

LAPORAN BULANAN

PEKERJAAN PEMBANGUNAN, PENGEMBANGAN
DAN PENGELOLAAN APLIKASI JAKARTA SATU



GIS Development

Ari Matiur S. T.

Pusat Data dan Informasi
Dinas Cipta Karya, Tata Ruang, dan Pertanahan
Provinsi DKI Jakarta

Gedung Dinas Teknis Jatibaru Lt. 4
Jl. Taman Jati Baru
Cideng, Gambir, Kota Jakarta Pusat
021-3503035/uptd.cktrp@jakarta.go.id

Juli 2021

DAFTAR ISI

DAFTAR ISI.....
DAFTAR GAMBAR.....	II
DAFTAR TABEL.....	IV
BAB I	
PENDAHULUAN.....	1
1.1 LATAR BELAKANG.....	1
1.2 RUMUSAN PEKERJAAN.....	1
BAB II	
TAHAPAN PELAKSANAAN PEKERJAAN.....	3
2.1 PEMBUATAN TOOLS PROCESSING LAS TO MULTIPATCH.....	3
BAB III	
HASIL PELAKSANAAN PEKERJAAN.....	20
3.1 PEMBUATAN TOOLS PROCESSING LAS TO MULTIPATCH.....	20
BAB IV	
KESIMPULAN DAN REKOMENDASI.....	24
4.1 PEMBUATAN TOOLS PROCESSING LAS TO MULTIPATCH.....	24
LAMPIRAN.....	25

DAFTAR GAMBAR

Gambar 2.1 Diagram Alir Tools.....	4
Gambar 2.2 Pembuatan project baru ArcGIS Pro.....	4
Gambar 2.3 Folder folder source tools.....	4
Gambar 2.4 Folder project ArcGIS Pro sebagai folder project Pycharm.....	5
Gambar 2.5 Pembuatan python file.....	6
Gambar 2.6 Import modul dan membuat function.....	6
Gambar 2.7 Blok script pembuatan LAS dataset.....	6
Gambar 2.8 Klasifikasi Ground LAS Dataset.....	7
Gambar 2.9 Klasifikasi Bangunan LAS Dataset.....	7
Gambar 2.10 Membuat Raster DTM dan Mask Raster DTM.....	8
Gambar 2.11 Membuat Raster DSM dan Mask Raster DSM.....	8
Gambar 2.12 Membuat Raster nDSM dan Mask Raster nDSM.....	9
Gambar 2.13 Membuat Clip Footprint Bangunan dengan AOI.....	9
Gambar 2.14 Membuat Clip Footprint Bangunan dengan AOI.....	9
Gambar 2.15 Script LAS_to_DEM.py.....	10
Gambar 2.16 Module yang digunakan file LAS_to_DEM_UI.py.....	11
Gambar 2.17 Variable default dan konfigurasi environment.....	11
Gambar 2.18 Non default variable.....	11
Gambar 2.19 Isi script file LAS_to_DEM_UI.py.....	12
Gambar 2.20 Membuat script baru pada toolbox.....	12
Gambar 2.21 Konfigurasi Tab General New Script.....	13
Gambar 2.22 Konfigurasi Tab Parameters New Script.....	13
Gambar 2.23 Tampilan Tool LAS to DEM.....	14
Gambar 2.24 File hasil download dari web esri.....	14
Gambar 2.25 File file yang di copy dari file download dari web esri.....	15
Gambar 2.26 Modifikasi file extract_roof_form.py.....	16
Gambar 2.27 Konfigurasi Tool DEM to Polygon 3D di Toolbox.....	16
Gambar 2.28 Tampilan Tool DEM to Polygon 3D.....	17
Gambar 2.29 File yang perlukan untuk tool Polygon 3D to Multipatch.....	17
Gambar 2.30 Modifikasi file fuse_building_parts.py.....	18
Gambar 2.31 Konfigurasi Tool Polygon 3D to Multipatch di Toolbox.....	18
Gambar 2.32 Tampilan Tool Polygon 3D to Multipatch.....	19
Gambar 2.33 Zip file semua resource tools.....	19
Gambar 3.1 File dan folder resource tool.....	20
Gambar 3.2 File file yang diperlukan untuk menjalankan tools.....	20

Gambar 3.3 Tool LAS to DEM yang sedang berjalan.....	21
Gambar 3.4 Output file hasil tool LAS to DEM.....	21
Gambar 3.5 Tool DEM to Polygon 3D yang sedang berjalan.....	22
Gambar 3.6 Output file hasil tool DEM to Polygon 3D.....	22
Gambar 3.7 Tool Polygon 3D to DEM yang sedang berjalan.....	23
Gambar 3.8 Output file hasil tool Polygon 3D to Multipatch.....	23

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Pusat Data dan Informasi Cipta Karya, Tata Ruang dan Pertanahan Provinsi DKI Jakarta merupakan salah satu Unit Pelaksana Teknis dari Satuan Kerja Perangkat Daerah (SKPD) Dinas Cipta Karya, Tata Ruang dan Pertanahan Provinsi DKI Jakarta.

Pusat Data dan Informasi Cipta Karya, Tata Ruang dan Pertanahan Provinsi DKI Jakarta sesuai dengan Peraturan Gubernur Provinsi DKI Jakarta Nomor 403 Tahun 2016 tentang Pembentukan Organisasi dan Tata Kerja Pusat Data dan Informasi Cipta Karya, Tata Ruang dan Pertanahan mempunyai tugas melaksanakan penghimpunan, pengolahan dan penyajian data dan informasi cipta karya, tata ruang dan pertanahan serta pengembangan dan pengelolaan sistem informasi cipta karya, tata ruang dan pertanahan.

Dalam pelaksanaan tugas di Pusat Data dan Informasi Cipta Karya, Tata Ruang dan Pertanahan dalam rangka mendukung tugas dan fungsi Dinas Cipta Karya, Tata Ruang dan Pertanahan, salah satunya adalah dengan mengembangkan Sistem Informasi Geospasial (SIG) untuk dapat membantu mencapai tujuan daerah dalam mewujudkan penataan ruang kota Jakarta yang terpadu dan berkelanjutan. Selanjutnya dalam pelaksanaan pengembangan Sistem Informasi Geospasial ini, Dinas Cipta Karya, Tata Ruang dan Pertanahan diamanatkan untuk membuat modeling sistem peta dan data untuk Program Jakarta Satu sesuai Instruksi Gubernur Provinsi DKI Jakarta Nomor 34 Tahun 2018 tentang Integrasi Sistem Peta dan Data Dalam Program Jakarta Satu.

Mendukung kebijakan Jakarta Satu “Satu Peta, Satu Data dan Satu Kebijakan”, maka diperlukan pengembangan sistem informasi yang akan dipergunakan sebagai media dalam mengimplementasikan penggunaan satu peta dasar bersama dan mengintegrasikan data di lingkungan Pemerintah Provinsi DKI Jakarta. Program Jakarta Satu dengan berbasis peta dasar tunggal yang harus digunakan oleh seluruh Perangkat Daerah/Unit Perangkat Daerah di Lingkungan Pemerintah Provinsi DKI Jakarta sesuai dengan Instruksi Gubernur Provinsi DKI Jakarta Nomor 107 Tahun 2018 tentang Pemanfaatan Peta Dasar Tunggal Provinsi DKI Jakarta. Implementasi kebijakan Satu Peta ini juga diamanatkan dalam Undang Undang Nomor 4 Tahun 2011 tentang Informasi Geospasial. Bahwa pengembangan sistem informasi melingkupi 10 aspek yang diprioritaskan dalam program Jakarta Satu yaitu, tata ruang, perizinan, aset, pajak, kependudukan, air tanah, lingkungan hidup, pendidikan, sosial, dan kesehatan.

1.2 RUMUSAN PEKERJAAN

Data LAS adalah data yang diakuisisi dengan survey LiDAR. Data ini berupa titik titik koordinat (point cloud) yang didapat dengan menggunakan sensor aktif (laser). Dta LAS bisa digunakan untuk berbagai macam kebutuhan salah satunya adalah pemodelan 3D. Pemodelan 3D yang bisa dihasilkan dari LAS berupa model 3D bangunan (multipatch) dan model 3D medan/terrain (DEM). Proses pengolahan dari data LAS

sampai menjadi model 3D membutuhkan proses panjang yang memerlukan banyak waktu, selain itu proses panjang ini juga sering terjadi kesalahan. Masalah ini bisa diatasi jika dibuat suatu sistem yang melakukan otomatisasi sehingga proses ini bisa lebih cepat dan terstandarisasi. Dengan menggunakan API Arcpy yang ada di ArcGIS Pro, otomatisasi ini bisa dilakukan dengan membuat tools berbasis python.

BAB II

TAHAPAN PELAKSAAN PEKERAAN

2.1 PEMBUATAN TOOLS PROCESSING LAS TO MULTIPATCH

Untuk bisa melakukan otomatisasi harus ditentukan terlebih dahulu input dan output dari sistem (tools) ini. Untuk pembuatan tools ini input yang diperlukan yaitu :

1. LAS Data
2. Footprint Bangunan (Digitasi Bangunan)
3. Area of Interest (AOI)

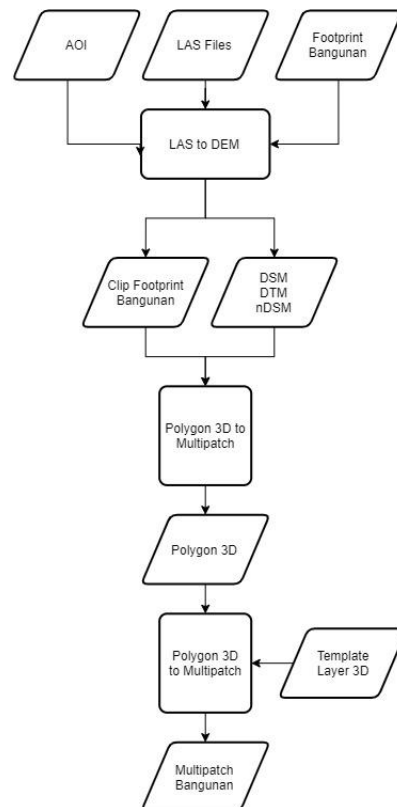
Sedangkan untuk output dari tools ini adalah:

1. Multipatch Bangunan
2. Digital Terrain Model (DTM)

Agar memudahkan proses pengembangan tools ini, tools ini akan dibagi menjadi 3 bagian. Tools akan terdiri dari :

1. LAS to DEM : Tool ini berguna untuk merubah data LAS menjadi DEM raster, DEM raster ini terbagi menjadi tiga raster DTM, DSM, nDSM.
2. DEM to Polygon 3D : Tool ini berguna untuk mengubah DEM dan Footprint Bangunan menjadi Footprint Bangunan dengan atribut 3D.
3. Polygon 3D to Multipatch : Tool ini berguna untuk mengubah Polygon 3D yang di simbology menggunakan template 3D menjadi Multipatch.

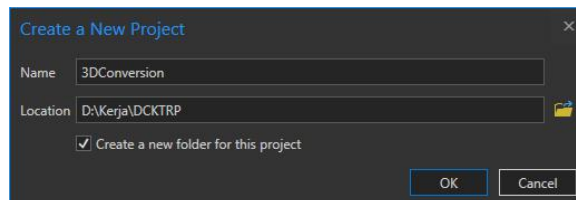
Berikut ini adalah diagram alir tools yang akan dibuat



Gambar 2.1 Diagram Alir Tools

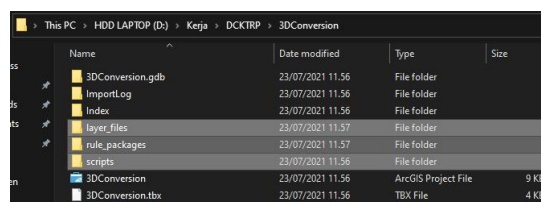
2.1.1. LAS to DEM

Tools ini akan membuat data DEM dan Clip Footprint Bangunan. Untuk membuat tools ini langkah pertama adalah membuat project ArcGIS Pro baru.



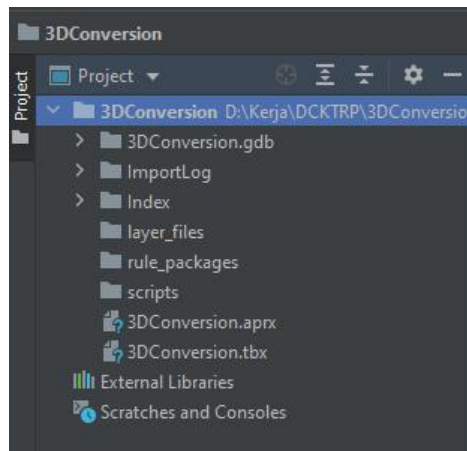
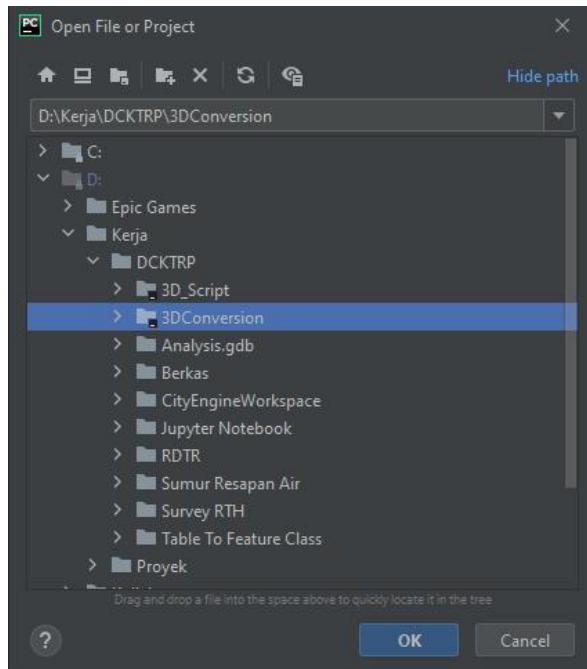
Gambar 2.2 Pembuatan prjct baru ArcGIS Pro

Setelah project dibuat bukan folder project dan buat folder folder yang diperlukan, folder yang harus dibuat adalah script, rule_packages dan layer_files. Folder folder ini akan berisikan file file yang diperlukan agar tools bisa digunakan. Folder Script adalah folder tempat file script python tools, folder rule_packages adalah folder tempat file rule simbologi layer (.rpk), folder layer_files adalah template untuk simbologi 3D polygon.



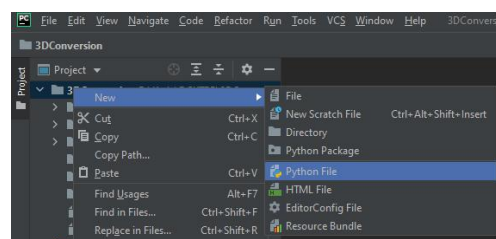
Gambar 2.3 Folder folder source tools

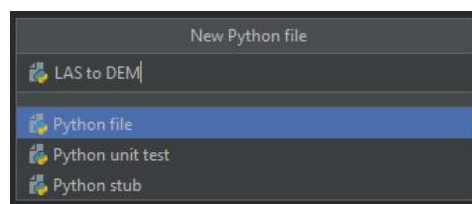
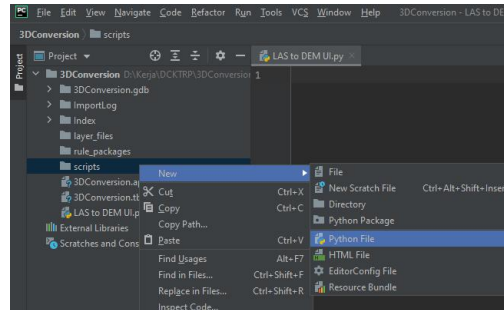
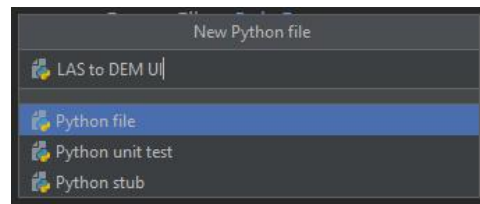
Tools yang akan dibuat menggunakan python oleh karena itu gunakan pycharm sebagai text editor untuk mulai membuat script. Buka pycharm kemudian buka folder project ArcGIS Pro, dengan membuka folder project ArcGIS Pro maka semua script yang akan dibuat akan berada di folder ini.



Gambar 2.4 Folder project ArcGIS Pro sebagai folder project Pycharm

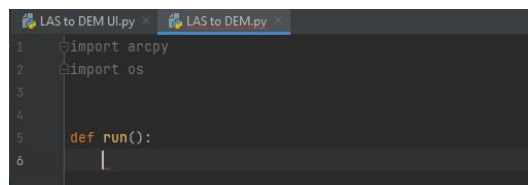
Setiap tools yang dibuat akan terdiri dari 2 komponen yaitu script tools dan script tampilan (user interface/ui) tools. Script tools akan disimpan di dalam folder script sedangkan script UI akan disimpan di project folder langsung. Pada folder project buat python file baru dengan nama LAS_to_DEM UI dan pada script folder buat python file baru dengan nama LAS_to_DEM.





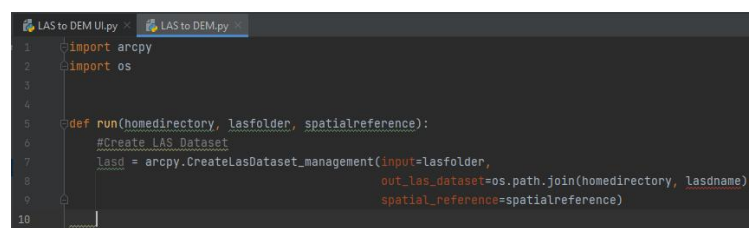
Gambar 2.5 Pembuatan python file

Buka LAS_to_DEM dan mulai menulis script tool. Script tool ini akan memiliki satu function yang mana function ini akan digunakan pada script UI tool. Pertama import modul yang diperlukan lalu buat function dengan nama run.



Gambar 2.6 Import modul dan membuat function

Function yang dibuat akan memiliki beberapa parameter dimana parameter ini akan menjadi variable untuk function parameter didalamnya, agar lebih mudah buat terlebih dahulu isi functionnya baru bisa ditentukan parameter parameter yang diperlukan. Isi function akan memiliki beberapa bagian sesuai tahapan pengolahan data. Tahap pertama adalah membuat LAS yang banyak menjadi satu file LAS Dataset. Isikan function dengan script seperti berikut.



Gambar 2.7 Blok script pembuatan LAS dataset

Pada blok script ini terdapat satu arcpy function yang digunakan yaitu CreateLasDataset_management, function ini akan membuat LAS menjadi LAS

dataset, function ini memerlukan parameter berupa input yaitu folder tempat penyimpanan data LAS lalu parameter out_las_dataset yaitu lokasi penyimpanan LAS dataset yang dihasilkan function dan terakhir adalah spatial_reference yaitu referensi sisten koordinat yang akan digunakan LAS dataset.

Blok script selanjutnya adalah untuk klasifikasi Ground LAS dataset. Pada blok script ini terdapat satu arcpy function yang digunakan yaitu ClassifyLasGround_3d, function ini akan mengklasifikasikan point cloud LAS dataset yang berada di tanah (ground), function ini memerlukan parameter berupa in_las_dataset yaitu LAS dataset yang akan diklasifikasi function ini akan mengedit input data oleh karena itu tidak ada output parameter, untuk parameter lainnya adalah metode yang digunakan untuk klasifikasi, buat seperti gambar di bawah ini.

```

5 def run(homedirectory, lasfolder, lasname, spatialreference):
6     # Create LAS Dataset
7     lasd = arcpy.CreateLasDataset_management(input=lasfolder,
8                                             out_las_dataset=os.path.join(homedirectory, lasname),
9                                             spatial_reference=spatialreference)
10
11     # Classify Ground
12     arcpy.ClassifyLasGround_3d(in_las_dataset=lasd,
13                               method="STANDARD",
14                               reuse_ground="RECLASSIFY_GROUND",
15                               dem_resolution=None,
16                               compute_stats="COMPUTE_STATS",
17                               boundary=None,
18                               process_entire_files="PROCESS_EXTENT")

```

Gambar 2.8 Klasifikasi Ground LAS Dataset

Blok script selanjutnya adalah untuk klasifikasi Bangunan LAS dataset. Pada blok script ini terdapat satu arcpy function yang digunakan yaitu ClassifyLasBuilding_3d, function ini akan mengklasifikasikan point cloud LAS dataset yang berada di Bangunan, function ini memerlukan parameter berupa in_las_dataset yaitu LAS dataset yang akan diklasifikasi function ini akan mengedit input data oleh karena itu tidak ada output parameter, untuk parameter lainnya adalah metode yang digunakan untuk klasifikasi, buat seperti gambar di bawah ini.

```

18 # Classify Building
19 arcpy.ClassifyLasBuilding_3d(in_las_dataset=lasd,
20                             min_height="2 Meters",
21                             min_area="6 SquareMeters",
22                             compute_stats="COMPUTE_STATS",
23                             extent="DEFAULT",
24                             boundary=None,
25                             process_entire_files="PROCESS_EXTENT",
26                             point_spacing=None,
27                             reuse_building="RECLASSIFY_BUILDING",
28                             photogrammetric_data="NOT_PHOTOGAMMETRIC_DATA",
29                             method="STANDARD",
30                             classify_above_roof="NO_CLASSIFY_ABOVE_ROOF",
31                             above_roof_height="1.5 Meters",
32                             above_roof_code=6,
33                             classify_below_roof="NO_CLASSIFY_BELOW_ROOF",
34                             below_roof_code=6)

```

Gambar 2.9 Klasifikasi Bangunan LAS Dataset

Blok script selanjutnya adalah untuk membuat Raster DTM (Digital Terrain Model). Pada blok script ini terdapat tiga arcpy function yang digunakan yaitu MakeLasDatasetLayer_management berguna untuk membuat layer LAS dataset

yang sudah di filter berdasarkan kode kelas point cloud karena untuk membuat DTM kode kelas yang dipakai adalah 2 hasil dari function ini berupa layer LAS dataset yang digunakan untuk function selanjutnya. Lalu function selanjutnya adalah `LasDatasetToRaster_conversion` function ini berguna untuk mengonversi layer LAS dataset menjadi raster, pada function ini terdapat parameter `out_raster` yang berisi lokasi penyimpanan raster isi parameter ini menggunakan function `os.path.join` untuk menggabungkan variable `projectws` dan `dtmname`, selain itu pada function ini juga terdapat variable `sampling_value` yang berisi variable `samplingvalue` untuk parameter lainnya adalah metode yang digunakan untuk interpolasi. Lalu function terakhir adalah `ExtractByMask`, function ini berguna untuk memotong DTM menjadi luasan tertentu menggunakan feature, parameter yang digunakan function ini adalah `in_raster` yaitu raster input dan `in_mask_data` yaitu feature pemotong raster, hasil dari function ini disimpan dalam variable kemudian variable tersebut digunakan dengan function `.save` untuk menyimpannya.

```
# Create DTM
arcpy.MakeLasDatasetLayer_management(in_las_dataset=lasd,
                                     out_layer="dtm_ld_layer",
                                     class_code="2")

dtm = arcpy.LasDatasetToRaster_conversion(in_las_dataset="dtm_ld_layer",
                                          out_raster=os.path.join(projectws, dtmname),
                                          value_field="ELEVATION",
                                          interpolation_type="BINNING MAXIMUM LINEAR",
                                          sampling_type="CELLSIZE",
                                          sampling_value=samplingvalue)

maskdtm = arcpy.sa.ExtractByMask(in_raster=dtm,
                                 in_mask_data=areaofinterest)

maskdtm.save(os.path.join(projectws, dtmname + "_mask_dtm"))]
```

Gambar 2.10 Membuat Raster DTM dan Mask Raster DTM

Blok script selanjutnya adalah untuk membuat Raster DSM (Digital Surface Model). Isi blok script ini hampir sama seperti blok sebelumnya, perbedaannya adalah di parameter function `MakeLasDatasetLayer_management`, pada blok ini parameter `class_code` menggunakan kode 2 dan 6 yang berarti point cloud yang digunakan adalah terrain dan bbuilding. Perbedaan lainnya hanyalah di lokasi penyimpanan dan penamaan output function.

```
# Create DSM
arcpy.MakeLasDatasetLayer_management(in_las_dataset=lasd,
                                     out_layer="dsm_ld_layer",
                                     class_code=["2", "6"],
                                     return_values=["Last return"])

dsm = arcpy.LasDatasetToRaster_conversion(in_las_dataset="dsm_ld_layer",
                                          out_raster=os.path.join(projectws, dsmname),
                                          value_field="ELEVATION",
                                          interpolation_type="BINNING MAXIMUM LINEAR",
                                          sampling_type="CELLSIZE",
                                          sampling_value=samplingvalue)

maskdsm = arcpy.sa.ExtractByMask(in_raster=dsm,
                                 in_mask_data=areaofinterest)

maskdsm.save(os.path.join(projectws, dsmname + "_mask_dsm"))]
```

Gambar 2.11 Membuat Raster DSM dan Mask Raster DSM

Blok script selanjutnya adalah untuk membuat Raster nDSM (Normalize Digital Surface Model). Raster ini dibuat dengan melakukan pengurangan antara

DSM dan DTM sehingga nilai piksel dari nDSM adalah selisih DSM dan DTM. Untuk membuat nDSM function pertama yang digunakan yaitu Minus_3D, function ini berfungsi untuk melakukan operasi pengurangan antara dua raster. Function selanjutnya adalah ExtractByMask sama seperti pada blok script DTM dan DSM.

```
# Create nDSM
ndsm = arcpy.Minus_3d(in_raster_or_constant1=dsm,
                      in_raster_or_constant2=dtm,
                      out_raster=os.path.join(projectws, ndsmname))
maskndsm = arcpy.sa.ExtractByMask(in_raster=ndsm,
                                  in_mask_data=areaofinterest)
maskndsm.save(os.path.join(projectws, ndsmname + "_mask_ndsm"))
```

Gambar 2.12 Membuat Raster nDSM dan Mask Raster nDSM

Blok script selanjutnya adalah untuk clip Footprint Bangunan dengan AOI. Pada blok script ini function yang digunakan adalah Clip_analysis, function ini berguna untuk memotong Footprint Bangunan yang ada di dalam AOI sehingga output dari function ini adalah Footprint Bangunan yang hanya ada di dalam AOI.

```
# Clip Building with AOI
arcpy.Clip_analysis(in_features=buildingfootprint,
                   clip_features=areaofinterest,
                   out_feature_class=os.path.join(projectws, "Clip_Building"))
```

Gambar 2.13 Membuat Clip Footprint Bangunan dengan AOI

Script function run sudah selesai, selanjutnya adalah membuat parameter parameter yang dibutuhkan dari function ini. Periksa script yang sudah dibuat kemudian buat parameter parameternya sehingga menjadi seperti berikut.

```
def run(homedirectory, projectws, lasfolder, lasdname,
       spatialreference, samplingvalue, dsmname, dtmname, ndsmname,
       buildingfootprint, areaofinterest):
```

Gambar 2.14 Membuat Clip Footprint Bangunan dengan AOI

Script parameter sudah lengkap maka file LAS_to_DEM.py ini sudah selesai. Berikut ini adalah isi keseluruhan file.


```

1 import arcpy
2 import os
3
4
5 def run(homedirectory, projectws, lasfolder, lasdname,
6         spatialreference, samplingvalue, dsmname, dtmname, ndsmname,
7         buildingfootprint, areaofinterest):
8     # Create LAS Dataset
9     lasd = arcpy.CreateLasDataset_management(input=lasfolder,
10                                              out_las_dataset=os.path.join(homedirectory, lasdname),
11                                              spatial_reference=spatialreference)
12
13     # Classify Ground
14     arcpy.ClassifyLasGround_3d(in_las_dataset=lasd,
15                                method="STANDARD",
16                                reuse_ground="RECLASSIFY_GROUND",
17                                dem_resolution=None,
18                                compute_stats="COMPUTE_STATS",
19                                boundary=None,
20                                process_entire_files="PROCESS_EXTENT")
21
22     # Classify Building
23     arcpy.ClassifyLasBuilding_3d(in_las_dataset=lasd,
24                                  min_height="2 Meters",
25                                  min_area="6 SquareMeters",
26                                  compute_stats="COMPUTE_STATS",
27                                  extent="DEFAULT",
28                                  boundary=None,
29                                  process_entire_files="PROCESS_EXTENT",
30                                  point_spacing=None,
31                                  reuse_building="RECLASSIFY_BUILDING",
32                                  photogrammetric_data="NOT_PHOTOGAMMETRIC_DATA",
33                                  method="STANDARD",
34                                  classify_above_roof="NO_CLASSIFY_ABOVE_ROOF",
35                                  above_roof_height="1.5 Meters",
36                                  above_roof_code=0,
37                                  classify_below_roof="NO_CLASSIFY_BELOW_ROOF",
38                                  below_roof_code=0)
39
40     # Create DTM
41     arcpy.MakeLasDatasetLayer_management(in_las_dataset=lasd,
42                                          out_layer="dtm_ld_layer",
43                                          class_code="2")

```

```

41 dtm = arcpy.LasDatasetToRaster_conversion(in_las_dataset="dtm_ld_layer",
42                                           out_raster=os.path.join(projectws, dtmname),
43                                           value_field="ELEVATION",
44                                           interpolation_type="BINNING MAXIMUM LINEAR",
45                                           sampling_type="CELLSIZE",
46                                           sampling_value=samplingvalue)
47
48 maskdtm = arcpy.sa.ExtractByMask(in_raster=dtm,
49                                  in_mask_data=areaofinterest)
50 maskdtm.save(os.path.join(projectws, dtmname + "_mask_dtm"))
51
52 # Create DSM
53 arcpy.MakeLasDatasetLayer_management(in_las_dataset=lasd,
54                                     out_layer="dsm_ld_layer",
55                                     class_code=["2", "6"],
56                                     return_values=["Last return"])
57
58 dsm = arcpy.LasDatasetToRaster_conversion(in_las_dataset="dsm_ld_layer",
59                                           out_raster=os.path.join(projectws, dsmname),
60                                           value_field="ELEVATION",
61                                           interpolation_type="BINNING MAXIMUM LINEAR",
62                                           sampling_type="CELLSIZE",
63                                           sampling_value=samplingvalue)
64
65 maskdsm = arcpy.sa.ExtractByMask(in_raster=dsm,
66                                  in_mask_data=areaofinterest)
67 maskdsm.save(os.path.join(projectws, dsmname + "_mask_dsm"))
68
69 # Create nDSM
70 ndsm = arcpy.Minus_3d(in_raster_or_constant1=dsm,
71                       in_raster_or_constant2=dtm,
72                       out_raster=os.path.join(projectws, ndsmname))
73
74 maskndsm = arcpy.sa.ExtractByMask(in_raster=ndsm,
75                                   in_mask_data=areaofinterest)
76 maskndsm.save(os.path.join(projectws, ndsmname + "_mask_ndsm"))
77
78 # Clip Building with AOI
79 arcpy.Clip_analysis(in_features=buildingfootprint,
80                     clip_features=areaofinterest,
81                     out_feature_class=os.path.join(projectws, "Clip_Building"))

```

Gambar 2.15 Script LAS_to_DEM.py

Selanjutnya adalah membuat UI tools LAS to DEM ini. Untuk mendesain UI file yang akan di edit adalah LAS_to_DEM_UI.py. Mulai mengedit dengan memasukkan module yang diperlukan. Module yang diperlukan adalah arcpy dan LAS_to_DEM yang ada di dalam folder script.

```
1 import arcpy
2 from scripts import LAS_to_DEM
3
```

Gambar 2.16 Module yang digunakan file LAS_to_DEM_UI.py

Selanjutnya adalah membuat variable default yang akan digunakan pada function yang akan dibuat. Selain itu konfigurasi juga environment overwriteOutput menjadi True, dengan konfigurasi environment ini tools akan bisa dijalankan berkali kali tanpa mengubah lokasi output karena output akan overwrite file sebelumnya.

```
3
4 # environment
5 arcpy.env.overwriteOutput = True
6
7 # default variables
8 aprx = arcpy.mp.ArcGISProject("CURRENT")
9 home_directory = aprx.homeFolder
10 project_ws = aprx.defaultGeodatabase
11
```

Gambar 2.17 Variable default dan konfigurasi environment

Selanjutnya adalah konfigurasi variable yang tidak default. Variable yang tidak default ini memungkinkan user untuk mengubah-ubah isi variable, supaya user bisa melakukan input untuk mengisi variable maka isi dari variable harus menggunakan function arcpy.GetParameterAsText, parameter function ini adalah index number, index ini akan menjadi refrensi pembuatan parameter di ArcGIS Pro. Buat variabel non default menjadi seperti ini.

```
11
12 # non default variables
13 input_las_folder = arcpy.GetParameterAsText(0)
14 las_dataset_name = arcpy.GetParameterAsText(1)
15 spatial_reference = arcpy.GetParameterAsText(2)
16 sampling_value = arcpy.GetParameterAsText(3)
17 dsm_name = arcpy.GetParameterAsText(4)
18 dtm_name = arcpy.GetParameterAsText(5)
19 ndsm_name = arcpy.GetParameterAsText(6)
20 building_footprint = arcpy.GetParameterAsText(7)
21 area_of_interest = arcpy.GetParameterAsText(8)
22
```

Gambar 2.18 Non default variable

Selanjutnya adalah menjalankan function run yang sudah dibuat di file LAS_to_DEM.py. Function ini memerlukan parameter parameter, isi parameter parameter tersebut dengan variable default dan non default yang sudah dibuat sebelumnya sehingga keseluruhan file menjadi seperti berikut.

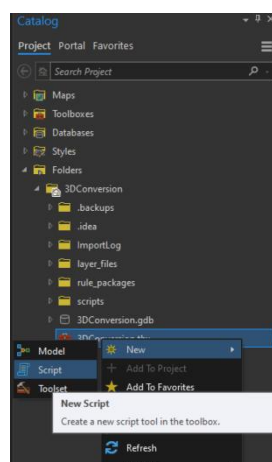
```

1 import arcpy
2 from scripts import LAS_to_DEM
3
4 # environment
5 arcpy.env.overwriteOutput = True
6
7 # default variables
8 aprx = arcpy.mp.ArcGISProject("CURRENT")
9 home_directory = aprx.homeFolder
10 project_ws = aprx.defaultGeodatabase
11
12 # non default variables
13 input_las_folder = arcpy.GetParameterAsText(0)
14 las_dataset_name = arcpy.GetParameterAsText(1)
15 spatial_reference = arcpy.GetParameterAsText(2)
16 sampling_value = arcpy.GetParameterAsText(3)
17 dsm_name = arcpy.GetParameterAsText(4)
18 dtm_name = arcpy.GetParameterAsText(5)
19 ndsm_name = arcpy.GetParameterAsText(6)
20 building_footprint = arcpy.GetParameterAsText(7)
21 area_of_interest = arcpy.GetParameterAsText(8)
22
23 LAS_to_DEM.run(homedirectory=home_directory,
24               projectws=project_ws,
25               lasfolder=input_las_folder,
26               lasdname=las_dataset_name,
27               spatialreference=spatial_reference,
28               samplingvalue=sampling_value,
29               dsmname=dsm_name,
30               dtmname=dtm_name,
31               ndsmname=ndsm_name,
32               buildingfootprint=building_footprint,
33               areaofinterest=area_of_interest)
34

```

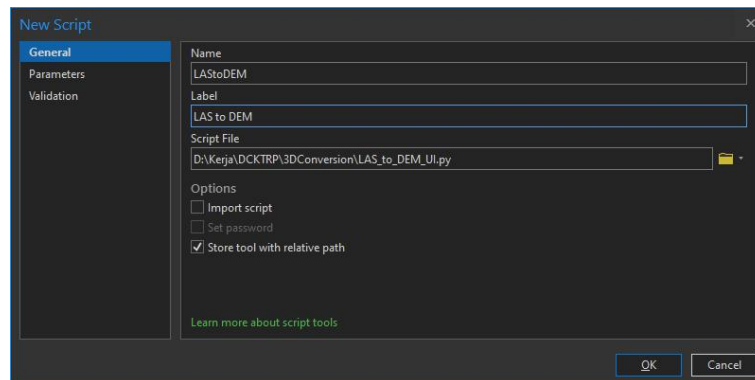
Gambar 2.19 Isi script file LAS_to_DEM_UI.py

Kedua file python yang sudah dibuat sudah selesai maka selanjutnya adalah import script UI ke dalam ArcGIS Pro sehingga menjadi tools. Buka folder project pada catalog ArcGIS Pro, secara default ketika membuat project ArcGIS Pro pada folder project akan terbuat file toolbox. Klik kanan pada toolbox kemudian pilih New lalu Script.



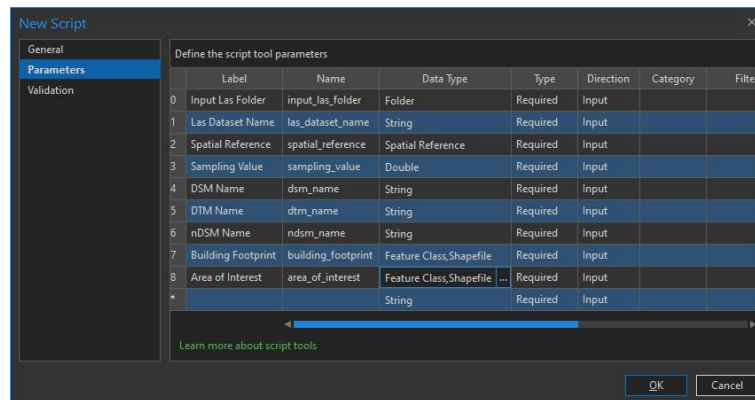
Gambar 2.20 Membuat script baru pada toolbox

Setelah itu akan muncul window New Script. Pada tab General isikan Name dan Label, name tidak bisa menggunakan spasi sedangkan label bisa. Lalu pada kolom Script File browse file LAS_to_DEM_UI.py.



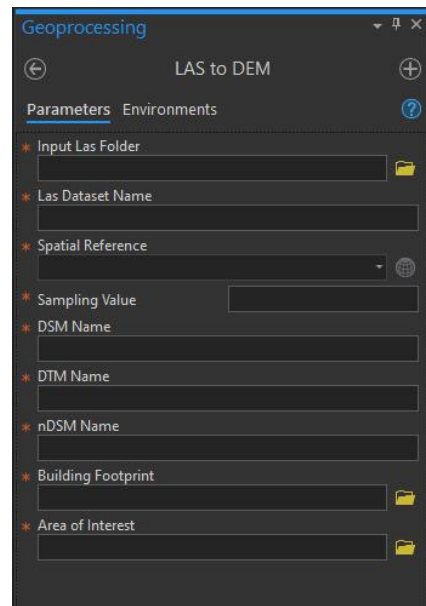
Gambar 2.21 Konfigurasi Tab General New Script

Selanjutnya klik Tab Parameters, pada tab ini konfigurasi parameter dari tools akan dibuat. Parameter yang perlu dibuat di sini hanya parameter yang berisi variabel non default atau menggunakan function arcpy.GetParameterAsText. Fucntion ini berisikan nomor indeks, olah karena itu pembuatan parameter harus menyesuaikan index yang ada di script. Setelah parameter dibuat menyesuaikan nomor indeks, konfigurasikan Data Type tiap parameter, Data Type parameter ditentukan oleh perannya di dalam Arcpy Function. Buat konfigurasi seperti berikut.



Gambar 2.22 Konfigurasi Tab Parameters New Script

Sampai tahap ini tools sudah selsai dibuat dan bisa dijalankan berikut adalah tampilan tools ketika dibuka.



Gambar 2.23 Tampilan Tool LAS to DEM

2.1.2. DEM to Polygon 3D

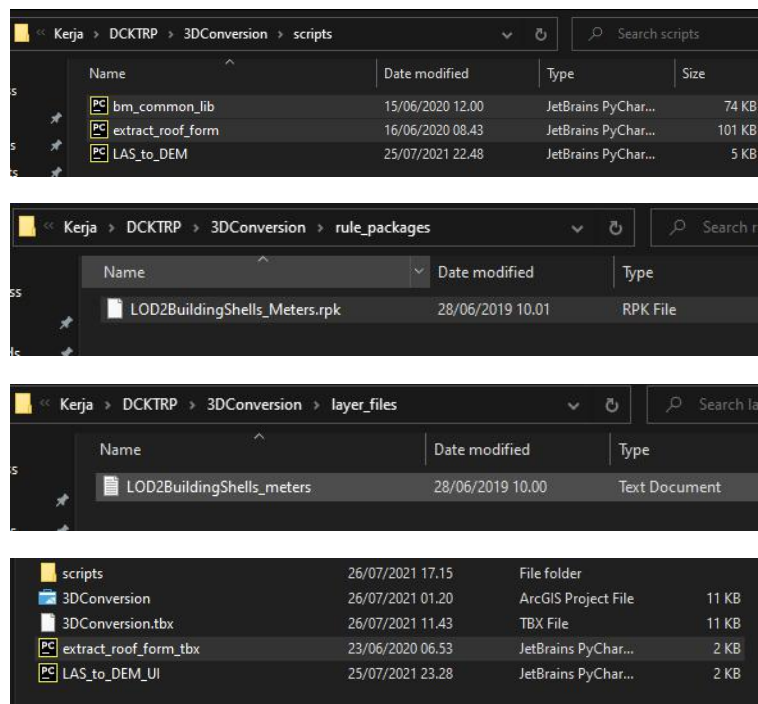
Selanjutnya adalah tools DEM to Polygon 3D. Tools ini tidak akan dibuat sendiri, tools ini didapatkan dengan mengunduhnya dari situs milik esri (<https://learn.arcgis.com/en/projects/extract-roof-forms-for-municipal-development>) yang berisi tutorial membuat multipatch. Berikut ini adalah file hasil download dari web esri.

Name	Date modified	Type	Size
.backups	18/07/2021 18.37	File folder	
3DBasemaps.gdb	26/07/2021 11.24	File folder	
Index	18/07/2021 18.37	File folder	
layer_files	17/07/2021 16.39	File folder	
rasterfunctionshistory	17/07/2021 16.39	File folder	
RasterFunctionTemplates	17/07/2021 16.39	File folder	
rule_packages	17/07/2021 16.39	File folder	
scripts	20/07/2021 17.35	File folder	
scripts_ddd	20/07/2021 17.35	File folder	
settings	17/07/2021 16.39	File folder	
tables	17/07/2021 16.39	File folder	
tasks	17/07/2021 16.39	File folder	
3Dbasemaps	18/07/2021 18.37	ArcGIS Project File	65 KB
3DBasemaps.tbx	23/06/2020 06.53	TBX File	39 KB
confidence_measurement_tbx	23/06/2020 06.53	JetBrains PyChar...	1 KB
create_building_mosaic_tbx	23/06/2020 06.53	JetBrains PyChar...	1 KB
create_distribution_lines_tbx	23/06/2020 06.53	JetBrains PyChar...	2 KB
create_transmission_lines_tbx	23/06/2020 06.53	JetBrains PyChar...	2 KB
CreateBridges.tbx	23/06/2020 06.53	TBX File	18 KB
CreateBuildings.tbx	23/06/2020 06.53	TBX File	126 KB
CreateFloors.tbx	23/06/2020 06.53	TBX File	47 KB
CreateFootPrints.tbx	23/06/2020 06.53	TBX File	310 KB
CreatePowerlines.tbx	23/06/2020 06.53	TBX File	73 KB
CreateTrees.tbx	23/06/2020 06.53	TBX File	27 KB
detect_footprint_changes_tbx	23/06/2020 06.53	JetBrains PyChar...	2 KB
extract_bridges_from_las_tbx	23/06/2020 06.53	JetBrains PyChar...	1 KB
extract_elevation_from_las_tbx	23/06/2020 06.53	JetBrains PyChar...	4 KB
extract_roof_form_tbx	23/06/2020 06.53	JetBrains PyChar...	2 KB
extrude_by_attribute_tbx	23/06/2020 06.53	JetBrains PyChar...	1 KB
footprints_from_raster_tbx	23/06/2020 06.53	JetBrains PyChar...	2 KB
fuse_building_parts_tbx	23/06/2020 06.53	JetBrains PyChar...	1 KB
mask_features_by_layer_tbx	23/06/2020 06.53	JetBrains PyChar...	1 KB
merge_features_tbx	23/06/2020 06.53	JetBrains PyChar...	1 KB
modify_dtm_tbx	23/06/2020 06.53	JetBrains PyChar...	1 KB
roof_part_segmentation_tbx	23/06/2020 06.53	JetBrains PyChar...	2 KB
set_texture_quality_attribute_tbx	23/06/2020 06.53	JetBrains PyChar...	1 KB
split_buildings_into_floors_tbx	23/06/2020 06.53	JetBrains PyChar...	3 KB
split_features_tbx	23/06/2020 06.53	JetBrains PyChar...	1 KB

Gambar 2.24 File hasil download dari web esri

Copy files yang diperlukan kemudian paste ke dalam project folder, file file yang diperlukan yaitu:

1. layer_files/LOD2BuildingShells_meters.txt paste ke folder layer_files yang ada di folder project
2. rule_packages/LOD2BuildingShells_meters.rpk paste ke folder rule_packages yang ada di folder project
3. scripts/extract_roof_form.py paste ke folder scripts yang ada di folder project
4. scripts/bm_common_lib.py paste ke folder scripts yang ada di folder project
5. extract_roof_form_tbx.py paste ke folder project



Gambar 2.25 File file yang di copy dari file download dari web esri

Setelah file di paste ke folder project, selanjutnya adalah memodifikasi file. Edit file `extract_roof_form.py` yang ada di dalam folder script menggunakan pycharm. Pada script delete dari line 2317 sampai 2319, alasan mengapa kita harus mendownload ini adalah karena script ini dibuat untuk berjalan di dalam package sedangkan tools yang diinginkan tidak di dalam package.

```

2311
2312     # fail safe for European comma's
2313     min_flat_roof_area = float(re.sub("[.]", ",", min_flat_roof_area))
2314     min_slope_roof_area = float(re.sub("[.]", ",", min_slope_roof_area))
2315     min_roof_height = float(re.sub("[.]", ",", min_roof_height))
2316     simplify_tolerance = float(re.sub("[.]", ",", simplify_tolerance))
2317
2318     if os.path.exists(os.path.join(home_directory, "p20")): # it is a package
2319         home_directory = os.path.join(home_directory, "p20")
2320
2321     arcpy.AddMessage("Project Home Directory is: " + home_directory)
2322

```

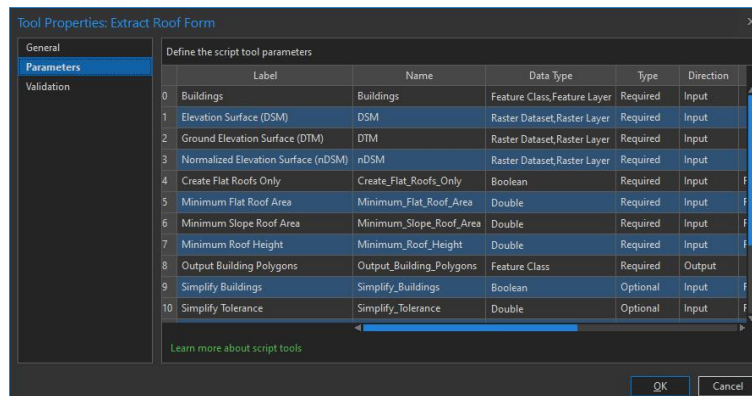
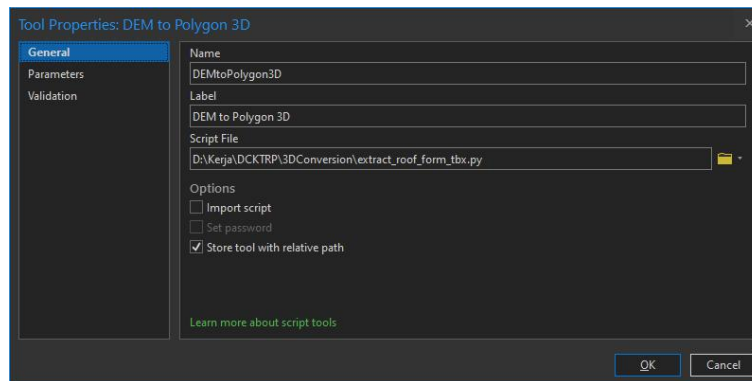
```

2312 # fail safe for European comma's
2313 min_flat_roof_area = float(re.sub("[,]", ".", min_flat_roof_area))
2314 min_slope_roof_area = float(re.sub("[,]", ".", min_slope_roof_area))
2315 min_roof_height = float(re.sub("[,]", ".", min_roof_height))
2316 simplify_tolerance = float(re.sub("[,]", ".", simplify_tolerance))
2317
2318 arcpy.AddMessage("Project Home Directory is: " + home_directory)
2319
2320 for a in home_directory:
2321     if a.isspace():
2322         raise HasSpace

```

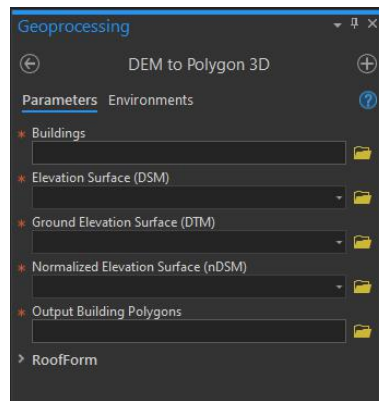
Gambar 2.26 Modifikasi file extract_roof_form.py

Selanjutnya adalah import script ke dalam toolbox. Sama seperti tools LAS to DEM konfigurasi yang diperlukan untuk import script ke dalam toolbox hanya perlu di tab general dan parameter saja. Berikut adalah konfigurasinya.



Gambar 2.27 Konfigurasi Tool DEM to Polygon 3D di Toolbox

Setelah konfigurasi tool selesai maka tampilan tool akan menjadi seperti berikut. Tool akan memerlukan input dari data data output tool LAS to DEM yaitu DTM, DSM, nDSM dan Footprint Clip Bangunan, sedangkan output yang dihasilkan adalah polygon yang memiliki 3D atribut.

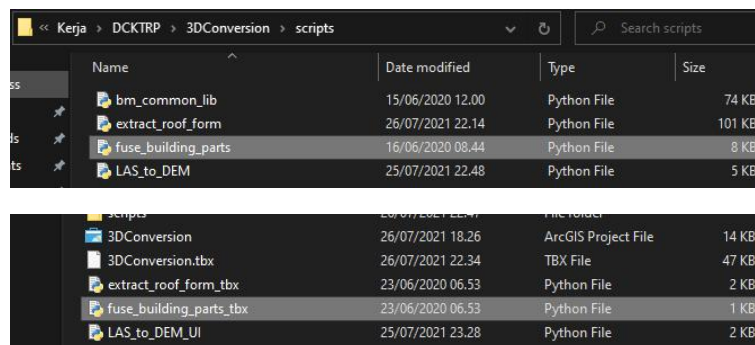


Gambar 2.28 Tampilan Tool DEM to Polygon 3D

2.1.3. Polygon 3D to Multipatch

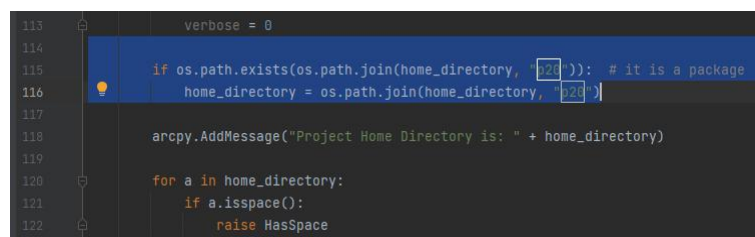
Tool terakhir adalah Polygon 3D to Multipatch. Tool ini akan mengubah Polygon 3D hasil output tool sebelumnya menjadi data multipatch. Sama seperti tool DEM to Polygon 3D tool ini juga dapat di dapat dari web esri. Pada folder download package esri copy dan paste data berikut ini:

1. scripts/fuse_building_parts.py paste ke folder script yang ada di folder project
2. fuse_building_parts_tbx.py paste ke folder project



Gambar 2.29 File yang perlukan untuk tool Polygon 3D to Multipatch

Sama seperti tool DEM to Polygon 3D file yang sudah di copy perlu diedit terlebih dahulu. Edit yang dilakukan juga sama seperti sebelumnya yaitu menghapus bagian script yang berisi konfigurasi script di dalam package. Pada script file fuse_building_parts.py hapus line 114 sampai line 116.



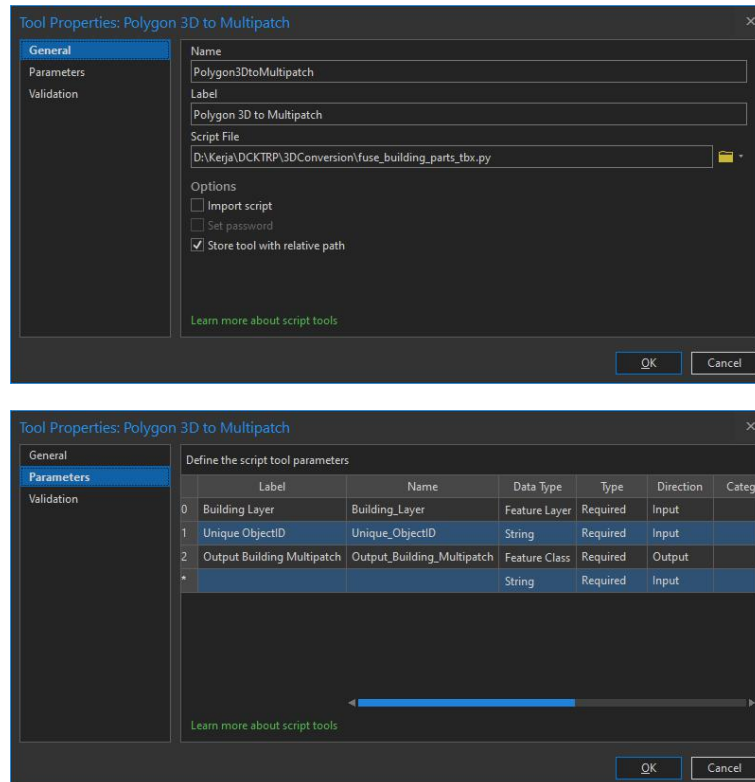
```

111         else:
112             delete_intermediate_data = True
113             verbose = 0
114
115             arcpy.AddMessage("Project Home Directory is: " + home_directory)
116
117             for a in home_directory:
118                 if a.isspace():
119                     raise HasSpace
120

```

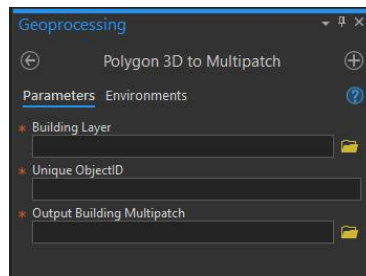
Gambar 2.30 Modifikasi file fuse_building_parts.py

Selanjutnya adalah import script menjadi tool di dalam toolbox. Import seperti dua tool sebelumnya dengan konfigurasi seperti berikut.



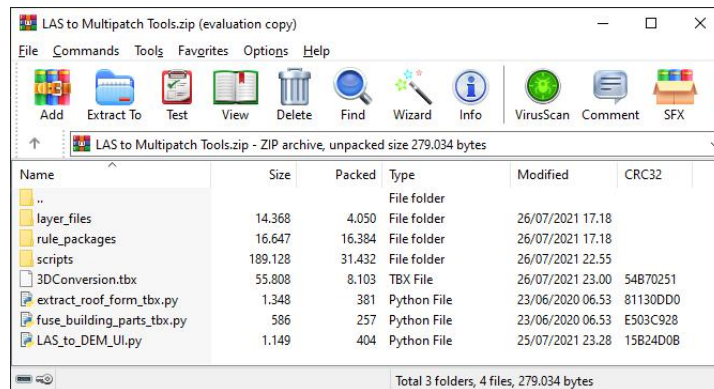
Gambar 2.31 Konfigurasi Tool Polygon 3D to Multipatch di Toolbox

Setelah konfigurasi tool selesai maka tampilan tool akan menjadi seperti berikut. Tool akan memerlukan input dari data data output tool DEM to Polygon 3D yaitu Feature Class polygon 3D yang ditampilkan dalam Feature Layer 3D Scene Layer simbologi Feature Layer ini menggunakan tool Apply Simbology From Layer dan untuk layer file menggunakan LOD2BuildingShells_meters.lyr yang di copy dari folder download package esri, dengan menjalankan tool Apply Simbology From Layer maka polygon akan ditampilkan menjadi model bangunan 3D walaupun data ini masih Feature Class bukan Multipatch. Tool Polygon 3D to Multipatch ini akan membuat model 3D polygon bangunan ini menjadi mutlipatch. Berikut adalah tampilan tool.



Gambar 2.32 Tampilan Tool Polygon 3D to Multipatch

Setelah semua tools selesai dibuat zip file yang diperlukan supaya bisa digunakan untuk user lainnya. Isi zip file dengan semua resource file menjadi seperti berikut.



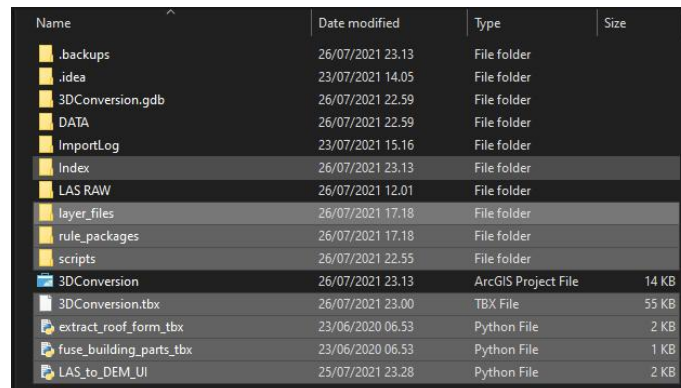
Gambar 2.33 Zip file semua resource tools

BAB III

HASIL PELAKSANA PEKERJAN

3.1 PEMBUATAN TOOLS PROCESSING LAS TO MULTIPATCH

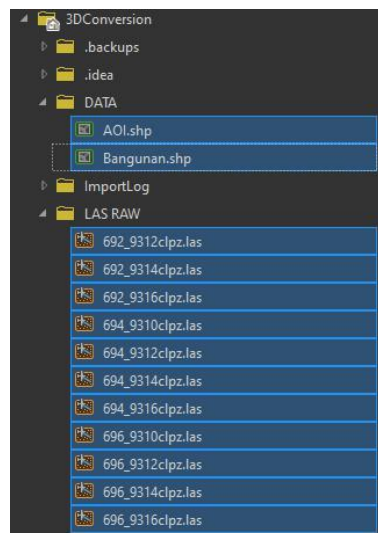
Hasil pekerjaan ini berupa file zip yang berisi resource untuk menjalankan tool. Untuk menggunakannya file zip ini bisa di extract di folder project user.



Name	Date modified	Type	Size
.backups	26/07/2021 23.13	File folder	
.idea	23/07/2021 14.05	File folder	
3DConversion.gdb	26/07/2021 22.59	File folder	
DATA	26/07/2021 22.59	File folder	
ImportLog	23/07/2021 15.16	File folder	
Index	26/07/2021 23.13	File folder	
LAS RAW	26/07/2021 12.01	File folder	
layer_files	26/07/2021 17.18	File folder	
rule_packages	26/07/2021 17.18	File folder	
scripts	26/07/2021 22.55	File folder	
3DConversion	26/07/2021 23.13	ArcGIS Project File	14 KB
3DConversion.tbx	26/07/2021 23.00	TBX File	55 KB
extract_roof_form_tbx	23/06/2020 06.53	Python File	2 KB
fuse_building_parts_tbx	23/06/2020 06.53	Python File	1 KB
LAS_to_DEM_UI	25/07/2021 23.28	Python File	2 KB

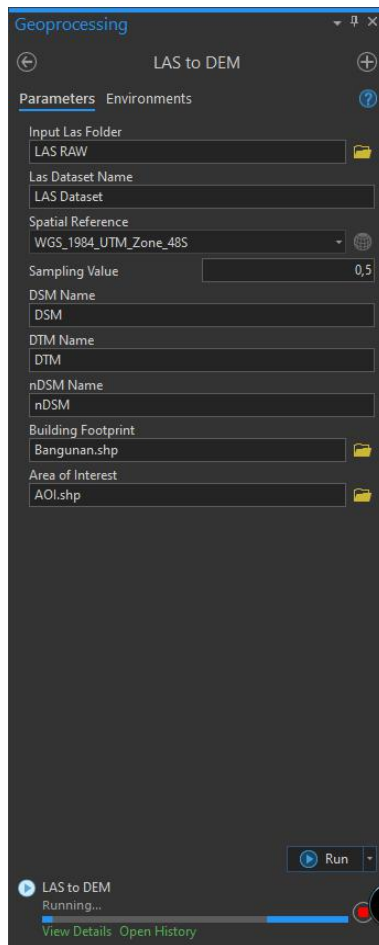
Gambar 3.1 File dan folder resource tool

Untuk menjalankan tool ini data yang diperlukan harus dipersiapkan terlebih dahulu dan disimpan di dalam project folder. Berikut ini adalah data yang diperlukan untuk menjalankan tools.

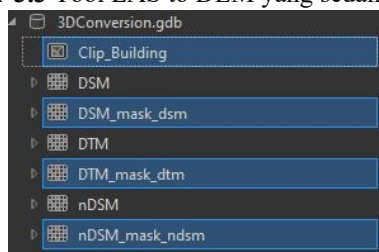


Gambar 3.2 File file yang diperlukan untuk menjalankan tools

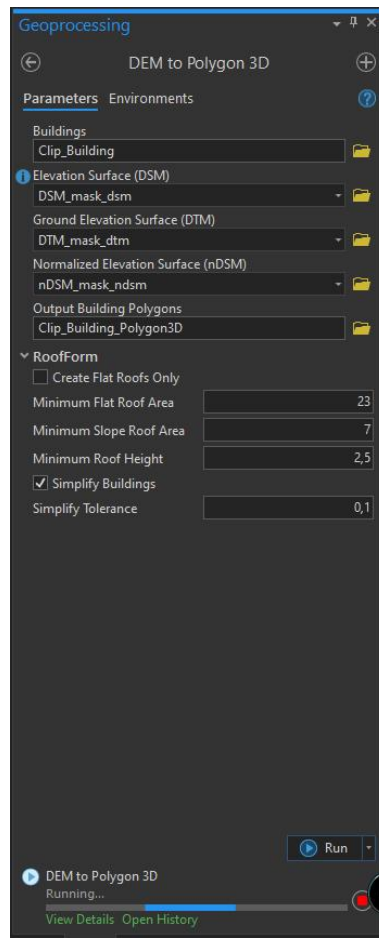
Berikut ini adalah pengujian tools dan output yang dihasilkan tools. Hasil output akan tersimpan di geodatabase default project.



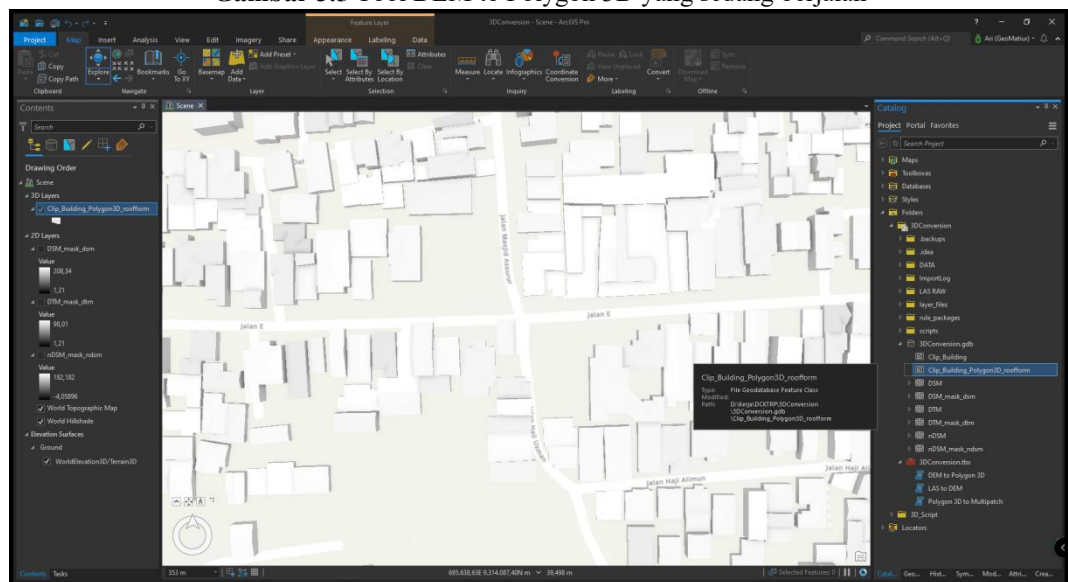
Gambar 3.3 Tool LAS to DEM yang sedang berjalan



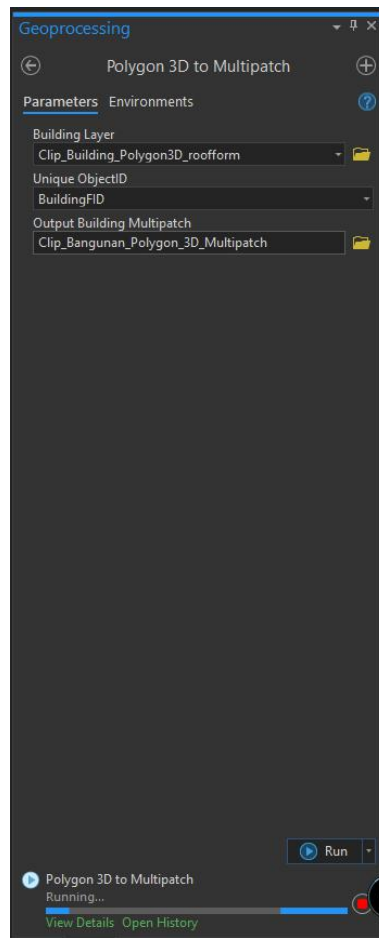
Gambar 3.4 Output file hasil tool LAS to DEM



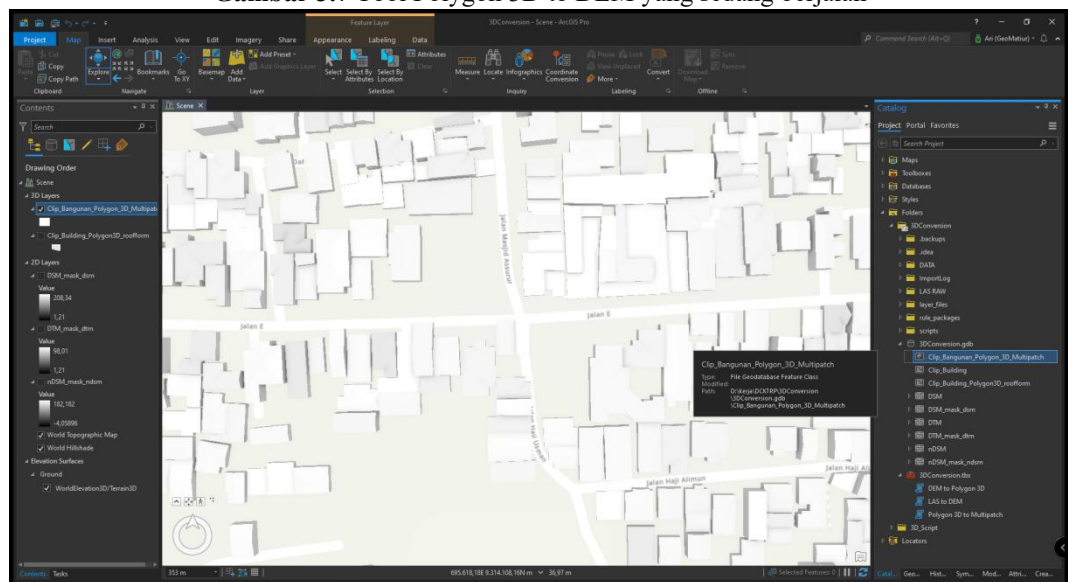
Gambar 3.5 Tool DEM to Polygon 3D yang sedang berjalan



Gambar 3.6 Output file hasil tool DEM to Polygon 3D



Gambar 3.7 Tool Polygon 3D to DEM yang sedang berjalan



Gambar 3.8 Output file hasil tool Polygon 3D to Multipatch

BAB IV

KESIMPULAN DAN REKOMENDASI

4.1 PEMBUATAN TOOLS PROCESSING LAS TO MULTIPATCH

Dengan adanya tools ini bisa mempercepat dan mengurangi kemungkinan kesalahan saat mengolah data LAS. Pembuatan tools menjadi satu tool saja bisa dilakukan tapi akan sulit untuk melakukan pengecekan (debugging) karena script yang dibuat akan menjadi sangat panjang. Untuk pekerjaan lainnya yang memerlukan prosedur sama berulang kali bisa dibuat tool supaya memudahkan pekerjaan.

LAMPIRAN

NO	TGL/BLN /TH	WAKTU	KEGIATAN	KETERANGAN
1	Kamis 1 Juli 2021	08.00-12.00	Membuat webmap untuk collector survey RTH	
		12.00-13.00	ISTIRAHAT	
		13.00-16.00	Membuat webmap untuk collector survey RTH	
	TGL/BLN /TH	WAKTU	KEGIATAN	KETERANGAN
2	Jumat 2 Juli 2021	07.30-12.00	Membuat webmap untuk collector survey RTH	
		12.00-13.00	ISTIRAHAT	
		13.00-16.00	Membuat webmap untuk collector survey RTH	
	TGL/BLN /TH	WAKTU	KEGIATAN	KETERANGAN
3	Senin 5 Juli 2021	07.30-11.30	Rapat Sumur Resapan Bangunan Tinggi	
		11.30-13.30	ISTIRAHAT	
		13.30-16.30	Rapat Sumur Resapan Bangunan Tinggi	
	TGL/BLN /TH	WAKTU	KEGIATAN	KETERANGAN
4	Selasa 6 Juli 2021	07.30-12.00	Rapat Sumur Resapan Bangunan Tinggi	
		12.00-13.00	ISTIRAHAT	
		13.00-16.00	Rapat Sumur Resapan Bangunan Tinggi	
	TGL/BLN /TH	WAKTU	KEGIATAN	KETERANGAN
5	Rabu 7 Juli 2021	07.30-12.00	Rapat Aplikasi Kelurahan	
		12.00-13.00	ISTIRAHAT	
		13.00-16.00	Rapat Aplikasi Kelurahan	

NO	TGL/BLN /TH	WAKTU	KEGIATAN	KETERANGAN
6	Kamis 8 Juli 2021	08.00-12.00	Rapat Pembuatan ArcGIS Urban Jakarta	
		12.00-13.00	ISTIRAHAT	
		13.00-16.00	Rapat Pembuatan ArcGIS Urban Jakarta	
	TGL/BLN /TH	WAKTU	KEGIATAN	KETERANGAN
7	Jumat 9 Juli 2021	07.30-12.00	Membuat Tools LAS to Multipatch	
		12.00-13.00	ISTIRAHAT	
		13.00-16.00	Membuat Tools LAS to Multipatch	
	TGL/BLN /TH	WAKTU	KEGIATAN	KETERANGAN
8	Senin 12 Juli 2021	07.30-11.30	Membuat Tools LAS to Multipatchi	
		11.30-13.30	ISTIRAHAT	
		13.30-16.30	Membuat Tools LAS to Multipatch	
	TGL/BLN /TH	WAKTU	KEGIATAN	KETERANGAN
9	Selasa 13 Juli 2021	07.30-12.00	Membuat Tools LAS to Multipatch	
		12.00-13.00	ISTIRAHAT	
		13.00-16.00	Membuat webmap untuk collector survey RTH	
	TGL/BLN /TH	WAKTU	KEGIATAN	KETERANGAN
10	Rabu 14 Juli 2021	07.30-12.00	Membuat Tools LAS to Multipatch	
		12.00-13.00	ISTIRAHAT	
		13.00-16.00	Memperbaiki Halaman Untuk Batas Administrasi Jakarta Satu	

NO	TGL/BLN /TH	WAKTU	KEGIATAN	KETERANGAN
11	Kamis 15 Juli 2021	08.00-12.00	Membuat Tools LAS to Multipatch	
		12.00-13.00	ISTIRAHAT	
		13.00-16.00	Membuat Tools LAS to Multipatch	
	TGL/BLN /TH	WAKTU	KEGIATAN	KETERANGAN
12	Jumat 16 Juli 2021	07.30-12.00	Membuat Tools LAS to Multipatch	
		12.00-13.00	ISTIRAHAT	
		13.00-16.00	Membuat Tools LAS to Multipatch	
	TGL/BLN /TH	WAKTU	KEGIATAN	KETERANGAN
13	Senin 19 Juli 2021	07.30-11.30	Membuat Tools LAS to Multipatch	
		11.30-13.30	ISTIRAHAT	
		13.30-16.30	Membuat Tools LAS to Multipatch	
	TGL/BLN /TH	WAKTU	KEGIATAN	KETERANGAN
14	Rabu 21 Juli 2021	07.30-12.00	Membuat Tools LAS to Multipatch	
		12.00-13.00	ISTIRAHAT	
		13.00-16.00	Membuat Tools LAS to Multipatch	
	TGL/BLN /TH	WAKTU	KEGIATAN	KETERANGAN
15	Kamis 22 Juli 2021	07.30-12.00	Membuat Tools LAS to Multipatch	
		12.00-13.00	ISTIRAHAT	
		13.00-16.00	Membuat Tools LAS to Multipatch	

NO	TGL/BLN /TH	WAKTU	KEGIATAN	KETERANGAN
16	Jumat 23 Juli 2021	08.00-12.00	Membuat Dashboard Informasi Sumur Resapan	
		12.00-13.00	ISTIRAHAT	
		13.00-16.00	Membuat Dashboard Informasi Sumur Resapan	
	TGL/BLN /TH	WAKTU	KEGIATAN	KETERANGAN
17	Senin 26 Juli 2021	07.30-12.00	Membuat Experience Informasi Sumur Resapan	
		12.00-13.00	ISTIRAHAT	
		13.00-16.00	Membuat laporan bulan Juli 2021	
	TGL/BLN /TH	WAKTU	KEGIATAN	KETERANGAN
18	Selasa 27 Juli 2021	07.30-11.30	Membuat laporan bulan Juli 2021	
		11.30-13.30	ISTIRAHAT	
		13.30-16.30	Membuat laporan bulan Juli 2021	
	TGL/BLN /TH	WAKTU	KEGIATAN	KETERANGAN
19	Rabu 28 Juli 2021	07.30-12.00	Mengumpulkan dokumen SPK,BA,SPPBJ	
		12.00-13.00	ISTIRAHAT	
		13.00-16.00	Rapat Standarisasi KUGI Data DSDA	
	TGL/BLN /TH	WAKTU	KEGIATAN	KETERANGAN
20	Kamis 29 Juli 2021	07.30-12.00	Vaksin	
		12.00-13.00	ISTIRAHAT	
		13.00-16.00	Membuat laporan bulan Juli 2021	
	TGL/BLN /TH	WAKTU	KEGIATAN	KETERANGAN
21	Jumat 30 Juli 2021	07.30-12.00	Memperbaiki data polder DSDA	
		12.00-13.00	ISTIRAHAT	
		13.00-16.00	Memperbaiki data polder DSDA	