

DAFTAR PUSTAKA

- Aries, R. (2018). Kajian Akurasi Peta Ortofoto Dari Data Wahana Udara Tanpa Awak (WUTA). Yogyakarta: Departemen Teknik Geodesi Fakultas Teknik Universitas Gadjah Mada Yogyakarta.
- Aini Qurotul. dkk,(2021). Deteksi dan Pengenalan Objek dengan Model Machine Learning: Model YOLO. Diakses pada 11 Mei 2023. Journal of computer Engineering System and Science. Vol. 6 No. 2 Juli 2021. P-ISSN: 2502-7131. E_ISSN:2502714x.
- Budiharto, Widodo. (2018). *AI For Beginner*. Diakses pada 30 Januari 2023. Dari <https://socs.binus.ac.id/files/2018/06/AI-for-Beginner-Prof.-Dr.-Widodo-Budiharto.pdf>
- Daniel, Akhila & Preeja, V (2014). *A Novel Technique for Automatic Road Distress Detection and Analysis*. Diakses pada 29 Januari 2023. Dari <https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=e67c04cccd668400788bd21ab1ca53c8c6f4922b>
- DPU Direktorat Jenderal Bina Marga. (1997). Tata Cara Perencanaan Geometrik Jalan Antar Kota. Diakses pada 30 Januari 2023. Dari <https://spada.uns.ac.id/mod/resource/view.php?id=94492>
- Fajar, Andriyana. dkk. (2022). Identifikasi Kerusakan Jalan dengan Metode *Faster RCNN* Studi Kasus di Jalan Pakansari Bogor Jawa barat. Diakses pada 30 Januari 2023. Dari https://ejournal.poltektegal.ac.id/index.php/smartcomp/article/download/3509/pdf_107.
- Hakim Muhammad Azizul, Heni Emawati & Dwi Ery Mujahiddin. (2021). Pemanfaatan Pesawat Udara Tanpa Awak Untuk Pemetaan dan Identifikasi Penutupan Lahan Pada Kawasan Hutan Pendidikan Unmul. Diakses pada 18 Maret 2023. Jurnal AGRIFOR Volume XX Nomor 1, Maret 2021. ISSN P: 1412-6885. ISSN O: 2503-4960.
- Hidayatulloh, Muhamad Syarif. (2021). Sistem Pengenalan Wajah Menggunakan Metode YOLO (*You Only Look Once*). Diakses pada 1 Februari 2023. Dari

<https://repository.dinamika.ac.id/id/eprint/5568/2/15410200028-2021-UNIVERSITADINAMIKA.pdf>

Khatami Muhammad Sauqi, (2022). Deteksi Kendaraan Menggunakan Algoritma You Only Look Once (YOLO) V3. Diakses pada 25 Juni 2023.

Kementrian PUPR Direktorat Jenderal Bina Marga. (2017). Tentang Penyampaian Manual Desain Perkerasan Jalan Revisi 2017 di Lingkungan Direktorat Jenderal Bina Marga. Diakses pada 30 Januari 2018. Dari [https://binamarga.pu.go.id/v3/assets/files/NSPK/pembangunan_jalan/2017_SE%20DIRJEN%20MANUAL%20DESAIN%20PERKERASAN%20JALAN%20\(REVISI%20%202017\)%20\(STEMPEL\)%20FINAL.pdf](https://binamarga.pu.go.id/v3/assets/files/NSPK/pembangunan_jalan/2017_SE%20DIRJEN%20MANUAL%20DESAIN%20PERKERASAN%20JALAN%20(REVISI%20%202017)%20(STEMPEL)%20FINAL.pdf)

Lapian Andre Kanisius Edguard, Sherwin R.U.A Sompie & Pinrolinvic D. K. Manembu, (2021). *You Only Look Once (YOLO) Implementation for Signatura Pattern Clasicication*. Jurnal Teknik Informatika Vol. 16 no. 3 July-September 2021. Pp, 337-346. P-ISSN: 2301-8364. E-ISSN : 2685-6131. <https://ejournal.unsrat.ac.id/index.php/informatika>

M. Frandi B Simamora, Bandi Sasmito & Hani'ah, (2015). Kajian Metode Segmentasi untuk Identifikasi Tutupan Lahan dan Luas Bidang Tanah menggunakan Citra pada Google Earth. Jurnal Geodesi Undip, Oktober 2015.

Mandaya Iradaf & Harintaka, (2020). Pemanfaatan Teknologi UAV (*Unmanned Aerial Vehicle*) untuk identifikasi dan klasifikasi jenis-jenis kerusakan jalan. FT-UGM, Departemen Teknik Geodesi. Magister Teknik Geomatika.

Manajang Dufan J.P., Sherwin R.U.A. Sompie & Agustinus Jacobus. (2020). Implementasi Framework Tensorflow Objek Detection dalam Mengklasifikasi Jenis Kendaraan Bermotor. Jurnal Teknik Informatika Vol. 15 no.3 Juli – September 2020. Hal, 171-178. P-ISSN : 2301-8364. e-ISSN : 2685-6131. Tersedia pada: <https://ejournal.unsrat.ac.id/index.php/informatika/>

Putra, Jan Wira Gotama. (2020). Pengenalan Konsep Pembelajaran Mesin dan *Deep Learning*. Diakses pada 30 Januari 2023. Dari <https://wiragotama.github.io/resources/ebook/intro-to-ml-secured.pdf>

Prahasta, E. 2009. Sistem Informasi Geografis : Konsep-konsep dasar Sistem Informasi Geografis. Informatika : Bandung.

- Pramestya, R. H. (2018). Tentang Deteksi dan Klasifikasi Kerusakan Jalan Aspal Menggunakan Metode YOLO Berbasis Citra Digital. Diakses pada 28 Januari 2023. Dari https://repository.its.ac.id/59044/1/06111650010019-Master_Thesis.pdf
- Rahmat, Pupu Saeful. (2009). Penelitian Kualitatif. Diakses pada 1 Januari 2023. Dari <http://yusuf.staff.ub.ac.id/files/2012/11/Jurnal-Penelitian-Kualitatif.pdf>
- RPJMD-Kabupaten-Berau. (2021). Tentang Rencana Pembangunan Jangka Menengah Daerah Kabupaten Berau. Diakses pada 28 Januari 2023. Dari <http://beraukab.go.id/v2/wp-content/uploads/2021/11/RPJMD-2021-2026-Dengan-Perda-1.pdf>
- Redmon, Joseph. dkk. (2016). Tentang YOLO, *Realtime Objek Detection*. Diakses pada 28 Januari 2023. Dari https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf
- Ryu, Ki-Seung, Taehyeon Kim & Young-Ro Kim. (2015). *Image-Based Pothole Detection System for ITS Service and Road Management System*. Diakses pada 29 Januari 2023. Dari https://www.researchgate.net/publication/283188383_Image-Based_Pothole_Detection_System_for ITS Service and Road Management_System
- Salsabila, R. (2017). Perbandingan Perhitungan Volume Stockpile Batu Bara Menggunakan Data Terrestrial Laser Scanner (TLS) dan Data Foto Udara Unmanned Aerial Vehicle (UAV). Program Studi Sarjana Teknik Geodesi Departemen Teknik Geodesi Fakultas Teknik Universitas Gadjah Mada Yogyakarta.
- Syauqani, A., Subiyanto, S., & Suprayogi, A. (2017). Pengaruh Variasi Tinggi Terbang Menggunakan Wahana Unmanned Aerial Vehicle (UAV) Quadcopter DJI Phantom 3 Pro Pada Pembuatan Peta Orthofoto. Jurnal Geodesi Undip, 6(1).
- Schneiderman, Henry, Kanade, T. (2000) „A statistical method for 3D object detection applied to faces and cars“, IEEE Computer Society Conference on

- Computer Vision and Pattern Recognition, (1), pp. 746–751. doi: 10.1109/CVPR.2000.855895.
- Subakti, B. (2017). Pemanfaatan Foto Udara UAV Untuk Pemodelan Bangunan 3D Dengan Metode Otomatis. ITN Malang, 15(30), 15 - 30.
- Sukwika, Tatan. (2018). Peran Pembangunan Infrastruktur Terhadap Ketimpangan Ekonomi Antarwilayah Indonesia. Diakses pada 28 Januari 2023. Dari <https://core.ac.uk/download/pdf/234033183.pdf>
- Triasanti, D. (2000) „Konsep dasar python“, AP2B, pp. 1–6.
- Tiwari, N. (2020). YOLOv3 Custom Object Detection with Transfer Learning. TFUG Mumbai Weekly. <https://tiwarinitin1999.medium.com/yolov3-custom-object-detection-withtransfer-learning-47186c8f166d>
- Triardhana, Y. Bandi S. Firman H. (2021). Identifikasi Kerusakan Jalan Menggunakan Metode *Deep Learning* (DL) Model *Convolutional Neural Network* (CNN). Diakses pada 30 Januari 2023. Dari <https://ejournal3.undip.ac.id/index.php/geodesi/article/download/31159/25697>
- Utama, R. M & Ida F. (2016). Tentang Evaluasi Kondisi Struktural Pada Jalan Berdasarkan Hubungan Antara Ketidakrataan Permukaan Jalan (IRI) dan Indeks Kondisi Jalan (RCI). Diakses pada 28 Januari 2023. Dari <https://jurnal.itg.ac.id/index.php/konstruksi/article/view/386>
- Undang-Undang No.2. (2022). Tentang Perubahan Kedua Atas Undang-Undang Nomor 38 Tahun 2004 Tentang Jalan. Diakses pada 28 Januari 2023. Dari <https://jdih.pu.go.id/internal/assets/assets/produk/UU/2022/01/UU2.pdf>
- Zulkhaidi Tengku Cut Al-Saidina, Eny Marya & Yulianto. (2019). Pengenalan Pola Bentuk Wajah dengan OpenCV. JURTI, Vol.3 No.2, Desember 2019, ISSN: 2579-8790.

LAMPIRAN

Lampiran 1 Dokumentasi persiapan sebelum penerbangan (pemasangan Propeller, baterai, SD card, dll)



**POLITEKNIK SINAR MAS
BERAU COAL**

Lampiran 2 Dokumentasi pengambilan data foto udara



Lampiran 3 Script training dataset

```
[ ] from google.colab import drive
    drive.mount('/content/drive')

    Mounted at /content/drive

[ ] %cd /content/drive/MyDrive/Training_jalan/

    /content/drive/MyDrive/Training_jalan

[ ] %ls /content/drive/MyDrive/Training_jalan/

    backup/          data_training/      Training_Jalan_ilham.ipynb
    darknet/         data_training2/     Training_Jalan.ipynb
    data_testing/    framework_person.ipynb  YOLOV4_baru.ipynb
    data_testing2/   Parameter/

[ ] # clone darknet repo
    !git clone https://github.com/AlexeyAB/darknet
```

```
[ ] # change makefile to have GPU and OPENCV enabled
%cd /content/drive/MyDrive/Training_jalan/darknet
lsed -i 's/OPENCV=0/OPENCV=1/' Makefile
lsed -i 's/GPU=0/GPU=1/' Makefile
lsed -i 's/CUDNN=0/CUDNN=1/' Makefile
lsed -i 's/CUDNN_HALF=0/CUDNN_HALF=1/' Makefile
```

```
/content/drive/MyDrive/Training_jalan/darknet
```

```
[ ] # verify CUDA
!usr/local/cuda/bin/nvcc --version
```

```
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2022 NVIDIA Corporation
Built on Wed_Sep_21_10:33:58_PDT_2022
Cuda compilation tools, release 11.8, V11.8.89
Build cuda_11.8.r11.8/compiler.31833905_0
```

```
[ ] # upload the custom .cfg back to cloud VM from Google Drive
!cp '/content/drive/MyDrive/Training_jalan/Parameter/yolov4_baru.cfg' '/content/drive/MyDrive/Training_jalan/darknet/cfg'
```

```
[ ] # upload the custom .cfg back to cloud VM from Google Drive
!cp '/content/drive/MyDrive/Training_jalan/Parameter/yolov4_baru.cfg' '/content/drive/MyDrive/Training_jalan/darknet/cfg'
```

```
[ ] # mindahin folder butuh parameter r
!cp '/content/drive/MyDrive/Training_jalan/data_training2/' -r '/content/drive/MyDrive/Training_jalan/darknet/data/'
!cp '/content/drive/MyDrive/Training_jalan/data_testing2/' -r '/content/drive/MyDrive/Training_jalan/darknet/data/'
```

```
[ ] # transforms = ['Identity', 'AutoContrast', 'Equalize', 'Rotate', 'Solarize', 'Color', 'Posterize', 'Contrast', 'Brightness', 'Sharpness', 'ShearX', 'ShearY', 'TranslateX', 'TranslateY']
# def randaugment(N, M):
#     """Generate a set of distortions.
#     # Args:
#     # N: Number of augmentation transformations to
#     # apply sequentially.
#     # M: Magnitude for all the transformations.
#     # """
#     sampled_ops = np.random.choice(transforms, M)
#     return [(op, M) for op in sampled_ops]
```

```
[ ] # !unzip '/content/drive/MyDrive/Persiapan Data Set Mining Eyes Analytics/DATA_SETELAH_DI_CEK/6.kelas/Training.zip' -d '/content/darknet/data/'
# !unzip '/content/drive/MyDrive/Persiapan Data Set Mining Eyes Analytics/DATA_SETELAH_DI_CEK/6.kelas/Testings.zip' -d '/content/darknet/data/'
```

```
[ ] # upload the obj.names and obj.data files to cloud VM from Google Drive
!cp '/content/drive/MyDrive/Training_jalan/Parameter/obj_jalandrone.names' '/content/drive/MyDrive/Training_jalan/darknet/data/'
!cp '/content/drive/MyDrive/Training_jalan/Parameter/obj.data' '/content/drive/MyDrive/Training_jalan/darknet/data/'
```

```
• # train your custom detector! (uncomment %%capture below if you run into memory issues or your Colab is crashing)
# %%capture
!./darknet detector train '/content/drive/MyDrive/Training_jalan/Parameter/obj.data' '/content/drive/MyDrive/Training_jalan/Parameter/yolov4_baru.cfg' yolov4.conv.137 -dont_show
```

```
Streaming output truncated to the last 5000 lines:
```

```
Loaded: 0.000077 seconds
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 139 Avg (IOU: 0.539827), count: 6, class_loss = 2.569276, iou_loss = 52.316711, total_loss = 54.885986
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 150 Avg (IOU: 0.706014), count: 6, class_loss = 0.634306, iou_loss = 10.722702, total_loss = 11.357009
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.926910), count: 2, class_loss = 0.010609, iou_loss = 0.837003, total_loss = 0.847612
total bbox = 2390258, rewritten_bbox = 0.272272 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 139 Avg (IOU: 0.693822), count: 20, class_loss = 5.122112, iou_loss = 245.848373, total_loss = 250.970490
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 150 Avg (IOU: 0.702432), count: 21, class_loss = 3.357688, iou_loss = 81.325424, total_loss = 84.683113
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.000000), count: 1, class_loss = 0.514184, iou_loss = 0.000000, total_loss = 0.514184
total bbox = 2390299, rewritten_bbox = 0.272265 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 139 Avg (IOU: 0.664010), count: 14, class_loss = 6.069415, iou_loss = 199.041916, total_loss = 205.111328
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 150 Avg (IOU: 0.782152), count: 7, class_loss = 1.200287, iou_loss = 10.621593, total_loss = 11.821879
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.724190), count: 2, class_loss = 0.001536, iou_loss = 1.355677, total_loss = 1.357214
total bbox = 2390322, rewritten_bbox = 0.272265 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 139 Avg (IOU: 0.708759), count: 8, class_loss = 3.892181, iou_loss = 100.688705, total_loss = 104.580887
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 150 Avg (IOU: 0.795497), count: 6, class_loss = 1.563287, iou_loss = 22.479736, total_loss = 24.043024
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.768675), count: 1, class_loss = 0.021152, iou_loss = 0.356996, total_loss = 0.378148
total bbox = 2390337, rewritten_bbox = 0.272263 %
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 139 Avg (IOU: 0.611619), count: 42, class_loss = 17.261497, iou_loss = 498.498566, total_loss = 515.760071
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 150 Avg (IOU: 0.695872), count: 30, class_loss = 13.593451, iou_loss = 51.914024, total_loss = 65.507477
v3 (iou loss, Normalizer: (iou: 0.07, obj: 1.00, cls: 1.00) Region 161 Avg (IOU: 0.674695), count: 5, class_loss = 1.720856, iou_loss = 1.714572, total_loss = 3.435428
total bbox = 2390414, rewritten_bbox = 0.272254 %
```

Lampiran 4 *Script* model deteksi

```

▶ from google.colab import drive
drive.mount('/content/drive')

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True)

▶ # import the necessary packages
from scipy.spatial import distance as dist
import numpy as np
import argparse
import imutils
import cv2
import os

[ ] # construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--input", type=str, default="",
    help="path to (optional) input video file")
ap.add_argument("-o", "--output", type=str, default="",
    help="path to (optional) output video file")
ap.add_argument("-d", "--display", type=int, default=1,
    help="whether or not output frame should be displayed")

args = vars(ap.parse_args(["--input", "", "--output", "", "--display", "1"]))

[ ] weightsPath = "/content/drive/MyDrive/Training_jalan/backup/yolov4_jalandrone3_best.weights"
configPath = "/content/drive/MyDrive/Training_jalan/Parameter/yolov4_jalandrone3.cfg"

net = cv2.dnn.readNetFromDarknet(configPath, weightsPath)

[ ] # load the COCO class labels our YOLO model was trained on
labelsPath = "/content/drive/MyDrive/Training_jalan/Parameter/obj_jalandrone.names"
LABELS = open(labelsPath).read().strip().split("\n") # \n beda baris di jadikan 1 list
print(LABELS)

['Potholes', 'Structural_crack', 'Alligator_crack', 'Longitudinal_crack']

[ ] MIN_CONF = 0.1
NMS_THRESH = 0.2

[ ] def detect_Potholes(frame, net, ln, Idx=0):
    (H, W) = frame.shape[:2]
    results = []

    # construct a blob from the input frame and then perform a forward
    blob = cv2.dnn.blobFromImage(frame, 1 / 255.0, (480, 480),
        swapRB=True, crop=False)
    net.setInput(blob)
    layerOutputs = net.forward(ln)

    # initialize our lists of detected bounding boxes, centroids, and
    boxes = []
    centroids = []
    confidences = []

    # loop over each of the layer outputs
    for output in layerOutputs:
        # loop over each of the detections
        for detection in output:
            scores = detection[5:]
            classID = np.argmax(scores)
            confidence = scores[classID]

            if classID == Idx and confidence > MIN_CONF:
                box = detection[0:4] * np.array([W, H, W, H])
                (centerX, centerY, width, height) = box.astype("int")

                # use the center (x, y) coordinates to derive the top

```

```
[ ] # construct the argument parse and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "--input", type=str, default="",
    help="path to (optional) input video file")
ap.add_argument("-o", "--output", type=str, default="",
    help="path to (optional) output video file")
ap.add_argument("-d", "--display", type=int, default=1,
    help="whether or not output frame should be displayed")

args = vars(ap.parse_args(["--input", "", "--output", "", "--display", "1"]))
```

```
▶ # determine only the *output* layer names that we need from YOLO
ln = net.getLayerNames()
ln = [ln[i - 1] for i in net.getUnconnectedOutLayers()]
```

```
[ ] path = '/content/drive/MyDrive/Training_jalan/Alligator_Crack/'
path_out = '/content/drive/MyDrive/Training_jalan/output_tes/'
for gambar in os.listdir(path):
    print(gambar)

    img = cv2.imread(path + gambar)
    height, width, _ = img.shape

    results_Potholes = detect_Potholes(img, net, ln, Idx=LABELS.index("Potholes"))
    results_Structural_Crack = detect_Structural_Crack(img, net, ln, Idx=LABELS.index("Structural_crack"))
    results_Alligator_Crack = detect_Alligator_Crack(img, net, ln, Idx=LABELS.index("Alligator_crack"))
    results_Longitudinal_Crack = detect_Longitudinal_Crack(img, net, ln, Idx=LABELS.index("Longitudinal_crack"))

    for (j, (prob, bbox, centroid)) in enumerate(results_Potholes):
        # extract the bounding box and centroid coordinates, then
        # initialize the color of the annotation
        (startX, startY, endX, endY) = bbox
        (cX, cY) = centroid
        color = (2, 36, 241)
        cv2.rectangle(img, (startX, startY), (endX, endY), color, 5)
        cv2.putText(img, 'Potholes', (startX, startY-10), cv2.FONT_HERSHEY_SIMPLEX, 2, (2, 36, 241), 2)

    for (j, (prob, bbox, centroid)) in enumerate(results_Structural_Crack):
        # extract the bounding box and centroid coordinates, then
        # initialize the color of the annotation
        (startX, startY, endX, endY) = bbox
        (cX, cY) = centroid
```

```
▶ for (j, (prob, bbox, centroid)) in enumerate(results_Longitudinal_Crack):
    # extract the bounding box and centroid coordinates, then
    # initialize the color of the annotation
    (startX, startY, endX, endY) = bbox
    (cX, cY) = centroid
    color = (7, 241, 13)
    cv2.rectangle(img, (startX, startY), (endX, endY), color, 5)
    # cv2.putText(img, 'Longitudinal crack', (startX, startY-10), cv2.FONT_HERSHEY_SIMPLEX, 0.2, (7, 241, 13), 1)

    # save hasilnya
    cv2.imwrite(path_out + 'output_' + gambar, img)
```

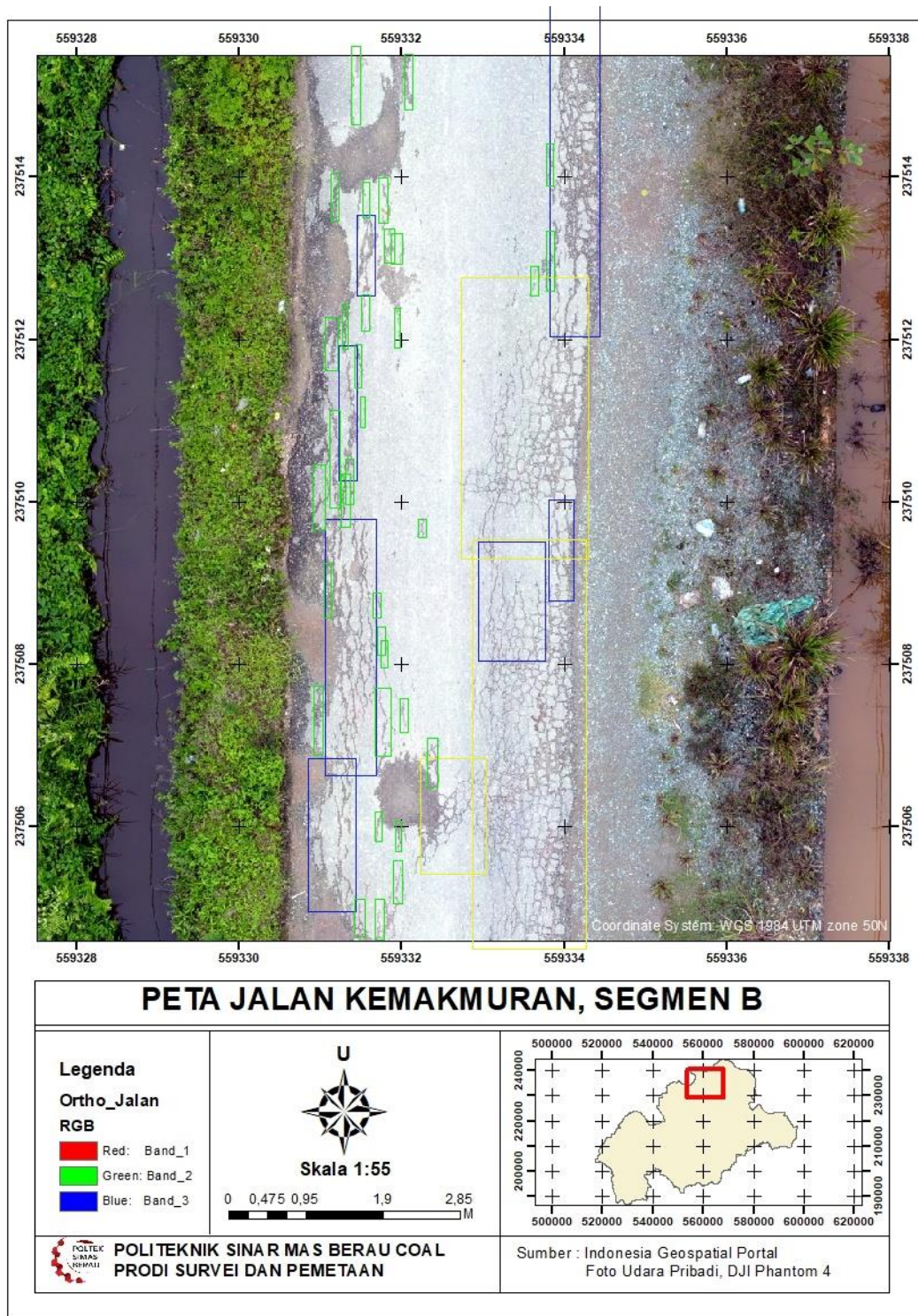
DJI_0189.JPG
 DJI_0188.JPG
 DJI_0191.JPG
 DJI_0190.JPG
 DJI_0193.JPG
 DJI_0192.JPG
 DJI_0194.JPG
 DJI_0195.JPG
 DJI_0050.JPG
 DJI_0049.JPG
 DJI_0053.JPG
 DJI_0052.JPG
 DJI_0051.JPG
 DJI_0075.JPG
 DJI_0074.JPG
 DJI_0073.JPG

Lampiran 5 Peta kerusakan Jalan yang telah teridentifikasi atau terklasifikasi dengan menggunakan model YOLO v4.

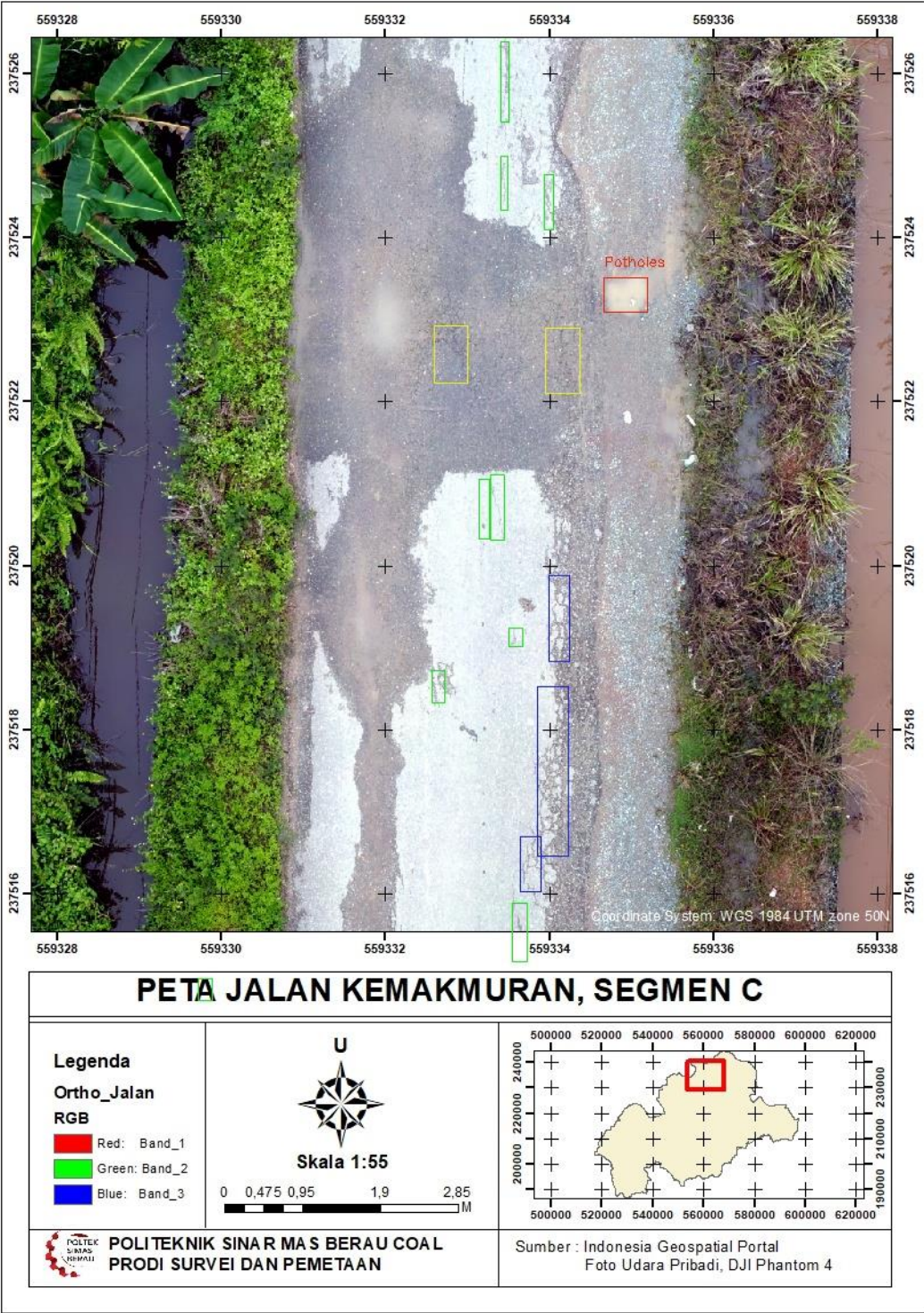
1. Peta kerusakan jalan segmen A



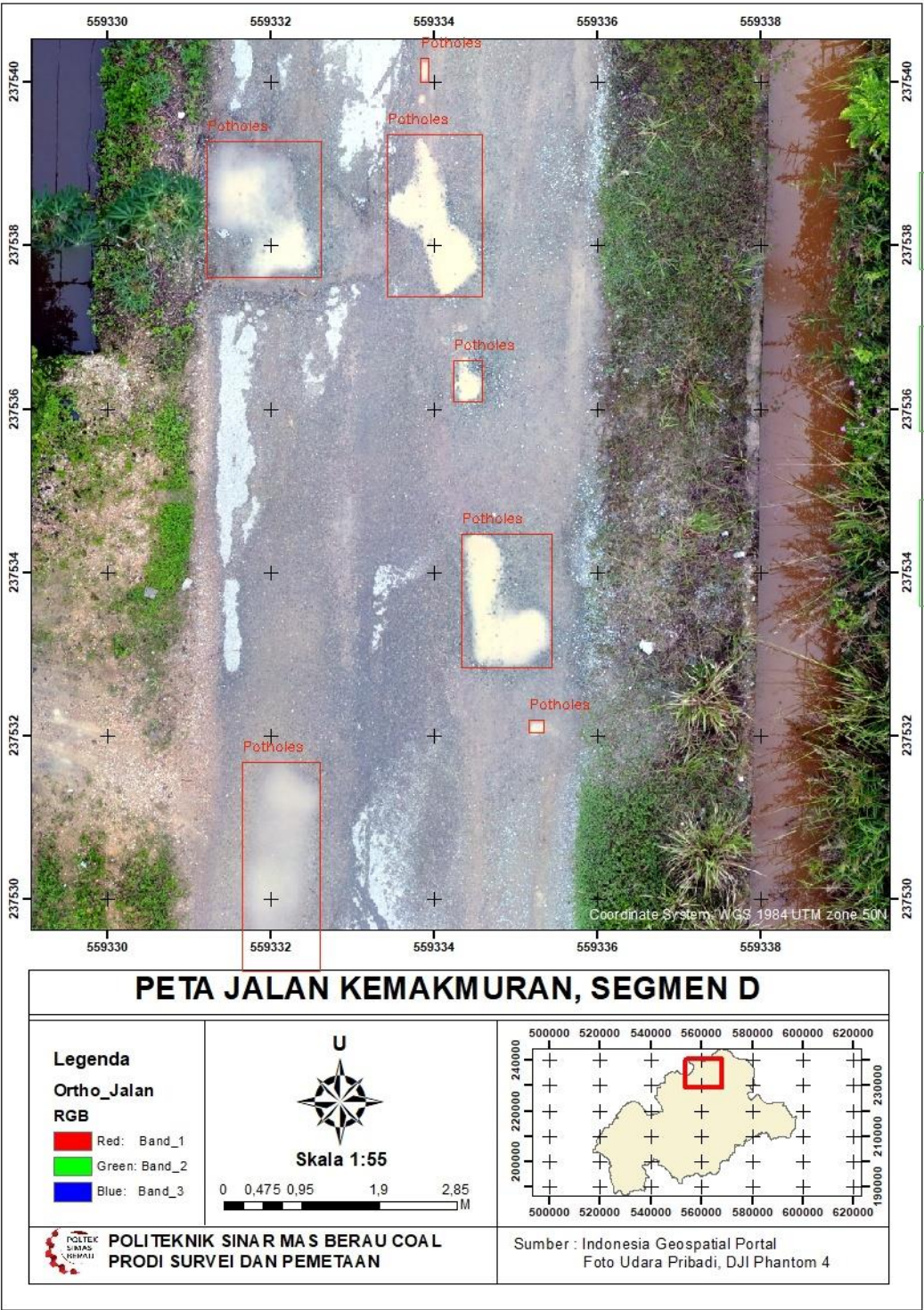
2. Peta kerusakan jalan segmen B



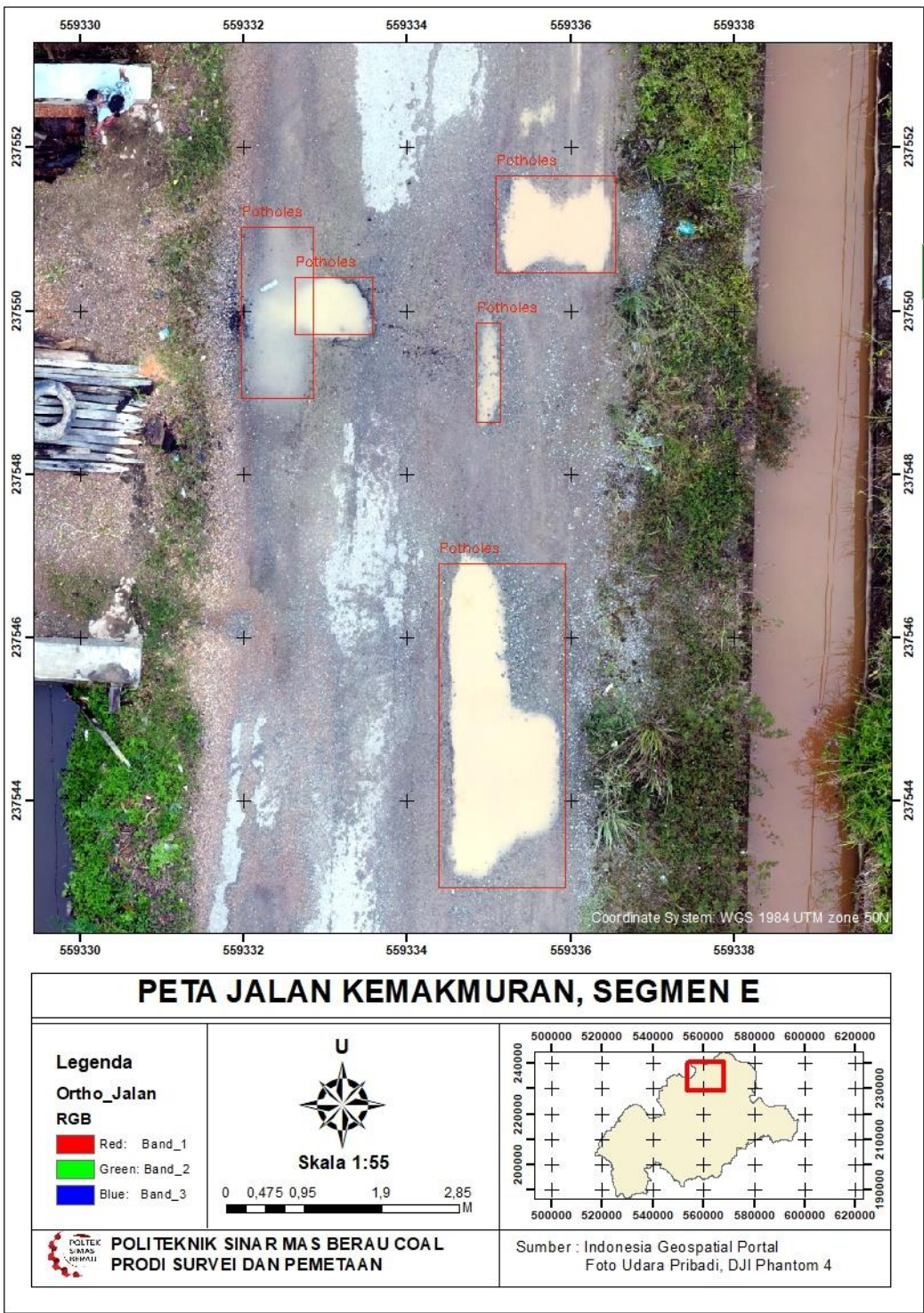
3. Peta kerusakan jalan segmen C



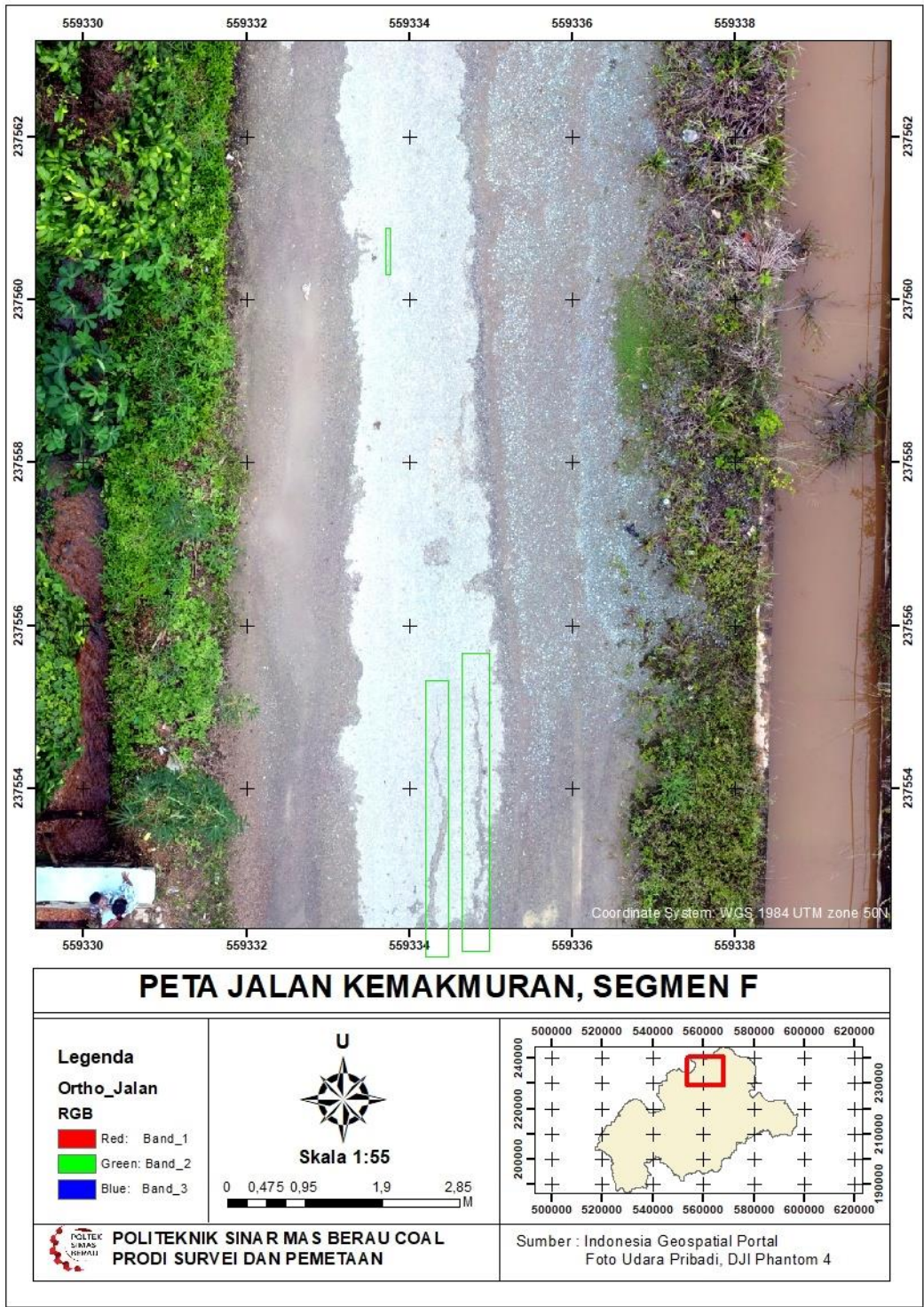
4. Peta kerusakan jalan segmen D



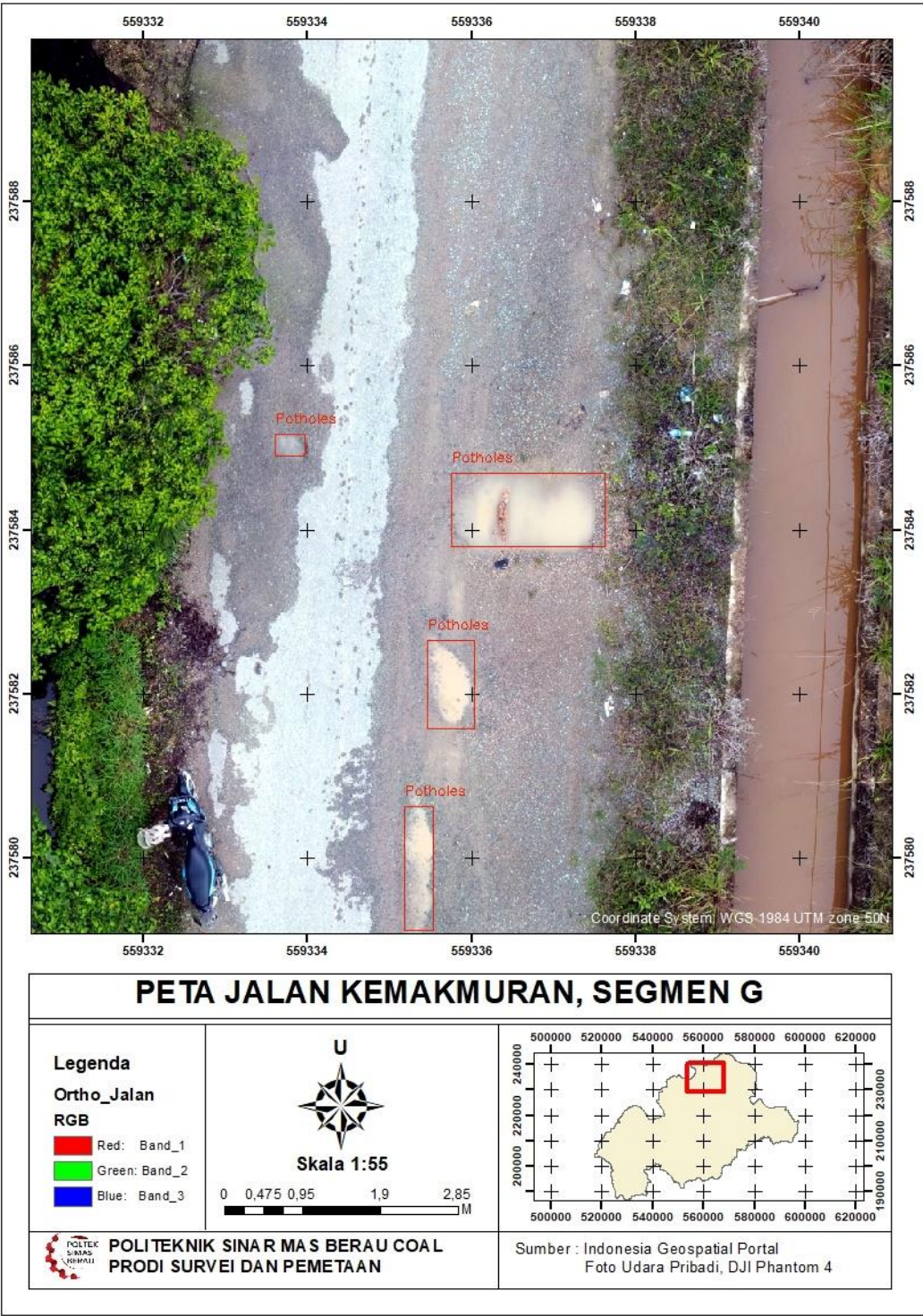
5. Peta kerusakan jalan segmen E



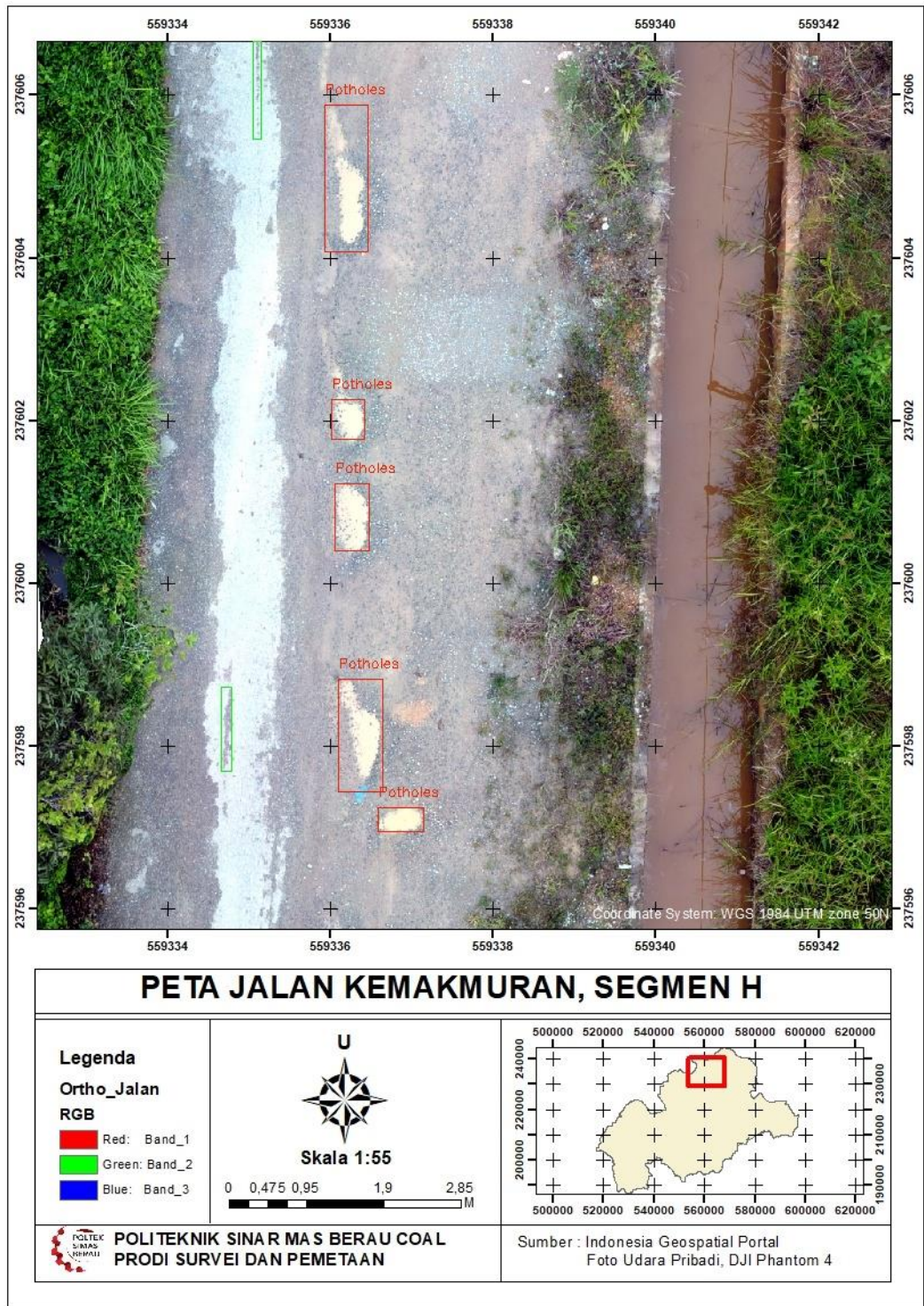
6. Peta kerusakan jalan segmen F



7. Peta kerusakan jalan segmen G



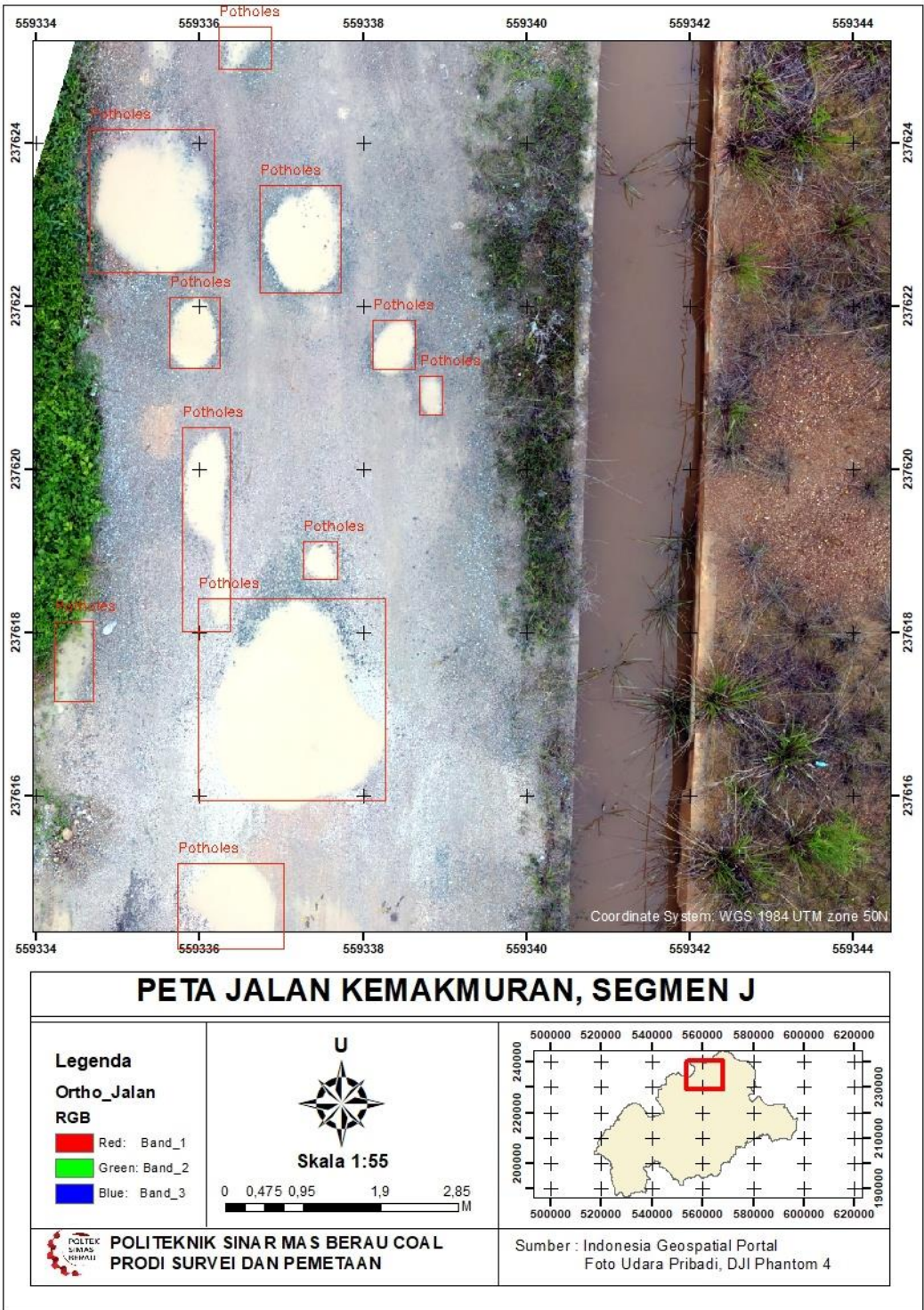
8. Peta kerusakan jalan segmen H



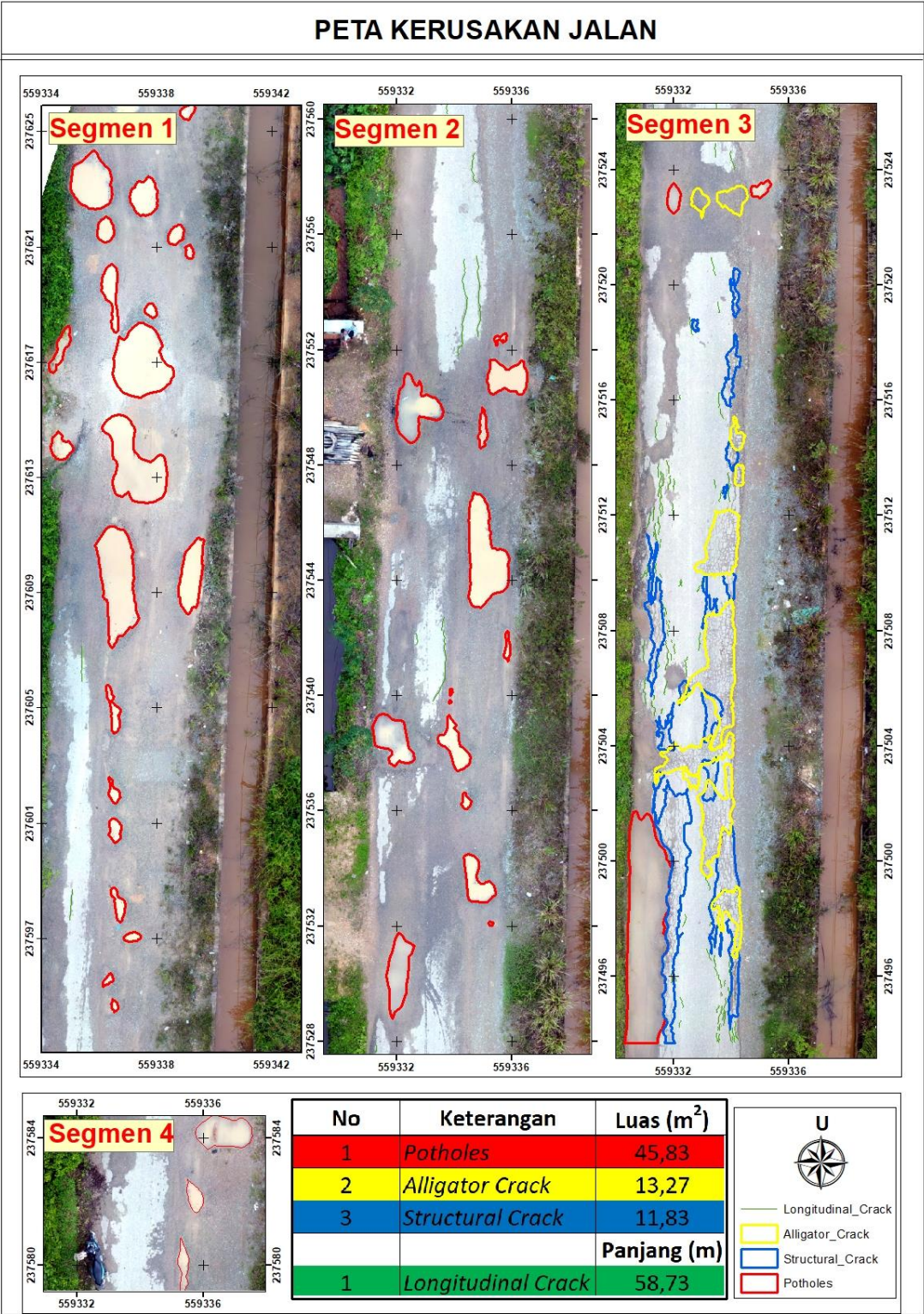
9. Peta kerusakan jalan segmen I



10. Peta kerusakan jalan segmen J



Lampiran 6 Peta Luas Keseluruhan Objek Kerusakan jalan





SURAT KETERANGAN

Dengan ini diterangkan bahwa TA/Skripsi yang ditulis oleh:

Nama : Sigit Tio Legowo
NIM : 20302001
Prodi : Survey dan Pemetaan
Judul : Implementasi Algoritma YOLO (*You Only Look Once*) untuk
Deteksi dan Klasifikasi Kerusakan Jalan Aspal Berbasis Citra UAV
(*Unmanned Aerial Vehicle*) di Sebagian Jalan Kemakmuran
Kecamatan Sambaliung

Berdasarkan hasil *scanning* menunjukkan tingkat plagiasi sebesar 24% dan
memenuhi syarat untuk tahapan selanjutnya dengan ketentuan yang berlaku.

Demikian surat ini dibuat untuk dipergunakan sebagaimana mestinya.

Berau, 14 Agustus 2023

Staf Perpustakaan,



BIODATA PENULIS



Penulis bernama lengkap Sigit Tio Legowo, dilahirkan di Purworejo, 12 Desember 2001. Penulis merupakan anak ke lima dari 6 bersaudara dari Bapak Margono dan Ibu Sri Purwandiyah. Penulis telah menempuh pendidikan formal di SD Negeri 028 Kecamatan Sambaliung, SMP Negeri 30 Berau dan SMA Negeri 12 Berau Kecamatan Sambaliung dan tamat pada tahun 2020. Dan pada tahun yang sama penulis terdaftar sebagai Mahasiswa di Politeknik Sinar Mas Berau Coal Program Studi Diploma Tiga Survei dan Pemetaan dengan NIM 20302001. Selama proses kegiatan perkuliahan penulis telah melaksanakan kegiatan KPI (Kuliah Praktik Industri) di PT. Berau Coal Site Binungan *Mine Operation 2* (BMO) pada tahun 2022. Kemudian penulis melaksanakan kegiatan PRAKERIN (Praktik Kerja Industri) 1 selama 3 bulan di PT. Berau Coal Departemen *Minetech* dalam *project Mining Eyes Analytic* (MEA) dan melanjutkan PRAKERIN 2 di PT. Berau Coal Site Gurimbang *Mine Operation* (GMO) pada tahun 2023. Penulis juga aktif mengikuti *project* selama kegiatan perkuliahan. Adapun *project* yang pernah diikuti adalah pengukuran Topografi untuk perencanaan pembangunan *McDonald's* di Tarakan pada tahun 2023 dan pengukuran Topografi untuk perencanaan perkerasan jalan di Kampung Segah pada tahun 2023. Pada akhir perkuliahan, penulis menyusun tugas akhir dengan judul “Implementasi Algoritma YOLO (*You Only Look Once*) Untuk Deteksi dan Klasifikasi Kerusakan Jalan Aspal Berbasis Citra UAV (*Unmanned Aerial Vehicle*) Di sebagian Jalan Kemakmuran Kecamatan Sambaliung”. Untuk keperluan penelitian, penulis dapat dihubungi melalui e-mail sigittiolegowo.svm20@gmail.com.