

Automation of Daily Landslide Potential Information Based on Remote Sensing Satellite Imagery Using Open-Source Software Technology

Ahmad Sutanto¹, Anwar Annas², Mohammad Ardha³,
Taufik Hidayat⁴, Muhammad Rokhis Khomarudin⁵

^{1,3,5} Remote Sensing Research Center – National Research and Innovation Agency (BRIN), Indonesia

^{2,4} Directorate of Laboratory Management, Research Facilities and Science Technology Area –
National Research and Innovation Agency (BRIN), Indonesia

sutantoahmad99@gmail.com

Received: 11.01.2023 ; Revised: 16.05.2023 ; Approved: 05.06.2023

ABSTRACT. This automation system automatically generated landslide potential information based on daily precipitation data. This system is essential to replace the previous manual processing system with an automated and integrated system. The results of the developed system are the distribution of areas with landslide potential based on daily precipitation data. The system was built using QGIS Processing Modeler Algorithms and Python script programming. This allows the automation process to be performed quickly and accurately. The landslide susceptibility map used is from the National Disaster Management Agency, so the information is more reliable. Himawari-8 is used to determine the potential for extreme precipitation in 10 minutes because this satellite has a very high temporal resolution. The system is already in use and has proven to replace manual processing and is faster than landslide information manual processing which executed each algorithm separately. Further development will be more challenging if the system can be connected to the sensors installed on site so that the sensors on site can issue a landslide warning in case of extreme precipitation so that the surrounding communities can respond immediately. Opportunities for future development of the system may also be incorporated into landslide potential prediction based on the precipitation forecast model.

Keywords: *landslide, rainfall, Himawari 8, early warning system*

1 INTRODUCTION

Landslides are disasters that frequently occur in Indonesia, especially during the rainy season in areas with steep slopes. In addition to soil type and rainfall intensity, land cover change also dramatically influences the occurrence of landslides (Susanti and Miardini 2019, Muzani *et al.* 2021). Remote sensing is a technology that can be used to analyze changes in land cover (Tarantino *et al.* 2007) and precipitation conditions (Adiningsih and Khomarudin 1998). In addition, Himawari-8 data with high temporal resolution can be used for real-time precipitation monitoring (Hirose *et al.* 2019, Ayasha 2020). This capability can establish an early warning system for landslide hazards (Miura and Nagai 2020, Budiati *et al.* 2020). An early warning requires fast and accurate information (Xu *et al.* 2020). For this

purpose, a system must be developed to provide early warning of landslides. This paper presents the development of an automation system that uses remote sensing data and open-source technology to detect the potential for daily landslides. The GIS technology supports the development of the automation system.

Landslide monitoring information systems have been developed in several countries, including using a web-based geographic information system (Tengtrairat *et al.* 2021, Ahmed *et al.* 2018), a landslide information system with a sensor system installed on-site (Giri *et al.* 2019, Jeong *et al.* 2019), and based on IOT (Aggarwal *et al.* 2018, Thirugnanam *et al.* 2022). In Indonesia, sensor-based systems and IoT have begun to be developed, but most research uses geographic information systems to determine the extent of landslide hazards (Bachri *et al.* 2021, Nurdin and Kubota

2018, Lukman and Le 2020). The development of an information system using remote sensing satellite data with a very high temporal resolution, such as Himawari-8, is still rare. Developing a landslide monitoring system using such data is necessary to provide information quickly, cheaply, and accurately. This paper focuses on developing a system that can monitor and real-time early warning for daily landslide potential.

Indonesia received Himawari-8 data that can be used to monitor rainfall conditions in real-time. On the other hand, the National Disaster Management Agency (BNPB) has released a map of landslide-prone areas throughout Indonesia. When these two data are combined, an early warning system for landslides in Indonesia can be established. However, manual data processing takes quite a long time, considering the large amount of data for a very large area in Indonesia. As a result, the early warning system does not work correctly. Therefore, developing an automation system to generate information about the potential of landslide areas in Indonesia is very important. In this paper, the development of the system is presented.

The purpose of developing a daily landslide potential automation system is to quickly provide information on the potential for landslides in a given area. Rainfall is the most important factor in providing landslide warnings. In addition, the system can be linked to BNPB's system so that BNPB can directly access it and issue daily warnings for landslide-prone areas.

2 METHODOLOGY

2.1 Data

The input data used are precipitation data extracted from the Himawari-8 satellite image. The sample data used in this process is "akumulasi_produkRFR2_20191208-000000_20191206-235000.tif", which contains daily accumulated precipitation data for December 8, 2019 between 00:00 and 23:50. We also use vector data for landslide-prone areas from the BNPB. These landslide-prone area data are needed to create daily landslide potential

areas based on precipitation data in that area. The vector data for the landslide-prone area is "Area_rawan_longsor_BNPB.shp".

2.2 Methods

A workflow from GIS typically goes through several stages, with one stage producing a secondary output used in the next stage. We must perform the whole process manually if we want to change the input data or optimize a parameter. QGIS software has a built-in graphical modeler. It can help us define and execute our workflow with a single command. We can run this workflow as a single layer or as multiple input layers. We can also convert the flow modeler into a Python script (PYQGIS developer Cookbook). We use the Processing Modeler included in the QGIS 3.10 software in this process.

In this study, we develop a model to extract information about landslide-prone areas based on daily precipitation data obtained from Himawari-8 satellite imagery. The daily precipitation data are filtered for a range of values that can cause landslides, i.e., precipitation values greater than or equal to 120 mm, and then converted into polygons. These polygons are clipped with polygon vector data for landslide-prone areas from the National Disaster Mitigation Agency (BNPB). The results of this pruning process are then used to calculate the area of each polygon, resulting in an area with the potential for daily landslides with an area of acres. The statistical value of precipitation in each polygon with daily landslide potential is then calculated. The calculated statistical values are the average value (mean), minimum value, maximum value, and mode value of precipitation in each polygon with daily landslide potential. To calculate these statistical values, we use the Zonal Statistics function. In this study, a complex workflow is implemented (Ujaval 2012).

Our workflow for this process is divided into the following phases.

1. Apply a Reclassify algorithm to the daily precipitation grid data. It filters the precipitation data with values ≥ 120 mm and classifies it into a new digital value, with

- values outside this value representing no data.
- Converts the output raster to a precipitation polygon layer that has the potential to cause landslides.
 - Performs the clipping process between two vector data, namely the daily precipitation polygon and the landslide-prone area polygon from the BNPB.
 - Calculate the area resulting from the clip process using a unit area of one hectare. This process results in a vector data area with daily landslide potential that contains information about the

- management area and the area with landslide potential.
- Calculate the statistical value of precipitation for each polygon with daily landslide potential. The calculated statistical values are the average value (mean), minimum value, maximum value, and mode value of precipitation in each polygon with daily landslide potential. To calculate these statistical values, we use the Zonal Statistics function.

The workflow of the module looks like the following in Figure 2-1.

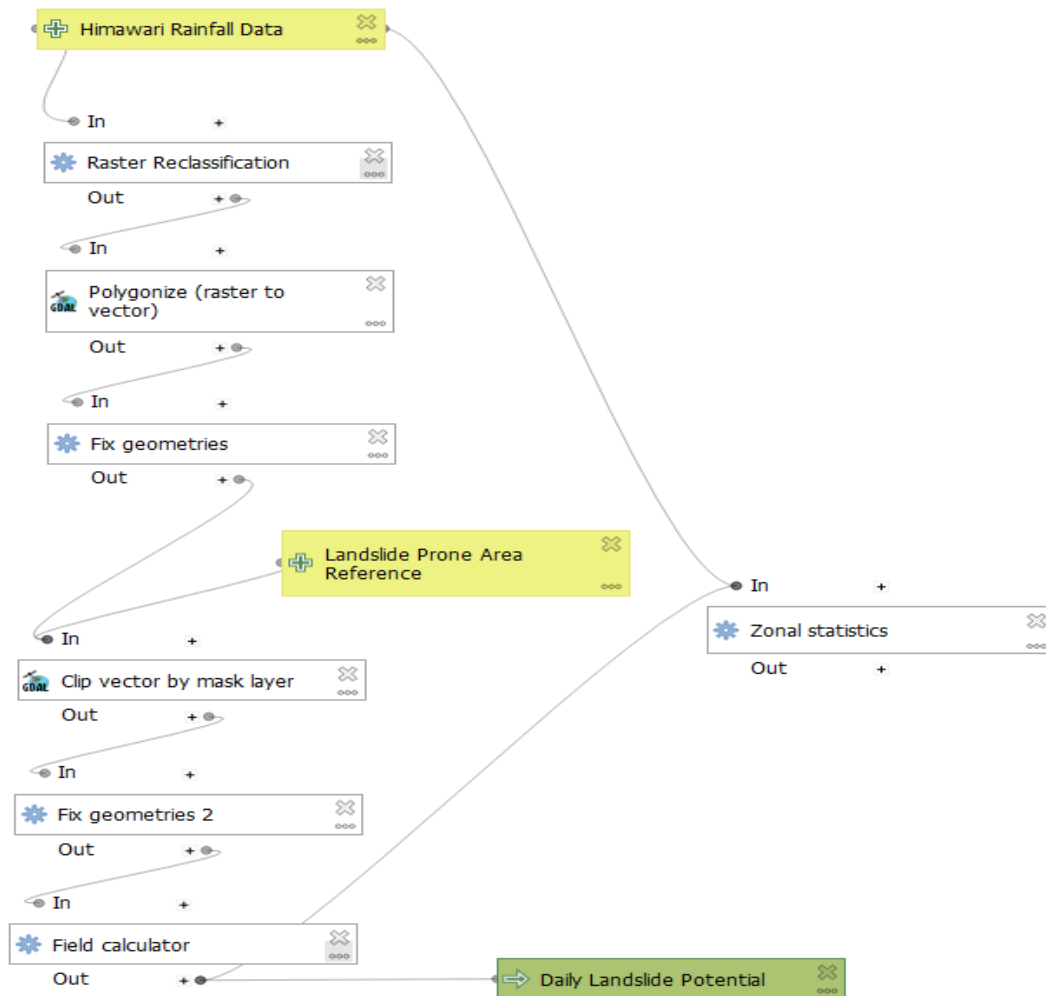


Figure 2-1. Workflow of daily landslide potential automation process

2.3. Automation Design

This program is an evolution of the previous program, namely the semi-automatic module using the QGIS Processing Modeler (Hadi 2013). This program is fully automatic and does not

require an operator since it uses the Python programming language and modules or libraries of the QGIS software (Sandhya 2020). The automation of daily information consists of 2 modules, namely a module for processing and a

module that automatically runs the processing program every day (Lee *et al.* 2020). The workflow of the program is shown in Figure 2-2.

This landslide prediction automation software runs in a virtual machine with the following specifications:

- Processor: 6 Core
- Memory: 8 GB
- Operating System: Fedora 31
- Software: QGIS and python

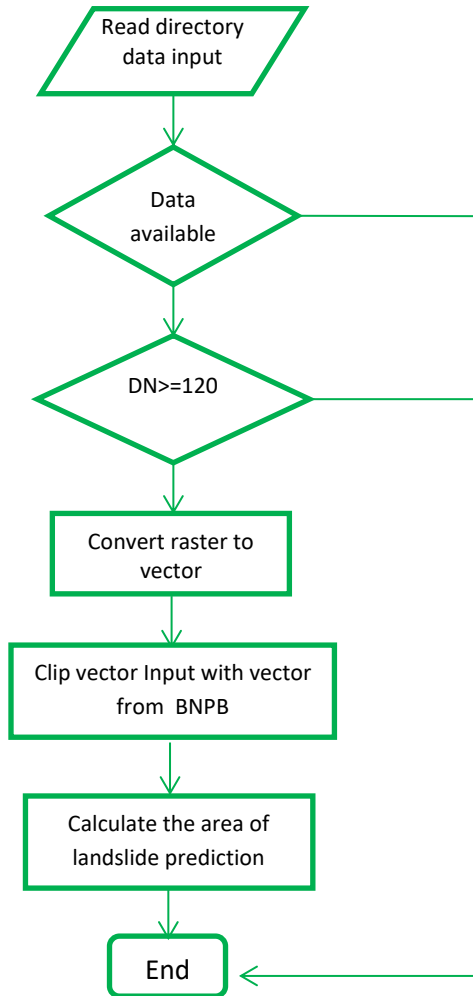


Figure 2-2. Automation processing flow chart

3 RESULTS AND DISCUSSION

3.1 Result

This software is run automatically using cron job software. This software can be run manually in the following ways:

Open the directory file, right click and choose "Open terminal."

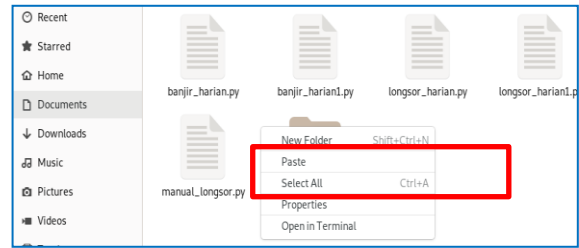


Figure 3-1. Source code file directory is located

A display like below

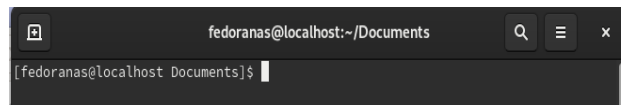


Figure 3-2. Terminal display

Type command "sudo python (name of source code)" then enter the root password

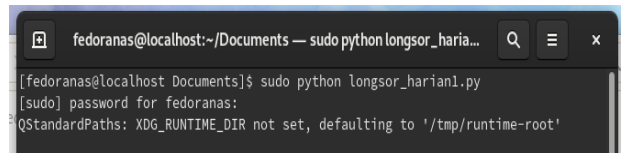


Figure 3-3. Command to execute source code

Check the output on the storage

Name	Date modified	Type	Size
Longsor_Harian.dbf	6/16/2020 9:15 PM	DBF File	51,472 KB
Longsor_Harian.prj	6/16/2020 9:14 PM	PRJ File	1 KB
Longsor_Harian.qpj	6/16/2020 9:14 PM	QPJ File	1 KB
Longsor_Harian.shp	6/16/2020 9:15 PM	SHP File	138,190 KB
Longsor_Harian.shx	6/16/2020 9:15 PM	SHX File	1,461 KB

Figure 3-4. The output files directory is located in the storage

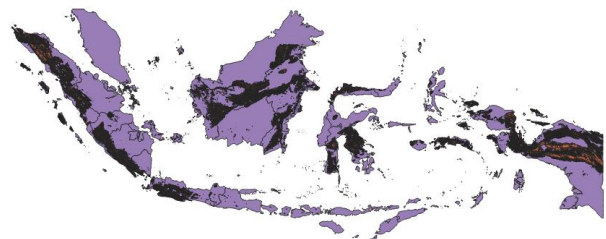


Figure 3-5. The output file, when opened with Qgis software and overlaid with a provincial map.

fid	OBJECTID	FID_ADMIN	KABUPATEN	PROVINSI	NEGARA	PULAU	Luas_Hekta
1	1	1	17 BANGKA SELAT...	BANGKA-BELIT...	INDONESIA	SUMATERA	2.00
2	2	2	17 BANGKA SELAT...	BANGKA-BELIT...	INDONESIA	SUMATERA	1.00
3	3	3	17 BANGKA SELAT...	BANGKA-BELIT...	INDONESIA	SUMATERA	9.02
4	4	4	17 BANGKA SELAT...	BANGKA-BELIT...	INDONESIA	SUMATERA	1.00
5	5	5	17 BANGKA SELAT...	BANGKA-BELIT...	INDONESIA	SUMATERA	2.00
6	6	6	17 BANGKA SELAT...	BANGKA-BELIT...	INDONESIA	SUMATERA	9.02
7	7	7	17 BANGKA SELAT...	BANGKA-BELIT...	INDONESIA	SUMATERA	4.01
8	8	8	17 BANGKA SELAT...	BANGKA-BELIT...	INDONESIA	SUMATERA	1.00
9	9	9	17 BANGKA SELAT...	BANGKA-BELIT...	INDONESIA	SUMATERA	14.03
10	10	10	17 BANGKA SELAT...	BANGKA-BELIT...	INDONESIA	SUMATERA	11.02
11	11	11	17 BANGKA SELAT...	BANGKA-BELIT...	INDONESIA	SUMATERA	61.12
12	12	12	17 BANGKA SELAT...	BANGKA-BELIT...	INDONESIA	SUMATERA	2.00
13	13	13	17 BANGKA SELAT...	BANGKA-BELIT...	INDONESIA	SUMATERA	4.01
14	14	14	17 BANGKA SELAT...	BANGKA-BELIT...	INDONESIA	SUMATERA	69.14
15	15	15	17 BANGKA SELAT...	BANGKA-BELIT...	INDONESIA	SUMATERA	1649.31
16	16	16	17 BANGKA SELAT...	BANGKA-BELIT...	INDONESIA	SUMATERA	61.10
17	17	17	17 BANGKA SELAT...	BANGKA-BELIT...	INDONESIA	SUMATERA	43.09
18	18	18	17 BANGKA SELAT...	BANGKA-BELIT...	INDONESIA	SUMATERA	2.00

Figure 3-6. Attributes of the output file when opened with Qgis software

Field fid and OBJECTID explain about polygon identity code. Field FID_ADMIN explain about admimistration area code. Field KABUPATEN explain name of district which polygon took place. Field PROVINSI explain name of province which polygon took place. Field NEGARA explain name of state which polygon took place. Field PULAU explain name of island which polygon took place. Field Luas_Hektar explain calculation area of polygon. Field CH_mean explain rainfall rate average inside polygon. CH_sum explain rainfall rate sum inside polygon.

To be able to run automatically, we use cron job software. Open the terminal and type command "sudo EDITOR=nano crontab -e"

```

fedoranas@localhost:~/Documents — sudo EDITOR=nano cronta...
[fedoranas@localhost Documents]$ sudo EDITOR=nano crontab -e
[sudo] password for fedoranas:

```

Figure 3-7. Terminal view and open cronjob

Type command for runing the source code every day at 10.00 A.M.

```

Activities Terminal Jun 25 09:55
fedoranas@localhost:~/Documents — sudo EDITOR=nano crontab -e
GNU nano 4.3 /tmp/crontab.rN53Ew Modified
10 * * * /usr/bin/python3 /home/fedoranas/documents/longsor_har'ian.py >> ~/cron.log 2>&1

```

Figure 3-8. Command for running the source code every day at 10.00 A.M.

3.2. Discussion

The automation system for landslide potential information has been successfully tested. It provides spatial information in the form of the distribution of areas with landslide potential based on daily precipitation accumulation data. The advantage of the developed system is that it was developed using free and open-source software, which allows users to implement this system without being restricted by commercial licenses. Moreover, this system can be used on PCs with different operating systems, from Windows to Linux. The source code of this system program is also open so that others can modify it. The processing time with this system is relatively fast and is about 10 minutes for a daily landslide potential information product. It is hoped that the short processing time can also speed up the dissemination of information to the public. Of course, this system is not one hundred percent perfect. There are still some weaknesses, such as the period in which precipitation accumulates 24 hours. The period could be shortened to accumulation every 3 or 6 hours to get more up-to-date information. In addition, the method can be improved in the future

by incorporating a precipitation forecast model so that predictions of possible landslides in the next day or two can also be made. Other challenges for the future include the use of higher-resolution satellite imagery and larger data sets. This requires computers that have larger and faster computing capacities.

On the other hand, programming algorithms must be improved to achieve more accurate and optimal results. The threshold used is the general threshold that occurs in Indonesia. The standard threshold that causes landslides as the primary trigger is high-intensity rainfall. Another parameter is used for other conditions that affect this result, namely the pre-existing landslide-prone zone. The results obtained will be better if other parameters are updated recently

An automation system has been built that quickly and directly transfers daily information on landslide potential to the built information system. However, whether this automation system is effective arises for daily landslide potential monitoring, it is still need improvement considering that landslides sometimes occur very quickly and at night. Many landslide events in which human lives were lost occurred at night. This is a challenge for the future to connect the built system with the existing sensor system in the field. This will be more effective when extreme rainfall occurs in an area and it is a potential landslide area, because then the alarm can be triggered, and the public can be warned early. People in landslide-prone areas can avoid them immediately.

4. Conclusion

The Daily Landslide Potential Information Automation System has been successfully developed using daily precipitation data from the Himawari 8 satellite. This system has succeeded in replacing manual processing, which takes longer processing time consumed. This system speeds up information processing compared to the previous system. This system can meet the current demand for automatically generated information.

5. Acknowledgment

We thank the members of the engineering team and the team for

preparing information in the dissemination department of PRPJ BRIN, who helped in the automation of daily information on landslide potential. We would also like to thank the management of PRPJ BRIN for full support of technical activities on automation of daily information on landslide potential in PRPJ BRIN..

References

- Susanti PD, Miardini A, (2019), Identifikasi Karakteristik dan Faktor Pengaruh pada Berbagai Tipe Longsor. *Agritech*, 39 (2): 97-107.
- Muzani, Setianingsi AI, Wahyuni SS, (2021), Analysis of The Cause of Landslide Disaster in Sukabumi, Indonesia (Using Analytical Hierarchy Process). *Journal of Southwest Jiaotong University*. 56(1)
- Tarantino C, Blonda P, Pasquariello G, (2007), Remote sensed data for automatic detection of land-use changes due to human activity in support to landslide Studies. *Nat Hazards* 41:245–267
- Adiningsih ES, Khomarudin, MR, (1998). Analisis pendugaan curah hujan dan kerawanan banjir dengan data satelit studi kasus Kota Semarang. *Majalah LAPAN* : 85.
- Hirose H, Shige S, Yamamoto MK, (2019). High Temporal Rainfall Estimations from Himawari-8 Multiband Observations Using the Random-Forest Machine-Learning Method. *Journal of the Meteorological Society of Japan*, 97(3): 689–710.
- Ayasha N, (2020), A Comparison of Rainfall Estimation Using Himawari-8 Satellite Data in Different Indonesian Topographies. *International Journal of Remote Sensing and Earth Science* Vol. 17 No. 2 December 2020: 189-200.
- Miura T, Nagai S, (2020), Landslide Detection with Himawari-8 Geostationary Satellite Data: A Case Study of a Torrential Rain Event in Kyushu, Japan. *Remote Sens.*, 12, 1734;
- Budiarti PW, Fathani TF, Faris F, (2020). Analysis of rainfall-triggered landslide in Baleagung Village,

- Magelang Regency, Central Java. E3S Web of Conferences 200, 01001
- Xu Q, Peng D, Zhang S, Zhu X, He C, Qi X, Zhao K, Xiu D, Ju N, (2020). Successful implementations of a real-time and intelligent early warning system for loess landslides on the Heifangtai terrace, China. *Engineering Geology* 278:105817
- Tengtrairat N, Woo WL, Parathai P, Aryupong C, Jitsangiam P, Ricchumphu D, (2021). Automated Landslide-Risk Prediction Using Web GIS and Machine Learning Models. *Sensors* 2021, 21, 4620.
- Ahmed B, Rahman MS, Islam R, Sammonds P, Zhou C, Uddin K, Hussaini TM, (2018). Developing a DynamicWeb-GIS Based Landslide Early Warning System for the Chittagong Metropolitan Area, Bangladesh. *ISPRS Int. J. Geo-Inf.* 2018, 7, 485
- Giri P, Ng K, Philips W, (2019). Wireless Sensor Network System for Landslide Monitoring and Warning. *IEEE Transactions on Instrumentation and Measurement*, Vol. 68, No. 4, April 2019
- Jeong S, Ko J, Ki, J, (2019). The Effectiveness of a Wireless Sensor Network System for Landslide Monitoring. *Digital Object Identifier* 10.1109/Access.2019
- Aggarwal S, Misra PK, Sumakar KVS, Chaturvedi P, (2018). Landslide Monitoring System Implementing IOT using video camera. 2018 3rd International Conference for Convergence in Technology (I2CT)
- Thirugnanam H, Uhlemann S, Reghunadh R, Ramesh MV, Rangan VP, (2022). Review of Landslide Monitoring Techniques With IoT Integration Opportunities. *IEEE Journal Of Selected Topics In Applied Earth Observations And Remote Sensing* Vol. 15
- Bachri S, Shrestha, RP, Yulianto F, Sumarmi S, Utomo KSB, Aldianto YE, (2021). Mapping Landform and Landslide Susceptibility Using Remote Sensing, GIS and Field Observation in the Southern Cross Road, Malang Regency, East Java, Indonesia. *Geosciences* 2021, 11, 4.
- Nurdin PF, Kubota T, (2018). GIS-Based Landslide Susceptibility Assessment and Factor Effect Analysis by Certainty Factor In Upstream of Jeneberang River, Indonesia. *Geoplanning: Journal of Geomatics and Planning* 5(1):75-90
- Lukman WL, Le CH, (2020). A Review on Remote Sensing and GIS Applications to Monitor Natural Disasters in Indonesia. *Korean Journal of Remote Sensing*, Vol.36, No.6-1:1303~1322
- PYQGIS developer Cookbook. <https://docs.qgis.org/3.4/pdf/en/QGIS-3.4-PyQGISDeveloperCookbook-en.pdf>
- Ujaval Gandhi, "Automating Complex Workflows using Processing Modeler", <http://google.com/+UjavalGandhi>. Translations by Maruli Tua Manullang, Bakhtiar Arif, (2012)
- Hadi F, (2013), "Modul Pelatihan QGIS versi Bahasa Indonesia", Penerbit ITB, Bandung,.
- Sandhya M C, (2020), "Exploring Opportunities with Open-Source GIS" *International Journal of Engineering Research & Technology (IJERT)*
- Lee H, Chae MS, Park JY, Lim KJ, Park YS, (2022) , "Development and Application of a QGIS-Based Model to Estimate Monthly Streamflow" *ISPRS Int. J. Geo-Inf.*

Appendix: Source code

Following script is source code for processing daily landslide potential. This programming script is written in Python programming language.

```
import os, sys
import datetime

from qgis.core import (
    QgsApplication,
```

```

QgsProcessing,
QgsProcessingMultiStepFeedback,
QgsProcessingParameterRasterLayer,
QgsProcessingParameterVectorLayer,
QgsProcessingParameterFeatureSink
)
from qgis.analysis import QgsNativeAlgorithms
os.environ["QT_QPA_PLATFORM"] = "offscreen"
#QgsApplication.setPrefixPath("/path/to/qgis/installation", True)
QgsApplication.setPrefixPath('/usr', True)
qgs = QgsApplication([], False)
qgs.initQgis()

sys.path.append('/usr/share/qgis/python/plugins')
from processing.core.Processing import Processing
Processing.initialize()
import processing
QgsApplication.processingRegistry().addProvider(QgsNativeAlgorithms())

#directory data input
in_dir = ('/media/share/AkumulasiRFR')
in_name = ('produkRFR2_ACC_HARIAN.tif')
out_dir = ('/media/BanjirLongsor/daily/landslide')
out_name = ('Longsor_Harian.shp')
today = datetime.datetime.today()
today = today - datetime.timedelta(days = 1)

date_dir = today.strftime('%Y-%m-%d')
thnbln_dir = today.strftime('%Y-%m')

if not os.path.isfile(os.path.join(*[out_dir,thnbln_dir,date_dir,out_name])):
    if not os.path.isdir(os.path.join(out_dir,thnbln_dir,date_dir)):
        if not os.path.isdir(os.path.join(out_dir,thnbln_dir)):
            os.mkdir(os.path.join(out_dir,thnbln_dir))
            os.mkdir(os.path.join(out_dir,thnbln_dir,date_dir))

ndir = os.path.join(*[in_dir,thnbln_dir,date_dir,in_name])
#reclassify by table
params_re = { 'DATA_TYPE': 0, 'INPUT_RASTER': ndir, 'NODATA_FOR_MISSING': True, 'NO_DATA': 0,
'RANGE_BOUNDARIES': 2, 'RASTER_BAND': 1, 'TABLE': [120,10000,1], 'OUTPUT':
QgsProcessing.TEMPORARY_OUTPUT }
ReklasifikasiRaster = processing.run('native:reclassifybytable', params_re, is_child_algorithm=True)

#vectorized
params_pol = {'BAND': 1, 'EIGHT_CONNECTEDNESS': True, 'EXTRA': "", 'FIELD': 'DN', 'INPUT':
ReklasifikasiRaster['OUTPUT'], 'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT }
PolygonizeRasterToVector = processing.run('gdal:polygonize', params_pol, is_child_algorithm=True)

#fix geometries
fg = ('/home/fedoranas/Documents/sup_long/fixgeo/fg2.gpkg')
params_fg1 = {'INPUT': PolygonizeRasterToVector['OUTPUT'], 'OUTPUT': fg}
FixGeometries1 = processing.run('native:fixgeometries', params_fg1, is_child_algorithm=True)

# Clip vector by mask layer
params_clip = { 'INPUT': '/home/fedoranas/Documents/sup_long/longsor/BNPB.gpkg', 'MASK':
FixGeometries1['OUTPUT'], 'OPTIONS': "", 'OUTPUT': QgsProcessing.TEMPORARY_OUTPUT }
ClipVectorByMaskLayer = processing.run('gdal:clipvectorbypolygon', params_clip, is_child_algorithm=True)

# Fix geometries 2
fg1 = ('/home/fedoranas/Documents/sup_long/fixgeo/fg3.gpkg')
params_fg2 = {'INPUT': ClipVectorByMaskLayer['OUTPUT'], 'OUTPUT': fg1}
FixGeometries2 = processing.run('native:fixgeometries', params_fg2, is_child_algorithm=True)

# Field calculator
out_lo = os.path.join(*[out_dir,thnbln_dir,date_dir,out_name])
params_calc = {'FIELD_LENGTH': 10, 'FIELD_NAME': 'Luas_Hektar', 'FIELD_PRECISION': 2, 'FIELD_TYPE': 0,
'FORMULA': '$area /10000', 'INPUT': FixGeometries2['OUTPUT'], 'NEW_FIELD': True, 'OUTPUT': out_lo}
FieldCalculator = processing.run('qgis:fieldcalculator', params_calc, is_child_algorithm=True)

```



```
if os.path.exists(fg):
    os.remove(fg)
if os.path.exists(fg1):
    os.remove(fg1)
if os.path.exists(ReklasifikasiRaster['OUTPUT']):
    os.remove(ReklasifikasiRaster['OUTPUT'])
if os.path.exists(PolygonizeRasterToVector['OUTPUT']):
    os.remove(PolygonizeRasterToVector['OUTPUT'])
if os.path.exists(ClipVectorByMaskLayer['OUTPUT']):
    os.remove(ClipVectorByMaskLayer['OUTPUT'])

qgs.exitQgis()
```