

BAB IV METODOLOGI PENELITIAN

Bab ini menjelaskan tahapan dan metode penelitian yang digunakan dalam proses evaluasi dan optimalisasi tata ruang Terminal Penumpang Pelabuhan Donggala. Penelitian ini menggunakan metode kuantitatif dengan pendekatan desain komputasional, karena penelitian memanfaatkan data numerik berupa luasan ruang, kapasitas ruang, kebutuhan ruang, serta hubungan antar ruang sebagai dasar dalam proses analisis dan optimasi tata ruang terminal. Pada bab ini dibahas proses pengumpulan data berdasarkan dokumen Detail Engineering Design (DED), analisis kondisi tata ruang eksisting, pemodelan bangunan menggunakan Autodesk Revit, penentuan parameter desain, serta pengembangan algoritma parametrik menggunakan Dynamo dalam menghasilkan *relayout* tata ruang terminal yang lebih optimal, efektif, dan efisien. Selain itu, bab ini juga menjelaskan tahapan proses optimasi seperti *space allocation*, hubungan antar ruang (*adjacency*), dan konfigurasi sirkulasi penumpang dalam proses pengembangan desain.

4.1 Kerangka Alur Penelitian

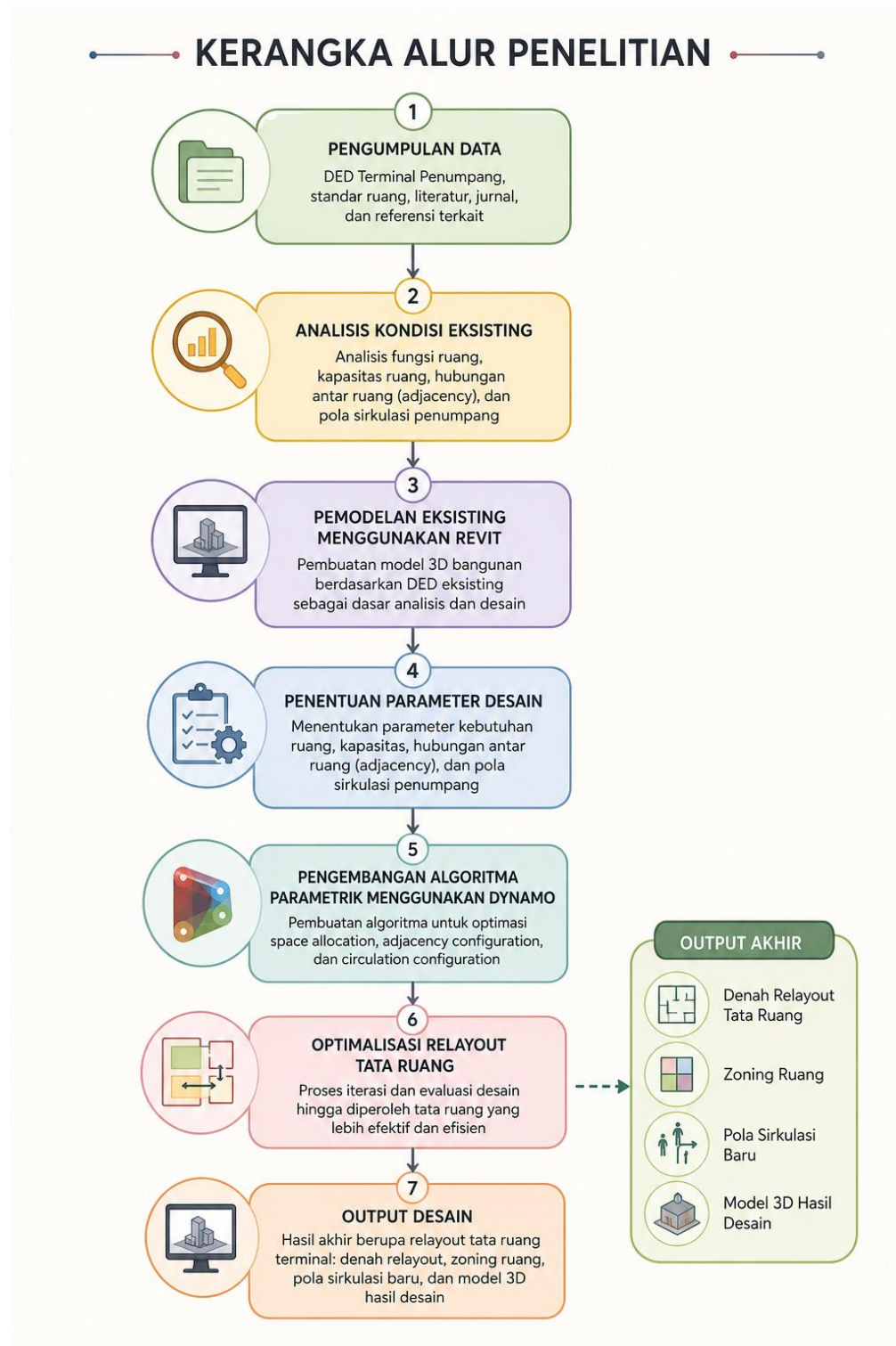
Kerangka alur penelitian disusun sebagai pedoman dalam proses evaluasi dan optimalisasi tata ruang Terminal Penumpang Pelabuhan Donggala melalui pendekatan parametric design berbasis Building Information Modeling (BIM). Penelitian diawali dengan pengumpulan data berdasarkan dokumen Detail Engineering Design (DED) untuk mengetahui kondisi tata ruang eksisting bangunan terminal, meliputi pembagian fungsi ruang, luasan ruang, kapasitas ruang, hubungan antar ruang, serta pola sirkulasi penumpang.

Tahap selanjutnya dilakukan analisis kondisi eksisting untuk mengidentifikasi potensi permasalahan tata ruang seperti distribusi kapasitas ruang yang belum proporsional, hubungan antar ruang yang kurang efektif, serta potensi bypassing circulation pada bangunan terminal. Hasil analisis tersebut kemudian digunakan sebagai dasar dalam menentukan parameter desain yang meliputi kebutuhan luas ruang, kapasitas pengguna, hubungan antar ruang (*adjacency*), dan pola sirkulasi penumpang.

Setelah parameter desain ditentukan, bangunan eksisting dimodelkan menggunakan Autodesk Revit sebagai dasar pengembangan desain secara digital. Selanjutnya dilakukan pengembangan algoritma parametrik menggunakan Dynamo untuk menghasilkan proses optimasi tata ruang melalui pengaturan ulang distribusi ruang, hubungan ruang, dan pola sirkulasi penumpang secara lebih sistematis.

Output akhir dari penelitian ini berupa relayout tata ruang Terminal Penumpang Pelabuhan

Donggala yang telah dioptimalkan berdasarkan kapasitas ruang, efisiensi hubungan antar ruang, dan pola sirkulasi penumpang. Hasil optimasi divisualisasikan dalam bentuk denah hasil relay layout, zoning ruang, pola sirkulasi baru, serta model tiga dimensi bangunan hasil pengembangan desain. Untuk memperjelas tahapan penelitian, kerangka penelitian ini akan disajikan dalam bentuk diagram alur penelitian.



Gambar 4. 1 Kerangka Alur Penelitian

4.2 Pengumpulan Data

4.2.1 Sumber data

Penelitian ini menggunakan data sekunder sebagai sumber utama dalam proses evaluasi tata ruang eksisting dan pengembangan *layout* desain Terminal Penumpang Pelabuhan Donggala.

Data sekunder yang digunakan dalam penelitian ini meliputi:

1. Dokumen Detail Engineering Design (DED)

Data utama penelitian diperoleh dari dokumen Detail Engineering Design (DED) Terminal Penumpang Pelabuhan Donggala yang disediakan oleh PT Witteveen Bos. Dokumen ini meliputi gambar denah bangunan, pembagian fungsi ruang, luasan ruang, layout bangunan, serta informasi teknis lainnya yang digunakan untuk menganalisis kondisi tata ruang eksisting dan proses pemodelan bangunan.

2. Data Pendukung Lokasi

Data pendukung terkait kondisi umum terminal diperoleh melalui sumber informasi daring seperti Google Maps, dokumentasi visual, dan sumber informasi publik lainnya untuk mengetahui lokasi terminal dan kondisi umum kawasan sekitar.

3. Standar dan Regulasi

Data standar kebutuhan ruang diperoleh dari regulasi Kementerian Perhubungan, Direktorat Jenderal Perhubungan Laut, serta standar perencanaan ruang yang digunakan sebagai acuan dalam mengevaluasi kesesuaian tata ruang eksisting.

4. Literatur dan Jurnal Ilmiah

Data pendukung lainnya diperoleh dari buku, jurnal nasional, jurnal internasional terindeks Scopus, dan penelitian terdahulu yang berkaitan dengan terminal penumpang pelabuhan, tata ruang, sirkulasi, kapasitas ruang, Building Information Modeling (BIM), *parametric design*, dan Dynamo. Seluruh data tersebut digunakan sebagai dasar dalam proses analisis kondisi eksisting, penentuan parameter desain, pemodelan bangunan menggunakan Autodesk Revit, serta optimalisasi *layout* tata ruang menggunakan Dynamo.

4.2.2 Teknik Pengumpulan Data

Teknik pengumpulan data dalam penelitian ini dilakukan melalui pengumpulan data sekunder yang digunakan sebagai dasar dalam mengevaluasi tata ruang eksisting Terminal Penumpang Pelabuhan Donggala serta proses pengembangan *layout* desain. Adapun teknik pengumpulan data yang digunakan meliputi:

1. Studi Dokumen

Studi dokumen dilakukan dengan mengumpulkan dan menganalisis dokumen Detail Engineering Design (DED) Terminal Penumpang Pelabuhan Donggala. Dokumen tersebut meliputi gambar denah bangunan, pembagian fungsi ruang, luasan ruang, serta data teknis bangunan yang digunakan untuk mengidentifikasi kondisi tata ruang eksisting dan sebagai dasar pemodelan bangunan pada Autodesk Revit.

2. Studi Literatur

Studi literatur dilakukan dengan mengumpulkan berbagai referensi dari buku, jurnal ilmiah, serta penelitian terdahulu yang berkaitan dengan terminal penumpang pelabuhan, tata ruang, sirkulasi, kapasitas ruang, Building Information Modeling (BIM), *parametric design*, dan penggunaan Dynamo.

3. Studi Standar Kebutuhan Ruang

Pengumpulan data juga dilakukan melalui studi terhadap standar kebutuhan ruang terminal penumpang yang bersumber dari regulasi Kementerian Perhubungan, Direktorat Jenderal Perhubungan Laut, serta referensi standar perancangan ruang. Data ini digunakan untuk mengevaluasi kesesuaian luas ruang eksisting terhadap kebutuhan ruang ideal.

4.3 Pengolahan data

4.3.1 Analisa dan Penggunaan Software

Dalam mendukung proses analisis dan pengembangan *relayout* tata ruang Terminal Penumpang Pelabuhan Donggala, penelitian ini menggunakan beberapa alat dan perangkat lunak yang berfungsi untuk pengolahan data, pemodelan bangunan, serta optimasi desain. Alat dan perangkat lunak yang digunakan dapat dilihat pada tabel berikut:

Tabel 4. 1 Alat dan Perangkat Lunak Penelitian

No	Alat/Software	Fungsi
1	Laptop/Komputer	Digunakan sebagai perangkat utama dalam proses pengolahan data, pemodelan, dan simulasi desain
2	Dokumen DED Terminal Penumpang Pelabuhan Donggala	Sebagai data utama berupa denah bangunan, luasan ruang, dan informasi teknis bangunan

3	Autodesk Revit	Digunakan untuk memodelkan bangunan eksisting ke dalam bentuk digital berbasis BIM
4	Dynamo	Digunakan untuk mengembangkan algoritma parametrik dan menghasilkan alternatif <i>layout</i> tata ruang
5	Google Earth / Google Maps	Digunakan untuk memperoleh informasi lokasi dan kondisi kawasan sekitar terminal
6	Microsoft Excel	Digunakan untuk menghitung kebutuhan ruang, kapasitas penumpang, dan pengolahan data numerik
7	Microsoft Word	Digunakan untuk penyusunan laporan penelitian

Penggunaan alat dan perangkat lunak tersebut mendukung seluruh tahapan penelitian mulai dari pengumpulan data, analisis kondisi eksisting, pemodelan bangunan, pengembangan algoritma parametrik, hingga menghasilkan desain akhir berupa optimalisasi *layout* tata ruang terminal.

4.3.2 Analisis Kondisi Eksisting

Analisis kondisi eksisting merupakan tahap awal yang dilakukan untuk memahami kondisi tata ruang Terminal Penumpang Pelabuhan Donggala berdasarkan dokumen Detail Engineering Design (DED). Tahap ini bertujuan untuk mengidentifikasi kondisi tata ruang eksisting sebagai dasar dalam proses optimalisasi *layout* desain.

Analisis dilakukan dengan mengkaji gambar denah pada dokumen DED untuk mengetahui pembagian fungsi ruang, luas masing-masing ruang, hubungan antar ruang, serta pola sirkulasi penumpang di dalam bangunan terminal. Selanjutnya dilakukan evaluasi terhadap tata ruang eksisting dengan membandingkan luasan ruang yang tersedia dengan standar kebutuhan ruang terminal penumpang berdasarkan kapasitas pengguna.

Pada tahap ini juga dilakukan analisis pola sirkulasi penumpang untuk memahami alur pergerakan pengguna dari area masuk terminal, proses check-in, pemeriksaan keamanan, ruang tunggu keberangkatan, hingga akses menuju kapal. Analisis ini digunakan untuk mengidentifikasi potensi ketidakefisienan sirkulasi, konflik pergerakan pengguna, serta

ketidaksesuaian fungsi ruang.

Hasil dari tahap analisis kondisi eksisting kemudian digunakan sebagai dasar dalam penentuan parameter desain yang akan diterapkan pada proses pemodelan bangunan menggunakan Autodesk Revit dan optimasi tata ruang menggunakan Dynamo.

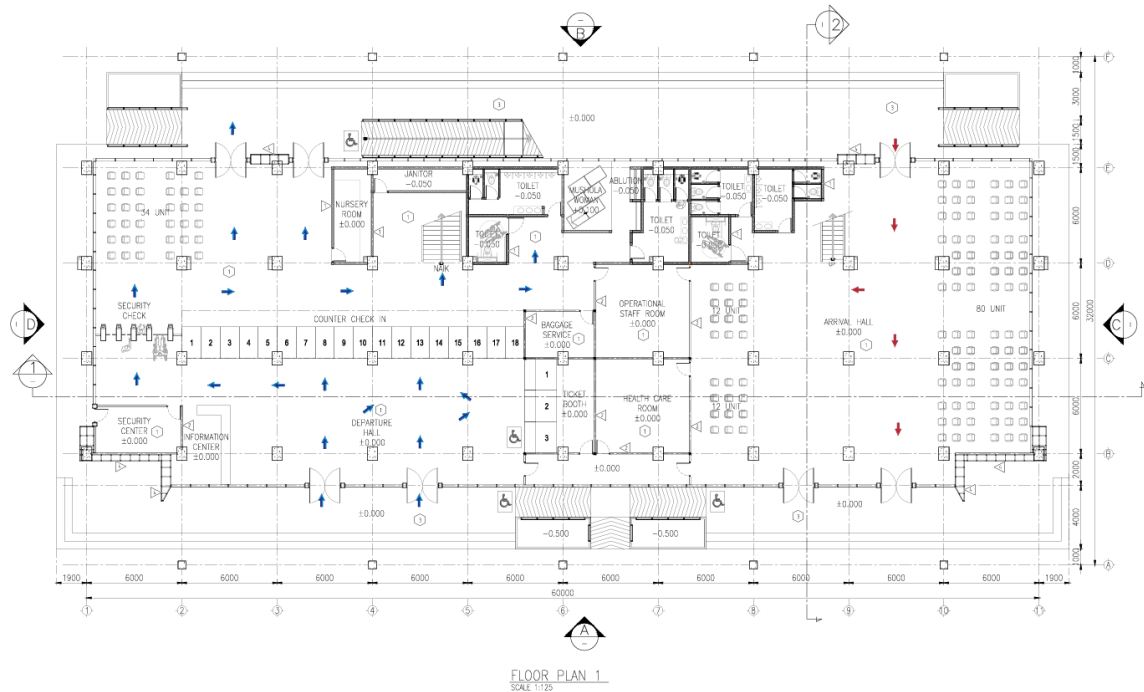
4.3.3 *Permodelan Eksisting Building Menggunakan Revit*

Tahap pemodelan *existing building* dilakukan menggunakan Autodesk Revit sebagai bagian dari penerapan Building Information Modeling (BIM) dalam penelitian ini. Pemodelan dilakukan dengan mentransformasikan data dua dimensi yang diperoleh dari dokumen Detail Engineering Design (DED) Terminal Penumpang Pelabuhan Donggala ke dalam model bangunan tiga dimensi berbasis BIM.

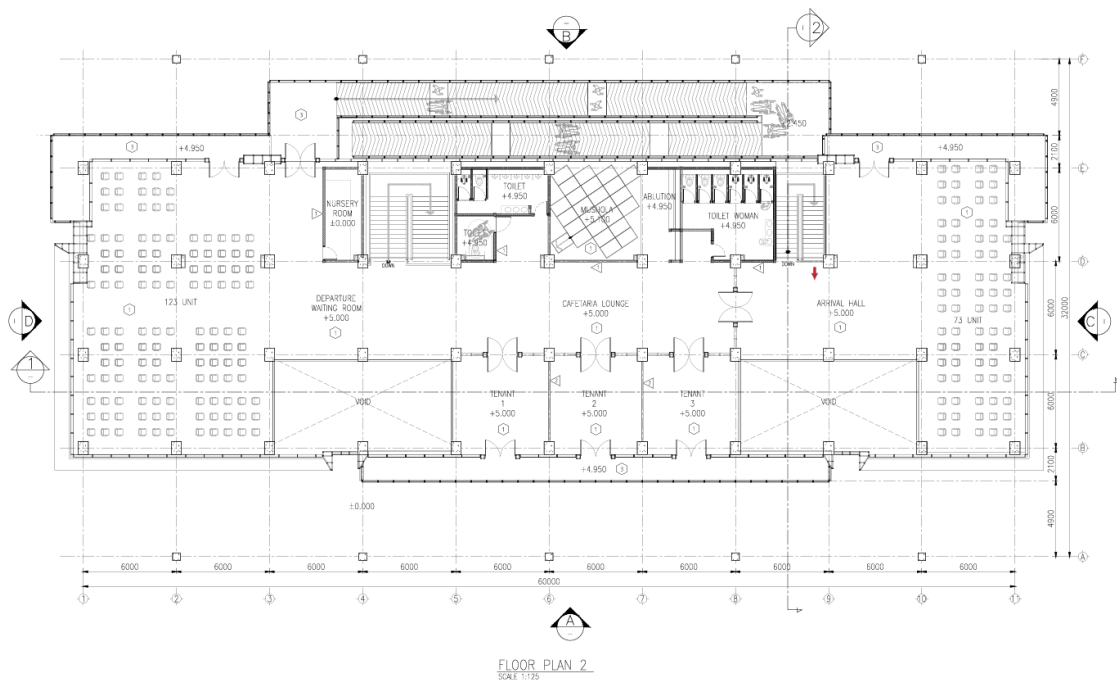
Proses pemodelan diawali dengan memasukkan gambar denah DED ke dalam Autodesk Revit sebagai acuan dasar dalam pembuatan model. Selanjutnya dilakukan pemodelan elemen-elemen bangunan seperti dinding, lantai, kolom, bukaan, serta pembagian ruang sesuai kondisi eksisting terminal. Pada tahap ini juga dilakukan identifikasi luasan masing-masing ruang serta batas ruang (*room boundaries*) yang nantinya digunakan dalam proses analisis kebutuhan ruang dan pengembangan algoritma parametrik pada Dynamo.

Model bangunan eksisting yang telah dibuat digunakan untuk merepresentasikan kondisi aktual tata ruang terminal secara digital sehingga mempermudah proses evaluasi ruang, hubungan antar ruang, dan pola sirkulasi pengguna. Selain itu, model ini juga berfungsi sebagai dasar integrasi dengan Dynamo dalam proses optimalisasi *relayout* tata ruang.

Untuk memperjelas tahapan pemodelan, pada bagian ini akan disajikan beberapa dokumentasi visual berupa hasil pemodelan denah, model tiga dimensi bangunan eksisting, serta tampilan pembagian ruang pada Autodesk Revit yang digunakan dalam penelitian.



Gambar 4. 2 Denah Eksisting Lantai 1



Gambar 4. 3 Denah Eksisting Lantai 2

4.3.4 Identifikasi Parameter Dynamo

Tahap identifikasi parameter dilakukan untuk menentukan variabel yang akan digunakan dalam proses optimasi *relayout* tata ruang menggunakan Dynamo. Pada tahap ini, parameter disusun berdasarkan hasil analisis kondisi eksisting, standar kebutuhan ruang terminal penumpang, serta hubungan aktivitas antar ruang pada Terminal Penumpang Pelabuhan

Donggala. Parameter tersebut berfungsi sebagai input dalam pengembangan algoritma parametrik pada Dynamo sehingga sistem dapat menghasilkan alternatif konfigurasi ruang yang lebih optimal sesuai kebutuhan terminal. Parameter yang digunakan dalam penelitian ini meliputi kapasitas pengguna, kebutuhan luas ruang, hubungan antar ruang, pola sirkulasi, serta batas bangunan eksisting. Berikut parameter yang digunakan dalam proses optimasi desain:

Tabel 4. 2 Parameter Proses Optimasi Penelitian

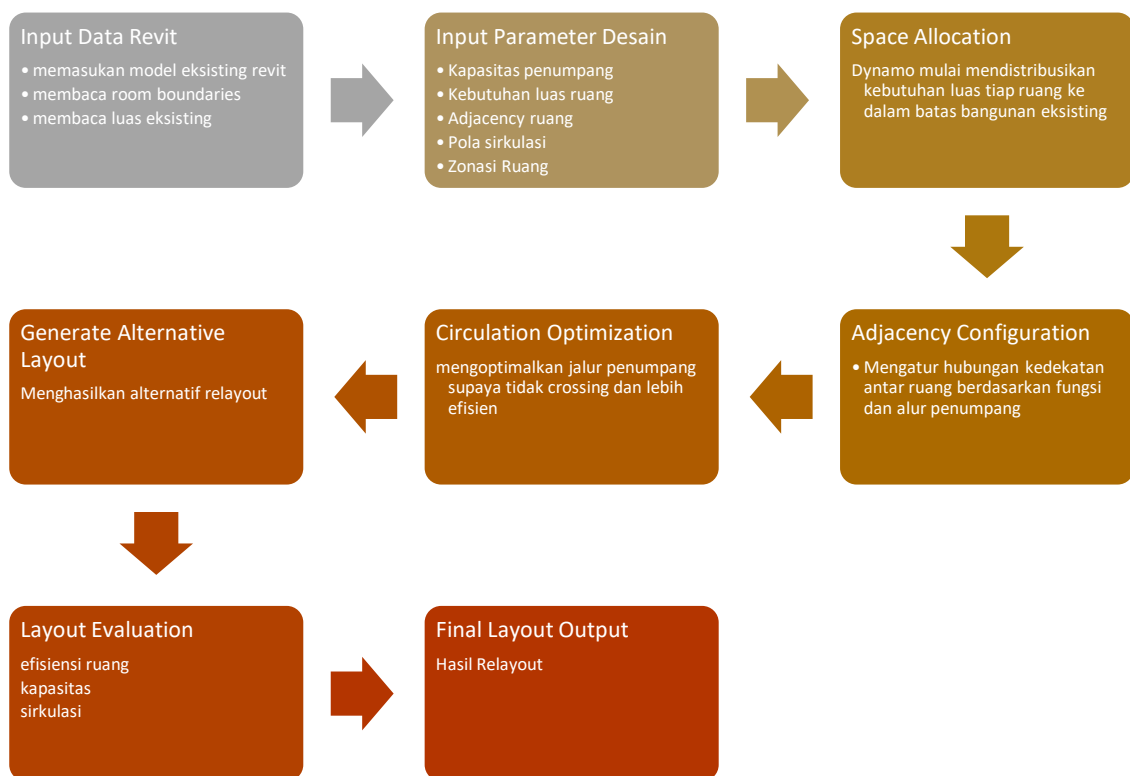
No	Parameter	Indikator	Sumber Data	Fungsi dalam Dynamo
1	Kapasitas Penumpang	Jumlah penumpang harian / peak hour	Data terminal/standar	Menentukan kapasitas ruang
2	Kebutuhan Luas Ruang	Luas minimum tiap ruang	Peraturan Menteri Perhubungan Nomor PM 37 Tahun 201, literatur	Menentukan ukuran ruang
3	Jenis Ruang	Departure, arrival, check-in, waiting room, dll	DED	Menentukan pembagian fungsi ruang
4	Hubungan Antar Ruang	Kedekatan antar fungsi ruang	Analisis aktivitas	Menentukan adjacency layout
5	Pola Sirkulasi	Jalur pergerakan penumpang	Analisis existing	Mengoptimalkan alur sirkulasi
6	Batas Bangunan	Dimensi existing building	DED/Revit model	Membatasi area relayout
7	Zonasi Ruang	Publik, semi publik, privat	Analisis ruang	Mengatur pengelompokan ruang

Parameter-parameter tersebut kemudian diinput ke dalam algoritma parametrik pada Dynamo yang terintegrasi dengan model bangunan di Autodesk Revit untuk menghasilkan beberapa alternatif *relayout* tata ruang yang lebih efisien dan sesuai kebutuhan operasional terminal.

4.4 Tahapan Perancangan Relayout dengan BIM dan Dynamo

Pada tahap ini dilakukan pengembangan algoritma parametrik menggunakan Dynamo yang terintegrasi dengan model bangunan eksisting pada Autodesk Revit. Algoritma dikembangkan berdasarkan parameter desain yang telah ditentukan sebelumnya untuk menghasilkan alternatif *relayout* tata ruang yang lebih optimal.

Proses algoritma pada Dynamo dilakukan melalui beberapa tahapan, yaitu:



Gambar 4. 4 Proses Algoritma pada Dynamo

Tahap pengembangan algoritma dilakukan menggunakan Dynamo sebagai alat *parametric design* untuk menghasilkan alternatif *relayout* tata ruang yang lebih optimal. Algoritma ini dikembangkan dengan menghubungkan model bangunan eksisting yang telah dibuat pada Autodesk Revit dengan parameter desain yang telah ditentukan sebelumnya, seperti kebutuhan luas ruang, hubungan antar ruang, pola sirkulasi, dan batas bangunan eksisting.

Penggunaan Dynamo dalam penelitian ini bertujuan untuk mengotomatisasi proses eksplorasi tata ruang sehingga perubahan pada parameter tertentu dapat secara langsung memengaruhi

konfigurasi layout yang dihasilkan. Melalui pendekatan ini, proses relayout tidak dilakukan secara manual, tetapi melalui sistem algoritma yang mampu menghasilkan alternatif penataan ruang berdasarkan kebutuhan operasional terminal.

Pengembangan algoritma dilakukan melalui beberapa tahapan utama, yaitu proses input data ruang, pembacaan batas ruang, distribusi kebutuhan ruang, pengaturan hubungan antar ruang, serta konfigurasi pola sirkulasi penumpang. Setiap tahapan tersebut saling terintegrasi dalam menghasilkan layout yang lebih efisien dan sesuai dengan kebutuhan terminal penumpang.

Pada tahap awal, algoritma membaca data ruang dari model Revit berupa luasan eksisting, nama ruang, serta batas bangunan yang menjadi area kerja relayout. Selanjutnya sistem melakukan proses penyesuaian kebutuhan luas ruang berdasarkan parameter kapasitas penumpang dan standar kebutuhan ruang.

Setelah kebutuhan luas ruang ditentukan, algoritma kemudian mengatur hubungan antar ruang berdasarkan urutan aktivitas penumpang seperti area *check-in*, pemeriksaan keamanan, ruang tunggu, hingga akses menuju kapal. Tahap akhir algoritma berfokus pada pengaturan pola sirkulasi penumpang untuk mengurangi potensi *bypassing circulation* dan menciptakan alur pergerakan yang lebih terarah.

Hasil dari pengembangan algoritma ini berupa beberapa alternatif konfigurasi tata ruang yang kemudian dievaluasi untuk menentukan layout yang paling optimal berdasarkan efisiensi ruang, kapasitas pengguna, dan kenyamanan sirkulasi penumpang.



Gambar 4. 5 Ringkasan Alur Optimasi Layout Terminal Penumpang

4.4.1 Input Node

Input node merupakan tahap awal dalam pengembangan algoritma Dynamo yang berfungsi untuk memasukkan data dan parameter desain ke dalam sistem. Pada tahap ini, data yang digunakan berasal dari model eksisting pada Revit dan hasil analisis kebutuhan ruang terminal. Data tersebut kemudian diolah sebagai dasar dalam proses space allocation, pengaturan hubungan antar ruang, serta konfigurasi sirkulasi penumpang.

Dalam penelitian ini, input node digunakan untuk membaca elemen ruang pada model Revit,

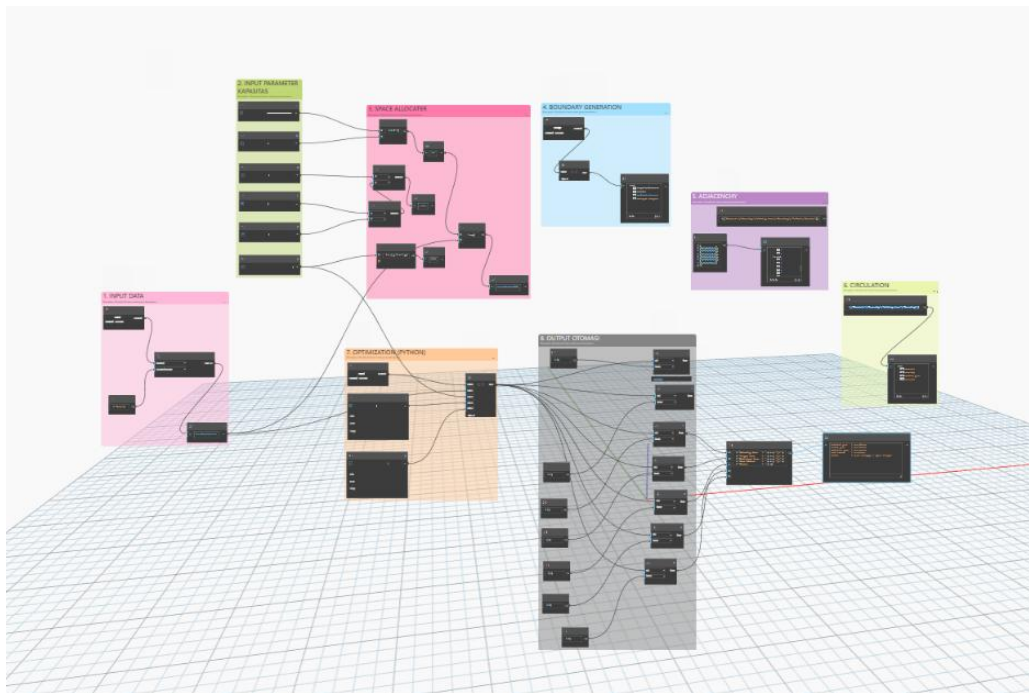
seperti nama ruang, luas ruang yang ada, dan batas ruang (batas ruangan). Selain itu, dimasukkan pula parameter kebutuhan ruang berdasarkan standar pelayanan terminal penumpang, seperti kebutuhan luas ruang tunggu keberangkatan, ruang keberangkatan, area check-in, area kedatangan, dan jalur sirkulasi penumpang. Parameter tersebut digunakan untuk menentukan kebutuhan luas ideal dan batas perubahan layout yang dapat dilakukan pada proses optimasi.

Tabel input node dapat disusun sebagai berikut:

Tabel 4. 3 *Input Node* pada Dynamo

No	Input Node	Data yang Diinput	Fungsi
1	Rooms from Revit	Nama dan data ruang eksisting	Membaca ruang yang ada pada model Revit
2	Area Parameter	Luas ruang eksisting	Mengetahui luas awal tiap ruang
3	Required Area	Luas ruang berdasarkan standar	Menentukan kebutuhan luas ideal
4	Building Boundary	Batas bangunan eksisting	Membatasi area relayout
5	Room Function	Fungsi ruang	Mengelompokkan ruang berdasarkan aktivitas
6	Circulation Input	Alur penumpang	Menentukan jalur pergerakan pengguna

Alur *input node* pada dynamo digambarkan secara berikut:



Gambar 4. 6 Node Graph Dynamo untuk Proses Optimasi *Layout* Ruang Terminal Penumpang

Secara teknis, proses input data pada Dynamo dimulai dengan pemilihan elemen ruang dari model Revit menggunakan node *Select Model Element*. Node tersebut digunakan untuk memilih ruang yang akan dijelaskan dan dioptimasi, khususnya ruang dengan tingkat aktivitas penumpang tinggi, seperti ruang tunggu keberangkatan. Elemen ruang yang telah dipilih kemudian dihubungkan dengan node *Element.GetParameterValueByName* untuk membaca parameter luas ruang (Area) sebagai nilai luas yang ada.

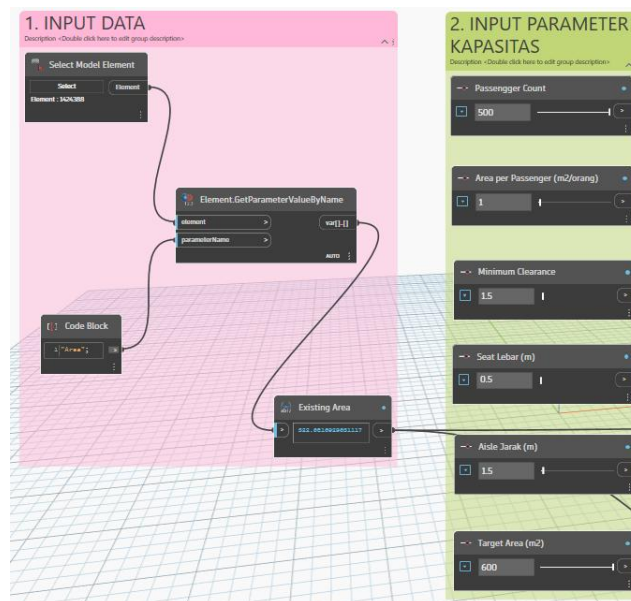
Nilai luas yang ada tersebut selanjutnya digunakan sebagai dasar dalam proses perhitungan kebutuhan ruang. Pada tahap ini, Dynamo membaca kondisi aktual ruang pada model Revit sehingga sistem dapat mengetahui apakah luas ruang masih memenuhi kebutuhan kapasitas penumpang atau memerlukan penyesuaian tata letak.

Selain membaca data ruang yang ada, penelitian ini juga memasukkan parameter desain melalui node *Number Slider* agar nilai parameter dapat diubah secara dinamis sesuai optimasi skenario yang diinginkan. Parameter yang dimasukkan meliputi jumlah penumpang (*passenger count*), kebutuhan luas per penumpang (*area per penumpang*), target luas ruang (*target area*), minimum clearance, lebar kursi, serta jarak sirkulasi. Penggunaan *Number Slider* memungkinkan beberapa simulasi kondisi kapasitas ruang tanpa perlu mengubah script utama secara manual.

Parameter kapasitas tersebut kemudian digunakan oleh Dynamo untuk menghitung kebutuhan

luas ruang secara otomatis. Misalnya, jumlah penumpang dikalikan dengan kebutuhan luas per orang untuk menghasilkan estimasi kebutuhan area ruang tunggu. Hasil perhitungan tersebut dibandingkan dengan luas yang ada sehingga diperoleh selisih nilai kebutuhan ruang berupa defisit atau surplus area.

Seluruh input data kemudian diintegrasikan dalam node graph Dynamo dan menjadi dasar pada proses alokasi ruang serta optimasi tata letak pada tahap berikutnya. Dengan pendekatan ini, Dynamo tidak hanya berfungsi sebagai alat visualisasi geometri, tetapi juga sebagai sistem parametrik yang mampu membaca kondisi yang ada, menghitung kebutuhan ruang, dan menghasilkan kuantitas dasar dalam proses relay layout terminal penumpang.



Gambar 4. 7 Input Data dan Parameter Kapasitas pada Dynamo yang Digunakan sebagai Dasar Perhitungan Kebutuhan Ruang

4.4.2 Room Boundary Generation

Tahap room boundary generation merupakan proses identifikasi batas fisik setiap ruang pada model bangunan eksisting yang telah dibuat pada Autodesk Revit. Pada tahap ini, Dynamo membaca elemen pembatas ruang seperti dinding, kolom, bukaan, serta garis batas ruang (room separator) untuk menentukan area yang dapat digunakan dalam proses relay layout.

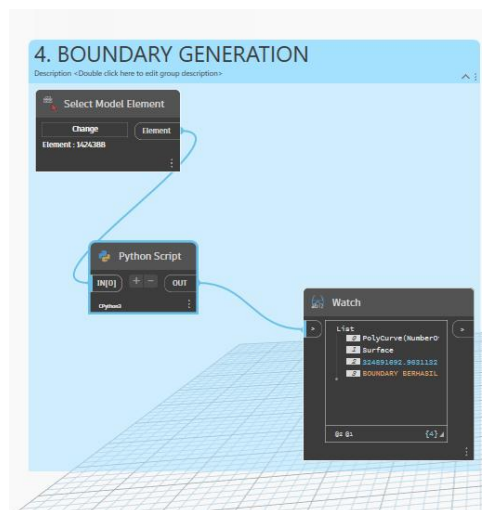
Proses ini bertujuan untuk mengetahui dimensi dan bentuk masing-masing ruang eksisting sehingga sistem dapat mengenali batas perubahan layout yang masih berada dalam area bangunan terminal. Selain itu, tahap ini juga digunakan untuk mengidentifikasi ruang yang tetap dipertahankan serta ruang yang berpotensi mengalami perubahan konfigurasi.

Dalam penelitian ini, room boundary generation difokuskan pada area utama terminal yang berkaitan langsung dengan aktivitas penumpang, seperti departure hall, departure waiting room, area check-in, area kedatangan, serta jalur akses menuju kapal. Batas ruang tersebut kemudian diterjemahkan ke dalam bentuk geometri digital yang akan digunakan pada tahap distribusi ulang ruang.

Adapun proses pada tahap ini meliputi:

1. Membaca elemen dinding sebagai batas ruang
2. Membaca luasan geometri tiap ruang
3. Mengidentifikasi posisi ruang eksisting
4. Menentukan area yang dapat dioptimalkan
5. Menetapkan batas bangunan yang tetap dipertahankan

Secara sederhana, alur proses room boundary generation dapat digambarkan sebagai berikut:



Gambar 4. 8 Input Node Bagian Boundary Room/Generation

Pada tahap ini, proses pembacaan boundary ruang dilakukan secara teknis menggunakan kombinasi node visual Dynamo dan Python Script. Proses dimulai dengan node Select Model Element untuk memilih elemen ruang pada model Revit yang akan dianalisis. Ruang yang dipilih kemudian diteruskan ke Python Script agar proses pembacaan boundary dapat dilakukan secara lebih stabil dan fleksibel.

Penggunaan Python Script pada penelitian ini dilakukan karena pembacaan boundary ruang menggunakan node visual standar Dynamo sering mengalami kendala apabila bentuk ruang memiliki banyak segmen garis, geometri tidak beraturan, atau boundary yang terbentuk dari beberapa elemen pembatas ruang. Dengan menggunakan Python Script, proses pembacaan

geometri ruang menjadi lebih stabil dan mampu membaca boundary ruang secara lebih detail.

```

Python
import clr

clr.AddReference("RevitServices")
from RevitServices.Persistence import DocumentManager

clr.AddReference("RevitAPI")
from Autodesk.Revit.DB import *

clr.AddReference("ProtoGeometry")
from Autodesk.DesignScript.Geometry import PolyCurve, Surface

clr.AddReference("RevitNodes")
import Revit
clr.ImportExtensions(Revit.GeometryConversion)

doc = DocumentManager.Instance.CurrentDBDocument

room = UnwrapElement(IN[0])

status = "OK"
polycurve = None
surface = None
area = 0

try:
    opt = SpatialElementBoundaryOptions()
    loops = room.GetBoundarySegments(opt)

    if loops is None:
        status = "ERROR: Room tidak punya boundary. Pastikan room tertutup."
    else:
        # pilih loop boundary dengan segmen terbanyak
        main_loop = None
        max_count = 0

        for loop in loops:
            if len(loop) > max_count:
                main_loop = loop
                max_count = len(loop)

        curves = []

        for seg in main_loop:
            revit_curve = seg.GetCurve()
            ds_curve = revit_curve.ToPrototype()
            curves.append(ds_curve)

        polycurve = PolyCurve.ByJoinedCurves(curves)
        surface = Surface.ByPatch(polycurve)
        area = surface.Area

        status = "BOUNDARY BERHASIL"
except Exception as e:
    status = "ERROR: " + str(e)

OUT = [polycurve, surface, area, status]

```

Gambar 4. 9 Script Phyton Boundary Room

Di dalam Python Script, elemen ruang dari Revit diproses menggunakan fungsi *GetBoundarySegments()* untuk membaca seluruh garis batas ruang yang terbentuk dari elemen *room bounding*, seperti dinding, kolom, dan *room separator*. Fungsi tersebut menghasilkan kumpulan segmen boundary yang kemudian diproses untuk menentukan loop boundary utama dari ruang yang dipilih.

Setelah boundary utama diperoleh, setiap segmen garis diubah menjadi kurva geometri Dynamo (*curve geometry*). Kurva-kurva tersebut kemudian digabungkan menggunakan proses join curve sehingga membentuk *PolyCurve* tertutup. Selanjutnya, *PolyCurve* tersebut dikonversi menjadi *Surface* yang merepresentasikan bentuk geometris ruang eksisting pada Dynamo.

Hasil akhir dari tahap ini berupa geometri digital ruang dalam bentuk *PolyCurve* dan *Surface*,

lengkap dengan informasi luas ruang eksisting yang terbaca secara otomatis dari model Revit. Geometri tersebut kemudian digunakan sebagai dasar pada tahap *space allocation* dan optimasi boundary ruang pada proses selanjutnya.

Melalui pendekatan ini, Dynamo tidak hanya membaca data numerik ruang, tetapi juga mampu merepresentasikan bentuk spasial ruang secara parametrik sehingga perubahan layout dapat dianalisis dan dioptimalkan berdasarkan kondisi eksisting bangunan terminal.

4.4.3 Space Allocation

Tahap *space allocation* merupakan proses distribusi ulang luasan ruang berdasarkan parameter kebutuhan ruang yang telah ditentukan sebelumnya. Pada tahap ini, Dynamo mengolah data luas ruang eksisting, kebutuhan luas ideal, serta batas bangunan untuk menghasilkan pembagian ruang yang lebih proporsional.

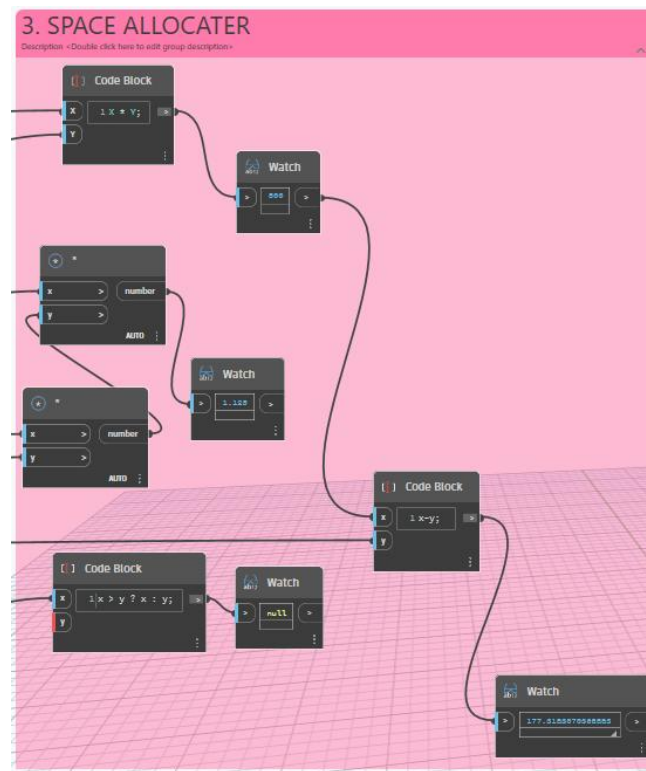
Proses ini dilakukan karena hasil analisis kondisi eksisting menunjukkan adanya ketidakseimbangan distribusi ruang, di mana beberapa ruang memiliki luasan yang relatif besar namun kurang optimal dalam mendukung aktivitas utama terminal. Sebaliknya, ruang dengan intensitas penggunaan tinggi seperti *departure waiting room* justru memiliki luas yang lebih kecil dibandingkan kebutuhan standar.

Pada tahap *space allocation*, sistem melakukan penyesuaian luas ruang dengan cara mengalokasikan kembali area bangunan agar sesuai dengan kebutuhan operasional terminal. Dalam penelitian ini, prioritas alokasi diberikan pada ruang-ruang utama yang memiliki aktivitas tinggi, seperti ruang tunggu keberangkatan, area keberangkatan, area kedatangan, dan area pemeriksaan penumpang. Proses yang dilakukan pada tahap ini meliputi:

1. membaca luas ruang eksisting
2. membandingkan dengan kebutuhan luas ideal
3. mengurangi ruang yang berlebih
4. menambah ruang yang kurang
5. mendistribusikan ulang ruang dalam batas bangunan eksisting

Sebagai contoh, luas *departure waiting room* pada kondisi eksisting sebesar $\pm 240 \text{ m}^2$ kemudian dialokasikan ulang menjadi $\pm 600 \text{ m}^2$ agar mampu memenuhi kebutuhan kapasitas penumpang. Sementara itu, luasan *departure hall* yang sebelumnya relatif besar disesuaikan kembali agar distribusi ruang menjadi lebih efisien.

Alur proses space allocation dapat digambarkan sebagai berikut:



Gambar 4. 10 Input Node Space Allocation

Secara teknis, proses *space allocation* pada Dynamo dilakukan dengan membaca nilai luas ruang eksisting dari model Revit menggunakan parameter *Area*. Nilai tersebut kemudian dibandingkan dengan kebutuhan luas ruang yang dihitung berdasarkan parameter kapasitas penumpang (*passenger count*) dan kebutuhan luas per orang (*area per passenger*).

Pada tahap ini, parameter desain dimasukkan melalui *Number Slider* sehingga perubahan nilai kapasitas dan kebutuhan luas dapat dilakukan secara dinamis. Dynamo kemudian melakukan proses perhitungan otomatis untuk menentukan kebutuhan luas ruang ideal berdasarkan rumus kapasitas ruang.

Hasil perhitungan kebutuhan luas tersebut dibandingkan dengan kondisi eksisting untuk mengetahui selisih kebutuhan ruang. Apabila luas ruang eksisting lebih kecil dibandingkan kebutuhan luas ideal, maka sistem mengidentifikasi adanya defisit ruang. Sebaliknya, apabila luas eksisting lebih besar dari kebutuhan, maka ruang tersebut dianggap mengalami surplus area yang berpotensi dialokasikan kembali pada proses relayout.

Berdasarkan hasil perbandingan tersebut, Dynamo kemudian menentukan target luas ruang (*target area*) yang akan digunakan sebagai acuan pada tahap optimasi boundary dan relayout ruang. Pada penelitian ini, proses distribusi ulang ruang tetap dibatasi oleh boundary bangunan

eksisting sehingga perubahan layout yang dihasilkan tetap realistis dan sesuai dengan kondisi bangunan terminal.

Hasil dari tahap *space allocation* berupa konfigurasi awal pembagian kebutuhan ruang baru yang kemudian digunakan pada tahap selanjutnya, yaitu *adjacency configuration* dan pengaturan sirkulasi penumpang. Dengan pendekatan parametrik ini, Dynamo mampu menghasilkan proses distribusi ruang yang lebih terukur, fleksibel, dan sesuai dengan kebutuhan operasional terminal penumpang.

4.4.4 *Adjacency Configuration*

Tahap *adjacency configuration* merupakan proses pengaturan hubungan kedekatan antar ruang berdasarkan urutan aktivitas pengguna di dalam terminal penumpang. Pada tahap ini, Dynamo mengatur posisi ruang agar hubungan antar fungsi menjadi lebih terorganisir dan mendukung operasional terminal secara lebih efisien.

Pengaturan kedekatan ruang dilakukan berdasarkan pola aktivitas penumpang keberangkatan dan kedatangan. Ruang-ruang yang memiliki hubungan aktivitas langsung ditempatkan berdekatan agar mempermudah pergerakan pengguna dan mengurangi perpindahan yang tidak efisien.

Dalam penelitian ini, hubungan ruang yang menjadi prioritas meliputi:

area masuk terminal → area check-in

area check-in → security check

security check → ruang tunggu keberangkatan

ruang tunggu → akses boarding menuju kapal

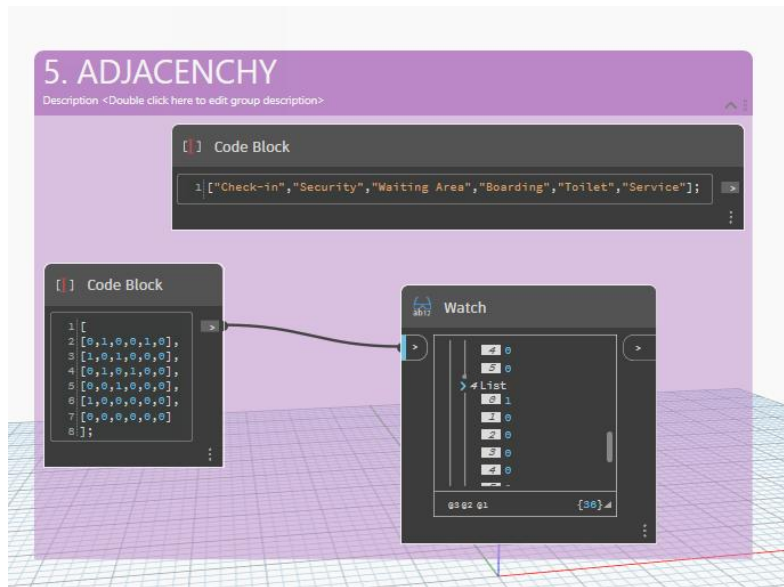
area kedatangan → area keluar terminal

Selain itu, posisi ruang seperti information center dan security center juga diatur ulang agar dapat berfungsi lebih optimal sebagai area kontrol pergerakan pengguna dan mengurangi potensi bypassing circulation.

Tahapan pada proses ini meliputi:

1. identifikasi hubungan aktivitas antar ruang
2. menentukan tingkat kedekatan ruang
3. mengatur ulang posisi ruang
4. menghasilkan konfigurasi hubungan ruang baru

Alur proses *adjacency configuration* dapat digambarkan sebagai berikut:



Gambar 4. 11 *Input Node Dynamo Adjacency*

Secara teknis, proses *adjacency configuration* pada Dynamo dilakukan menggunakan Code Block yang berisi daftar nama ruang serta hubungan antar ruang dalam bentuk *adjacency matrix*. Daftar ruang digunakan untuk mendefinisikan fungsi-fungsi utama pada terminal penumpang, seperti *check-in*, *security*, *waiting area*, dan *boarding*.

Sementara itu, *adjacency matrix* digunakan untuk menentukan tingkat hubungan antar ruang. Pada penelitian ini, nilai hubungan ruang direpresentasikan menggunakan parameter biner, yaitu nilai “1” untuk ruang yang harus berdekatan dan nilai “0” untuk ruang yang tidak memerlukan hubungan langsung. Matriks tersebut kemudian digunakan oleh Dynamo sebagai dasar dalam membaca pola hubungan aktivitas penumpang di dalam terminal.

Melalui pendekatan parametrik ini, sistem dapat mengenali ruang mana yang harus memiliki koneksi langsung berdasarkan urutan aktivitas pengguna. Sebagai contoh, ruang *check-in* harus memiliki hubungan langsung dengan area *security check*, sedangkan ruang tunggu keberangkatan harus memiliki hubungan langsung dengan area *boarding*. Sebaliknya, ruang yang tidak memiliki hubungan aktivitas langsung tidak diprioritaskan untuk ditempatkan berdekatan.

Hasil dari tahap ini berupa konfigurasi hubungan antar ruang yang lebih logis dan mendukung alur aktivitas pengguna terminal. Konfigurasi tersebut kemudian digunakan sebagai dasar pada tahap pengaturan sirkulasi (*circulation configuration*) dan optimasi layout ruang pada proses selanjutnya. Dengan pendekatan ini, relayout tidak hanya mempertimbangkan kebutuhan luas

ruang, tetapi juga memperhatikan hubungan fungsional antar ruang agar sistem pelayanan terminal menjadi lebih efektif dan terarah.

4.4.5 *Circulation Configuration*

Tahap *circulation configuration* merupakan proses pengaturan jalur pergerakan pengguna dalam bangunan terminal agar alur sirkulasi menjadi lebih jelas, efisien, dan terkontrol. Pada tahap ini, Dynamo mengoptimalkan pola pergerakan penumpang berdasarkan konfigurasi ruang yang telah dihasilkan sebelumnya.

Fokus utama pada tahap ini adalah mengurangi potensi terjadinya *bypassing circulation*, yaitu kondisi ketika penumpang dapat langsung menuju ruang tunggu tanpa melalui alur pelayanan yang seharusnya seperti area *check-in* dan pemeriksaan keamanan.

Melalui tahap ini, jalur sirkulasi penumpang diatur agar mengikuti urutan aktivitas yang lebih terarah, yaitu:

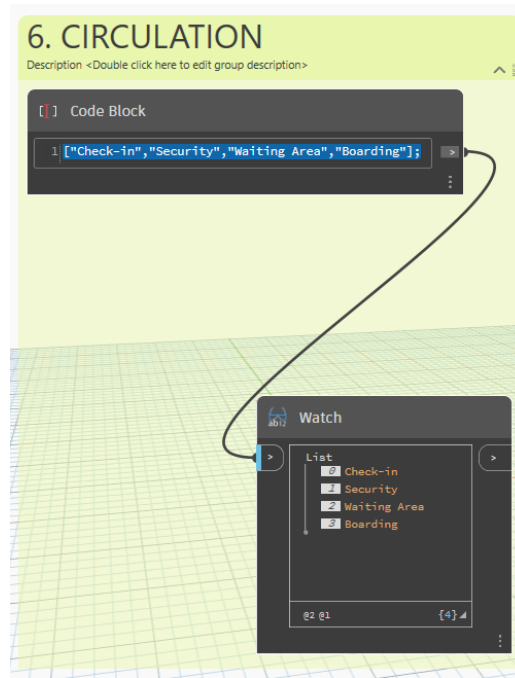
1. Masuk terminal
2. Area *check-in*
3. Pemeriksaan keamanan
4. Ruang tunggu keberangkatan
5. *Boarding* menuju kapal

Selain itu, jalur kedatangan penumpang juga diatur agar tidak bertabrakan dengan jalur keberangkatan sehingga konflik sirkulasi dapat diminimalkan.

Tahapan pada proses ini meliputi:

1. Identifikasi jalur sirkulasi eksisting
2. Analisis potensi konflik pergerakan
3. Pengaturan jalur sirkulasi baru
4. Evaluasi efisiensi pergerakan pengguna

Alur proses *circulation configuration* dapat digambarkan sebagai berikut:



Gambar 4. 12 *Input Node Circulation Configuration*

Secara teknis, proses *circulation configuration* pada Dynamo dilakukan dengan menghubungkan hubungan antar ruang yang telah disusun pada tahap *adjacency configuration* ke dalam urutan alur aktivitas pengguna. Pada tahap ini, Dynamo membaca posisi ruang, hubungan kedekatan antar ruang, serta titik akses masuk dan keluar untuk membentuk pola sirkulasi terminal.

Konfigurasi sirkulasi disusun berdasarkan urutan pelayanan penumpang sehingga ruang-ruang dengan hubungan aktivitas langsung dihubungkan dalam satu jalur pergerakan yang berkesinambungan. Jalur tersebut kemudian digunakan sebagai dasar evaluasi efisiensi sirkulasi pada layout terminal.

Dalam penelitian ini, proses konfigurasi sirkulasi juga mempertimbangkan posisi ruang kontrol seperti *security center* dan *information center*. Penempatan kedua ruang tersebut dioptimalkan agar mampu mengontrol arah pergerakan pengguna dan mengurangi kemungkinan penumpang melewati jalur pelayanan tanpa mengikuti prosedur yang telah ditentukan.

Selain itu, pengaturan jalur sirkulasi dilakukan dengan mempertimbangkan pemisahan jalur keberangkatan dan kedatangan penumpang. Pemisahan tersebut bertujuan untuk mengurangi potensi *cross circulation* atau perpotongan arus pengguna yang dapat menyebabkan kepadatan dan konflik pergerakan pada area terminal.

Hasil dari tahap ini berupa pola sirkulasi baru yang lebih terarah, memiliki urutan aktivitas

yang lebih jelas, serta mendukung kenyamanan dan efisiensi operasional Terminal Penumpang Pelabuhan Donggala. Pola sirkulasi tersebut kemudian menjadi dasar dalam proses optimasi layout dan evaluasi hasil relayout pada tahap selanjutnya.

4.4.6 *Optimization dengan Python Script*

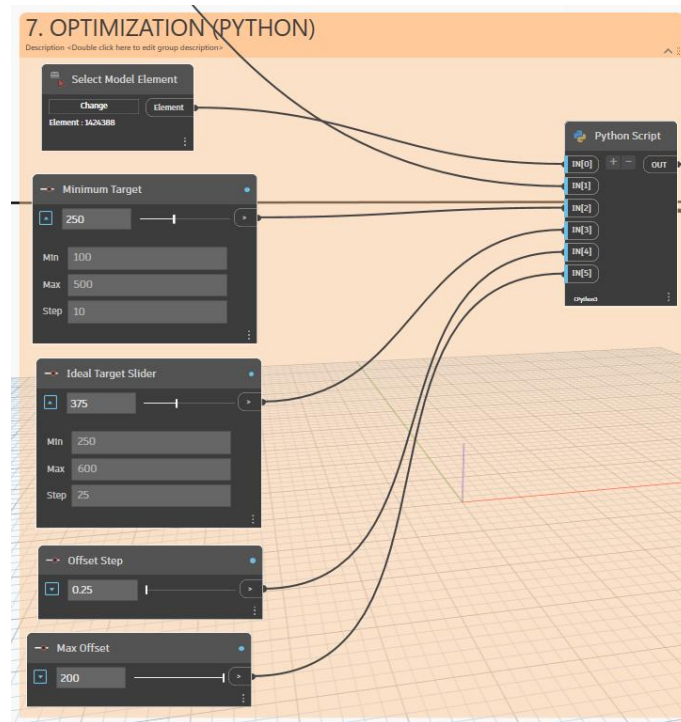
Tahap optimization dengan Python Script merupakan tahap utama dalam proses optimasi relayout tata ruang terminal penumpang. Pada tahap ini, Dynamo digunakan sebagai media penghubung antara data ruang dari model Revit dan proses komputasi menggunakan Python. Berbeda dengan tahap sebelumnya yang hanya membaca data dan membentuk hubungan ruang, tahap ini berfungsi untuk mengevaluasi apakah ruang yang dipilih telah memenuhi kebutuhan minimum dan kebutuhan ideal berdasarkan skenario kapasitas penumpang.

Dalam penelitian ini, proses optimasi difokuskan pada ruang utama terminal yang memiliki peran penting terhadap aktivitas penumpang, khususnya departure waiting room. Ruang tersebut dipilih karena memiliki intensitas penggunaan tinggi dan berpotensi mengalami penumpukan penumpang pada kondisi peak hour. Oleh karena itu, proses optimasi tidak hanya diarahkan untuk mengetahui luas eksisting ruang, tetapi juga untuk membandingkan luas tersebut terhadap dua tingkat kebutuhan, yaitu minimum target dan ideal target.

Secara teknis, proses optimasi dimulai dengan pemilihan elemen ruang menggunakan node Select Model Element. Elemen ruang yang dipilih kemudian dihubungkan ke Python Script sebagai input utama. Selain elemen ruang, script juga menerima beberapa parameter optimasi, yaitu existing area, minimum target, ideal target, offset step, dan max offset. Parameter existing area merupakan luas ruang eksisting yang terbaca dari model Revit. Parameter minimum target digunakan sebagai batas kebutuhan minimum agar ruang masih dapat berfungsi, sedangkan ideal target digunakan sebagai target kebutuhan ruang yang lebih nyaman pada kondisi peak hour. Parameter offset step digunakan untuk mengatur interval perubahan boundary pada setiap iterasi, sedangkan max offset digunakan sebagai batas maksimum perubahan boundary yang diperbolehkan dalam proses optimasi.

Pada konfigurasi terbaru, nilai minimum target dan ideal target dimasukkan melalui node Number Slider agar dapat diubah secara dinamis sesuai skenario perancangan. Dalam penelitian ini, minimum target digunakan untuk mengevaluasi ketercapaian kebutuhan minimum ruang, sedangkan ideal target digunakan untuk mengetahui apakah ruang tersebut

masih membutuhkan relay layout tambahan untuk mencapai kondisi yang lebih optimal. Pendekatan ini digunakan karena kapasitas penumpang pada terminal tidak seluruhnya ditempatkan dalam satu ruang secara bersamaan, melainkan terbagi berdasarkan zona aktivitas, seperti area check-in, security check, waiting hall, dan akses boarding.



Gambar 4. 13 Node Optimization dengan Python Script pada Dynamo

Di dalam Python Script, elemen ruang dari Revit dibaca menggunakan Revit API. Script kemudian mengambil boundary ruang melalui fungsi `GetBoundarySegments()`. Boundary tersebut dikonversi menjadi geometri Dynamo berupa `PolyCurve` dan `Surface` agar dapat digunakan dalam proses simulasi perubahan ruang. Setelah boundary dasar terbentuk, script melakukan proses iterasi offset secara bertahap berdasarkan nilai offset step yang telah ditentukan.

Pada setiap iterasi, script membuat boundary baru melalui proses offset, kemudian menghitung luas area hasil perubahan tersebut. Luas hasil iterasi dibandingkan dengan minimum target dan ideal target. Apabila luas hasil optimasi sudah mencapai minimum target, maka sistem memberikan status bahwa kebutuhan minimum telah tercapai. Namun, apabila luas tersebut belum mencapai ideal target, maka sistem tetap memberikan rekomendasi bahwa relay layout area sekitar masih diperlukan.

Perbedaan penting pada script terbaru adalah sistem tidak hanya menghasilkan satu status

“optimal” atau “belum tercapai”, tetapi membagi hasil evaluasi menjadi dua tingkat, yaitu Status Minimum dan Status Ideal. Hal ini membuat hasil optimasi menjadi lebih akurat secara akademik, karena ruang dapat dinilai memenuhi kebutuhan minimum, namun belum tentu memenuhi kebutuhan ideal pada kondisi puncak. Dengan demikian, hasil simulasi tidak dipaksakan menjadi optimal secara keseluruhan, tetapi dibaca berdasarkan tingkat ketercapaian kebutuhan ruang.

Output utama dari script terbaru berupa teks ringkasan hasil optimasi yang ditampilkan melalui node Watch. Ringkasan tersebut memuat nilai Existing Area, Minimum Target, Ideal Target, Optimized Area, Best Offset, Status Minimum, Status Ideal, dan Rekomendasi. Output ini menjadi dasar dalam menentukan apakah ruang yang dianalisis sudah memenuhi kebutuhan minimum atau masih memerlukan penyesuaian desain pada tahap relayout.

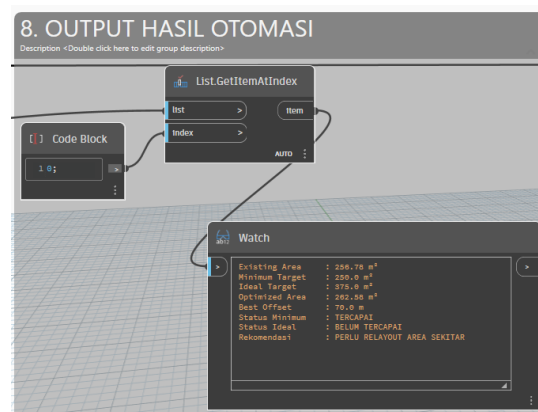
4.4.7 Output Optimasi

Tahap output optimasi merupakan tahap akhir dari proses node graph Dynamo yang berfungsi untuk menampilkan hasil evaluasi dari Python Script. Pada tahap ini, hasil komputasi optimasi ditampilkan melalui node Watch dalam bentuk ringkasan data numerik dan status ketercapaian target ruang.

Output yang dihasilkan meliputi luas ruang eksisting (existing area), target minimum (minimum target), target ideal (ideal target), luas hasil optimasi (optimized area), nilai best offset, status ketercapaian minimum, status ketercapaian ideal, serta rekomendasi layout. Data tersebut digunakan untuk mengetahui apakah ruang yang dianalisis telah memenuhi kebutuhan minimum dan apakah masih memerlukan penyesuaian untuk mencapai kondisi ideal.

Pada penelitian ini, output optimasi tidak hanya digunakan untuk menunjukkan hasil perhitungan luas ruang, tetapi juga menjadi dasar dalam menentukan kebutuhan relay layout area sekitar. Apabila status minimum telah tercapai tetapi status ideal belum tercapai, maka ruang tersebut dinilai masih dapat berfungsi secara minimum, namun belum optimal untuk kondisi peak hour. Oleh karena itu, sistem memberikan rekomendasi berupa perlunya relay layout pada area sekitar ruang yang dianalisis.

Dengan demikian, tahap output optimasi berfungsi sebagai jembatan antara proses komputasi pada Dynamo dan pengambilan keputusan desain. Hasil yang ditampilkan pada node Watch digunakan sebagai dasar untuk menyusun usulan relay layout tata ruang yang lebih sesuai dengan kebutuhan kapasitas, hubungan ruang, dan sirkulasi penumpang.



Gambar 4. 15 Output Hasil Optimasi Pada Dynamo