 10.23968/2500-0055-2019-4-3-30-38.
- [3] A. Hazra, S. Bose, A. Chakraborty, S. Qasemi, M. A. A. Obeid, and E. Behruz, "THE BIM-BASED AUTOMATED COST COMPUTATION AND VISUALIZATION OF COST DATA," *Системные технологии*, vol. 1, doi: 10.55287/22275398_2023_1_12.
- [4] D. Nezamaldin, "Parametric design with Visual Programming in Dynamo with Revit," 2019. [Online]. Available: www.kth.se
- [5] T. Tran, L. Hoang, and T. M. Ha, "Dynamo scripts for the digital takeoff of architectural walls in BIM models," 2023, doi: 10.21203/rs.3.rs-2650134/v1.
- [6] W. Thabet, J. Lucas, and S. Srinivasan, "Linking life cycle BIM data to a facility management system using Revit Dynamo," *Organization, Technology and Management in Construction*, vol. 14, no. 1, pp. 2539–2558, Jan. 2022, doi: 10.2478/otmcj-2022-0001.
- [7] Z. Wang, "BIM-based Automated Schedule Generation in Reinforced Concrete-framed Buildings," 2018.
- [8] P. Sittisom et al., "Computer Script Development in Dynamo-Revit Software for Building Foundation Design," *Journal of Architectural/Planning Research and Studies (JARS)*, vol. 22, no. 2, 2025, doi: 10.56261/jars.v22.271117.
- [9] M. Sheikhhoshkar, F. Pour Rahimian, M. H. Kaveh, M. R. Hosseini, and D. J. Edwards, "Automated planning of concrete joint layouts with 4D-BIM," *Autom Constr*, vol. 107, Nov. 2019, doi: 10.1016/j.autcon.2019.102943.
- [10] Luz Fernanda and Perez Palha Rachel, "USE OF THE BIM SYSTEM TO QUANTIFY CIVIL CONSTRUCTION WASTE: USE 2 OF REVIT AND DYNAMO COMBINED WITH PYTHON PROGRAMMING," 2024. [Online]. Available: <https://ssrn.com/abstract=4962174>
- [11] A. Akcamete, Y. Çepni, Y. Cepni, and R. Klein, "Automated BIM-based Formwork Quantity Take-Off," 2020. [Online]. Available: <https://www.researchgate.net/publication/345896054>

