Kapita Selekta 1

First Break Automation

Anggota Kelompok

- Muhammad Khibran Ibrahim (5017231061)
- Muhammad Arsyad (5017231068)
- Naila Davina Azzahra (5017241001)
- Aldyansyah Guna Pradangga (5017241002)
- Naomi Almira Kirani (5017241029)
- Agni Yumna Fahira (5017241038)
- Evelina Dewi Rahmawati (5017241055)

Latar Belakang

First Break adalah **waktu pertama kali gelombang** seismik terdeteksi oleh geophone setelah sumber ditembakkan. Digunakan untuk menentukan **waktu tempuh gelombang** dan menghitung **kecepatan lapisan bawah permukaan** pada metode seismik refraksi.

Proses first break picking biasanya dilakukan manual atau semi-manual, dengan cara mengamati seismogram satu per satu.
Masalah utama: memakan waktu lama, subjektif, dan sulit saat data ber-noise.

Latar Belakang

Untuk mengatasi hal tersebut, dikembangkan ide otomatisasi **first break picking berbasis machine learning (CNN U-Net)** dengan teknik image segmentation

CNN U-Net mampu mengenali pola waktu tiba pertama secara cepat, akurat, dan konsisten bahkan pada data yang bising..

Sehingga poses picking menjadi lebih efisien, minim kesalahan manusia, dan mendukung interpretasi kecepatan bawah permukaan secara lebih cepat.

Tujuan

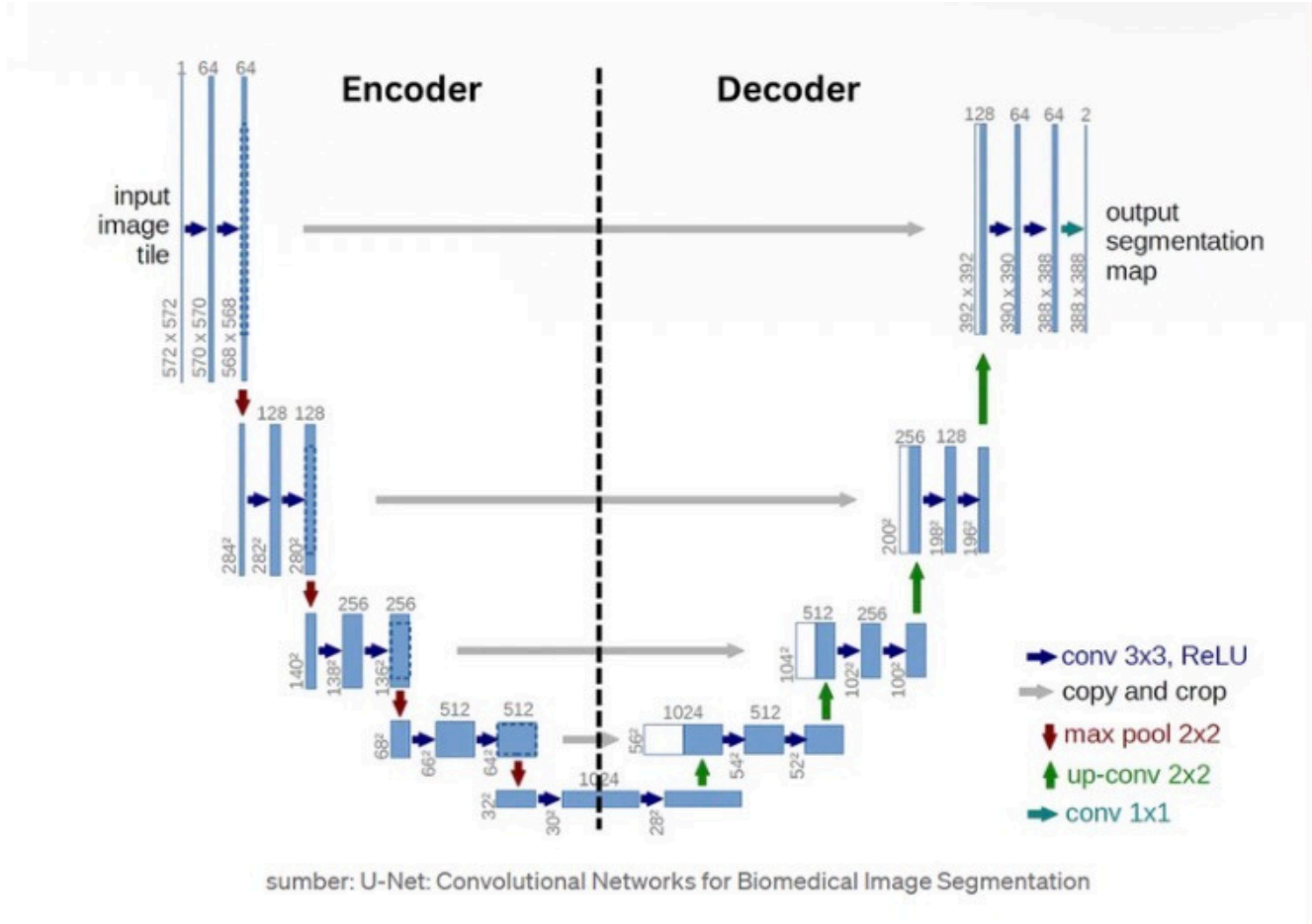
1. Mengetahui bagaimana cara dalam menerapkan model CNN U-Net yang mampu dalam melakukan deteksi dan segmentasi first break secara otomatis pada data seismik

1. Mengetahui bagaimana akurasi hasil automasi first break menggunakan CNN U-Net jika dibandingkan dengan metode picking manual atau konvensional

CNN U-NET

CNN U-Net adalah arsitektur Convolutional Neural Network (CNN) berbentuk huruf “U” yang digunakan untuk **segmentasi citra**. Model ini memiliki encoder untuk **mengekstraksi fitur** dan **decoder** untuk memulihkan detail citra, dengan skip connection agar informasi spasial tetap terjaga. CNN U-Net banyak digunakan dalam citra medis, penginderaan jauh, dan geofisika seperti deteksi lapisan seismik dan klasifikasi batuan.

Arsitektur Model



sumber: U-Net: Convolutional Networks for Biomedical Image Segmentation

Workflow

Preprocessing :
1. Normalization
2. Data Masking
3. Augmentation (If necessary since our data is big already)
4. Splitting Data

Model Building and Model Compilation

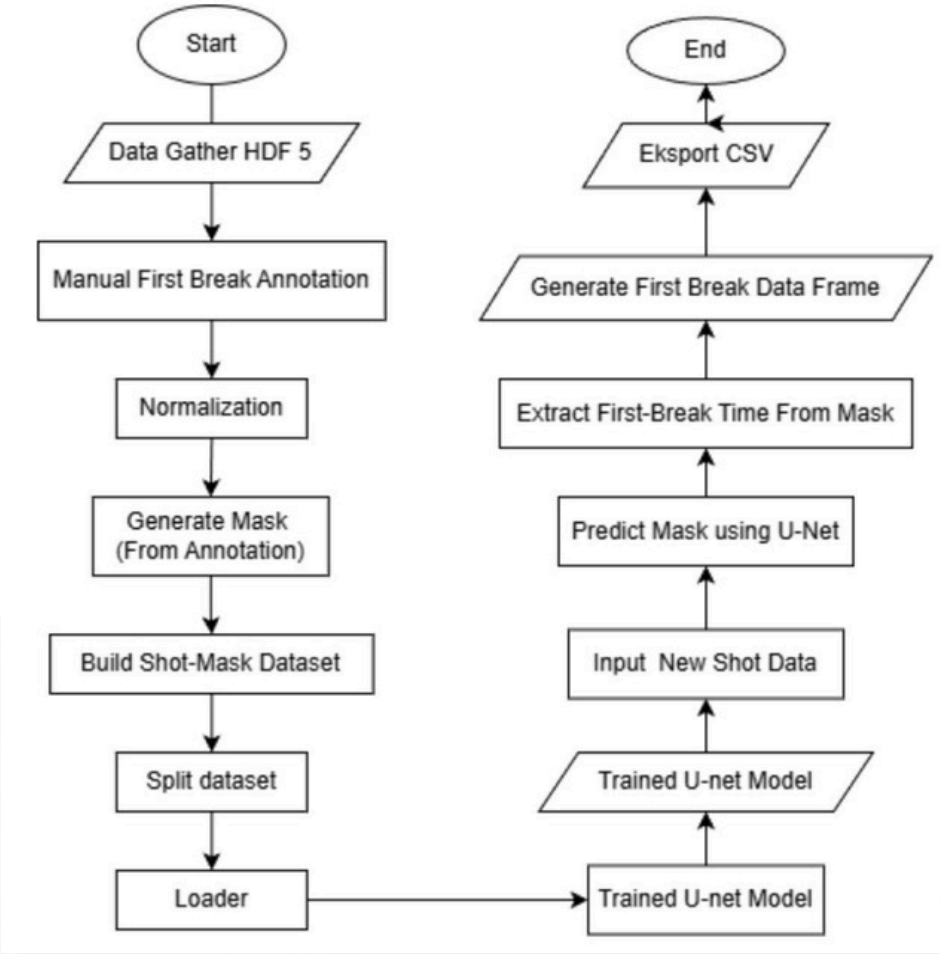
Training :
1. epochs
2. Batch-size
3. Validation Lose

Front end :
1. UI/Ux preparation
2. HTML + CSS

Back-end and integrate it with the model

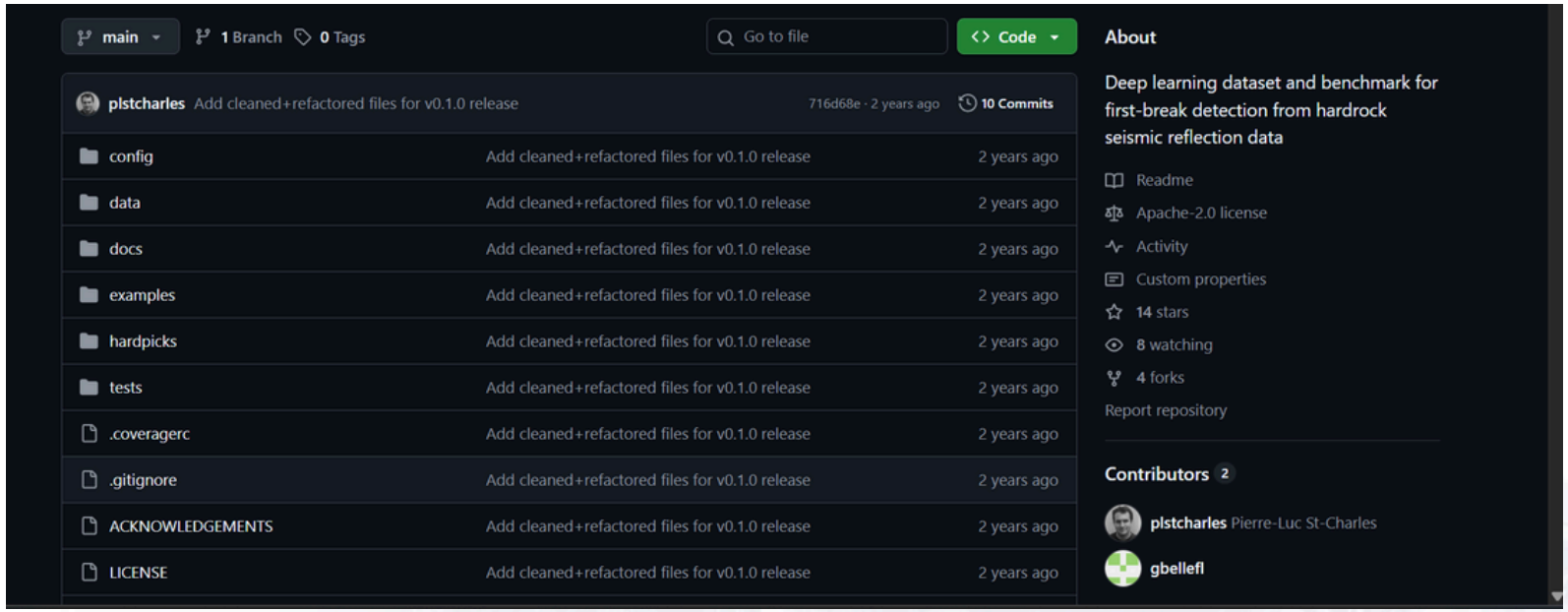
Deploy on vercell

FLOWCHART


Start
Data Gather HDF 5
Manual First Break Annotation
Normalization
Generate Mask
(From Annotation)
Build Shot-Mask Dataset
Split dataset
Loader
Trained U-net Model
Input New Shot Data
Predict Mask using U-Net
Extract First-Break Time From Mask
Generate First Break Data Frame
Eksport CSV
End

DATASET

Sumber : https://github.com/mila-iqia/hardpicks?tab=readme-ov-file



Dataset diambil dari github, yang merupakan real data milik perusahaan Mila and Natural Resources Canada, Brunswick. Data sebesar 8GB dengan 4496540 traces, 1541 shots, 6063 receivers, dan 18475 line gather

DATA PREPROCESSING

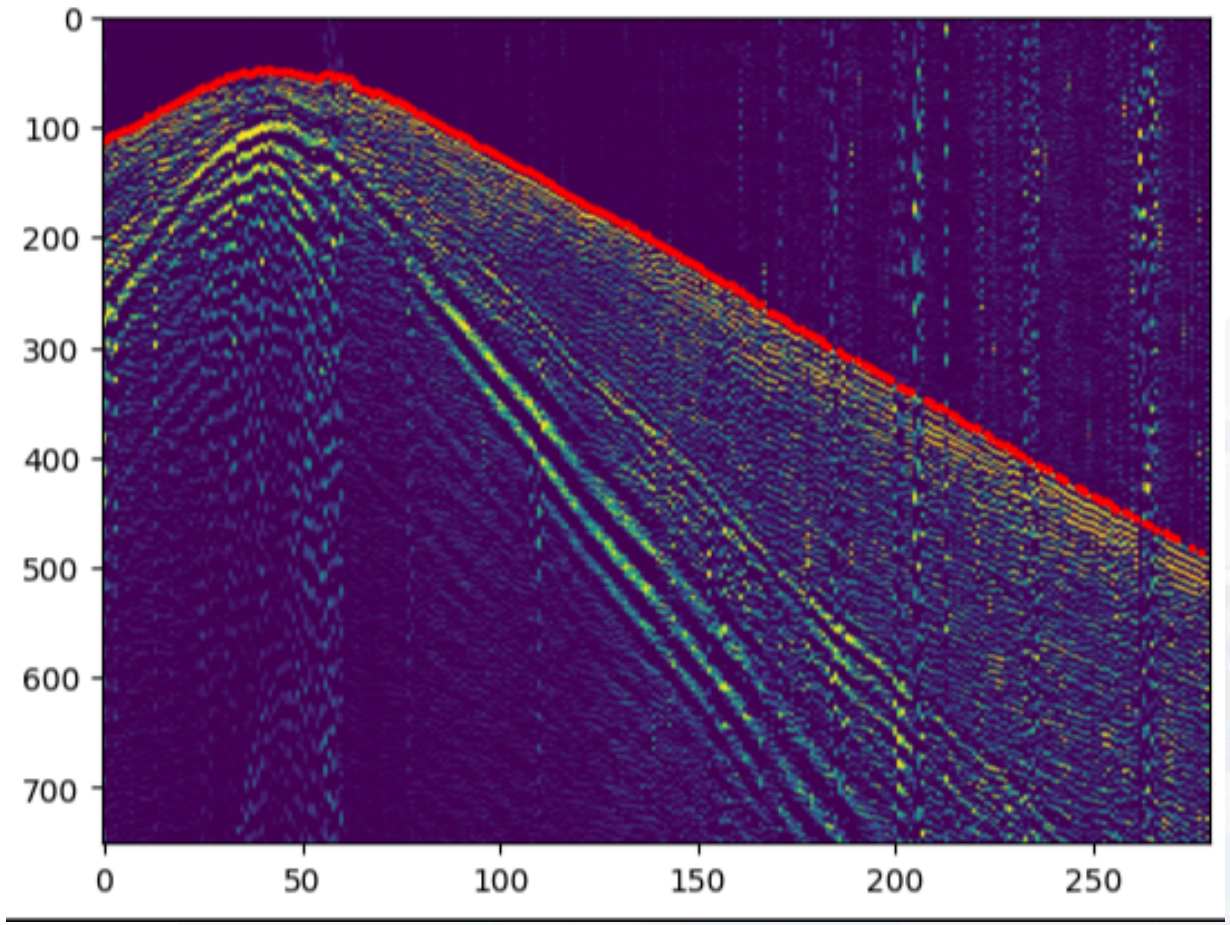
1. DATA READING AND VISUALIZATION

- Setup path & site info
- Dekompresi & validasi file MD5
- Validasi struktur HDF5
- Parse dengan TraceDataset (18475 gathers)
- Visualisasi raw (dengan clipping)

2. RAW AND MASKING DATA PRREPARATION

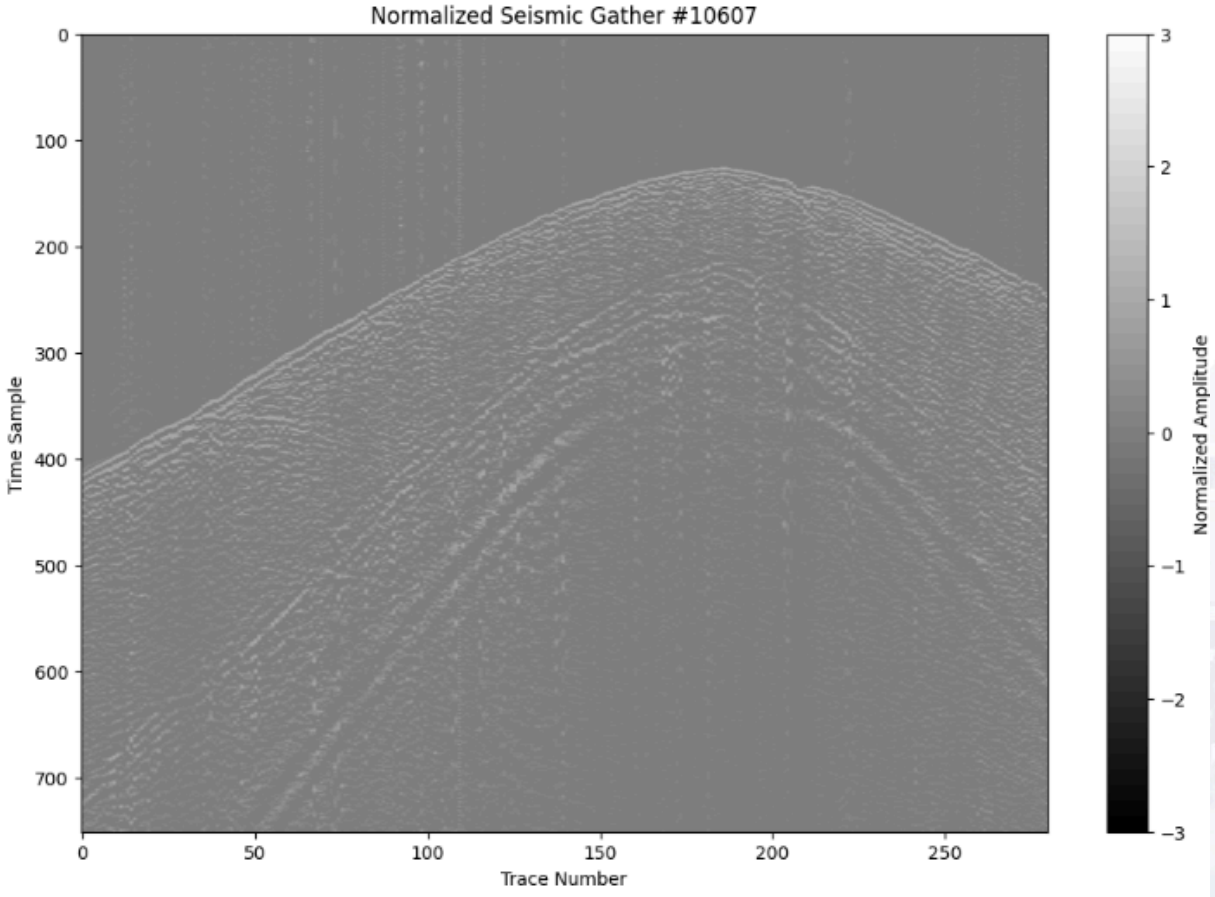
- Normalization
- Raw data visualization (tanpa clipping)
- Create mask (metode yang kami pakai : gaussian blob mask)
- Visualisasi raw seismic + mask + overlay
- Data Saving
- Data Splitting
- Add padding and Resize (jadi 512 x 256)

DATA PREPROCESSING



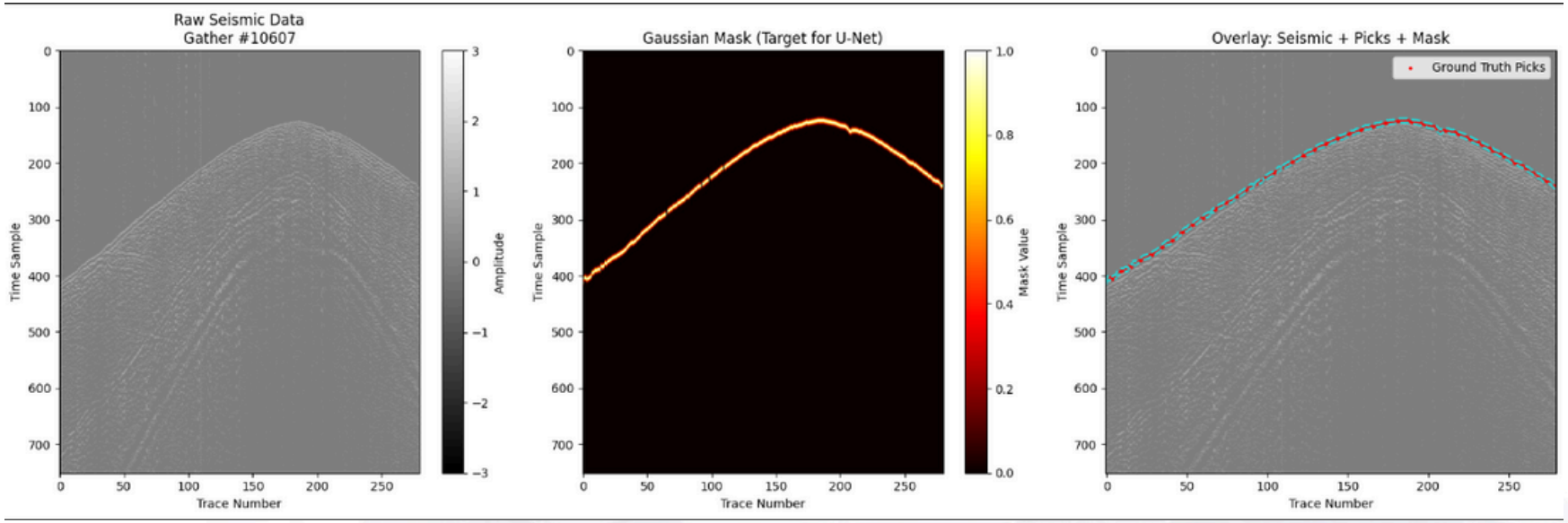
Karena dataset yang digunakan adalah real hasil akuisisi di lapangan, maka pasti amplitudonya memiliki skala yang curam, sehingga harus dinormalisasi. Selanjutnya, data tersebut di **staking** dan **clip** untuk memastikan distribusi nilai 0-1, lalu divisualisasikan dengan data time first-break nya (yang warna merah)

DATA PREPROCESSING



Divisualisasikan ulang tanpa data first time break, lalu color-map diatur ke grayscale.

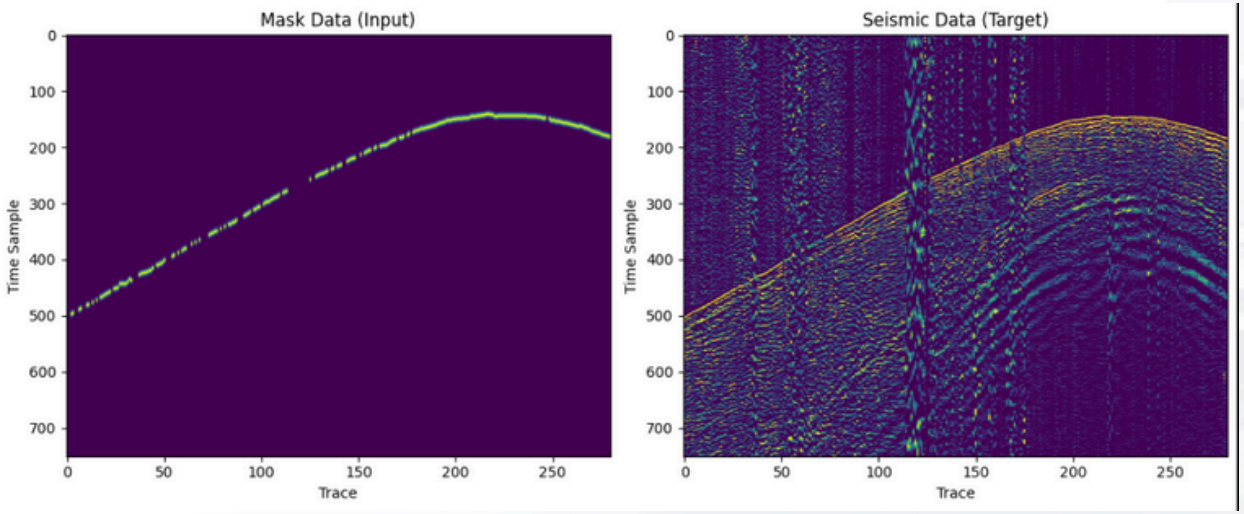
DATA PREPROCESSING



Masking data terhadap data raw yang belum dinormalisasi. Hasil mask kemudian divisualisasikan dan dibandingkan dengan data raw-normalized dan dan data ground-pick (data asli yang ada time first-break nya). Untuk time first break di ground truth menggunakan warna merah, sedangkan mask nya warna biru.

DATA PREPROCESSING

```
!unzip -o -q "/content/drive/MyDrive/FIRSTBREAK/ayo_gek_lang_training_dong.zip" -d /content/data_seismic
!ls -lh /content/data_seismic/

total 30G
-rw-rw-rw- 1 root root 30G Dec  2 16:36 ayo_gek_lang_training_dong.hdf5
```



Data diupload dalam bentuk zip ke Drive, lalu diunzip di google colab. Dilakukan pengecekan Data Sebelum masuk ke tahap splitting

DATA PREPROCESSING

```python
with h5py.File(hdf5_path, 'r') as f:
    total_samples = f['mask'].shape[0]
    all_indices = list(range(total_samples))

    train_indices, temp_indices = train_test_split(
        all_indices, test_size=0.3, random_state=config.random_seed
    )
    val_indices, test_indices = train_test_split(
        temp_indices, test_size=0.5, random_state=config.random_seed
    )

    train_ratio = 0.5
    val_ratio   = 0.5
    test_ratio  = 0.5

    train_indices = train_indices[:int(len(train_indices) * train_ratio)]
    val_indices   = val_indices[:int(len(val_indices)   * val_ratio)]
    test_indices  = test_indices[:int(len(test_indices) * test_ratio)]

    logger.info(f"Train samples: {len(train_indices)}")
    logger.info(f"Val   samples: {len(val_indices)}")
    logger.info(f"Test  samples: {len(test_indices)}")
```

Rasio yang digunakan untuk data splitting adalah 70% training, 15% validation, dan 15% testing.

Namun karena data 30Gb terlalu besar, maka digunakan subset untuk mereduce data ke setengahnya

DATA PREPROCESSING

```
        self.resized_w = new_w

        logger.info(f"Pad Resize mode aktif!")
        logger.info(f"  Scale factor : {self.scale:.3f}")
        logger.info(f"  Resize ke    : {new_w}x{new_h}")
        logger.info(f"  Final padding: top={self.pad_top}, bottom={self.pad_bottom}, "
                    f"left={self.pad_left}, right={self.pad_right}")

    def preprocess(self, data: np.ndarray) -> np.ndarray:
        if data.ndim == 2:
            data = np.expand_dims(data, axis=-1)

        # Resize
        tensor = tf.convert_to_tensor(data, dtype=tf.float32)
        tensor = tf.expand_dims(tensor, 0)
        resized = tf.image.resize(tensor, [self.resized_h, self.resized_w],
                                  method='bilinear')
        resized = tf.squeeze(resized, 0)

        # PAD
        padded = tf.pad(
            resized,
            [[self.pad_top, self.pad_bottom],
             [self.pad_left, self.pad_right],
            [0, 0]],
            mode='constant',
            constant_values=0
        )
        return padded.numpy()
```

Karena ukuran data (781, 280) maka di pad ke ukuran (1024, 5120) lalu di resize ke ukuran (512, 256). Dilakukan penambahan dimensi juga karena encoder yang dipakai adalah resnet 50 yang membutuhkan input 3 dimensi.

BUILD ARCHICTECTURE

3. Overall Structure of U-Net

- Classic U-Net architecture (encoder-decoder with skip connections)
- Encoder: Pretrained ResNet50 (ImageNet weights)
- Decoder: Vanilla decoder with 5 upsampling blocks
- Input: 512 × 256 × 3 (resized + letterbox padded seismic data)
- Output: 512 × 256 × 1 (probability map of first break)

Model: "unet_first_break"

Layer (type)	Output Shape	Param #
resnet_encoder (Functional)	((None, 256, 128, 64), (None, 128, 64, 256), (None, 64, 32, 512), (None, 32, 16, 1024), (None, 16, 8, 2048))	23,587,712
vanilla_decoder_3 (VanillaDecoder)	?	6,962,049

Total params: 30,549,761 (116.54 MB)
Trainable params: 30,494,657 (116.33 MB)
Non-trainable params: 55,104 (215.25 KB)

BUILD ARCHICTECTURE

4. Encoder (Resnet 50)

- Feature extraction layers:
  - conv1_relu
  - conv2_block3
  - conv3_block4
  - conv4_block6
  - conv5_block3
- Pretrained on ImageNet
- Output feature maps: 64, 256, 512, 1024, 2048 channels

5. Decoder (Vanilla)

- 5 decoder blocks with channels: [256, 128, 64, 32, 16]
- Each block: Conv2DTranspose (2×2 upsampling) + skip connection + 2× Conv2D + BatchNorm + ReLU
- Final layer: 1×1 Conv, menggunakan sigmoid activation (probability 0–1)

6. Loss Function and Metrics

- Combined loss: Binary Cross-Entropy + Dice Loss
- Key Metrics : Dice Coefficient, Accuracy, Precision

TRAINING, VALIDATING, AND TESTING

7. Training Setup

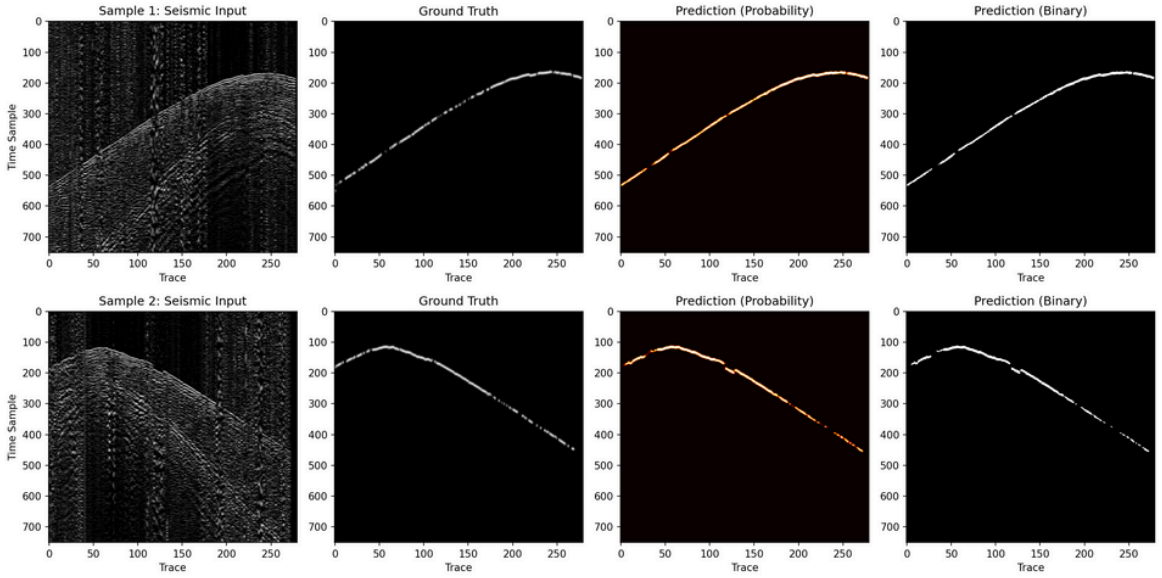
- Batch size: 24
- Learning rate: 0.002 (Adam optimizer)
- Max epochs: 10
- Scheduler: StepLR

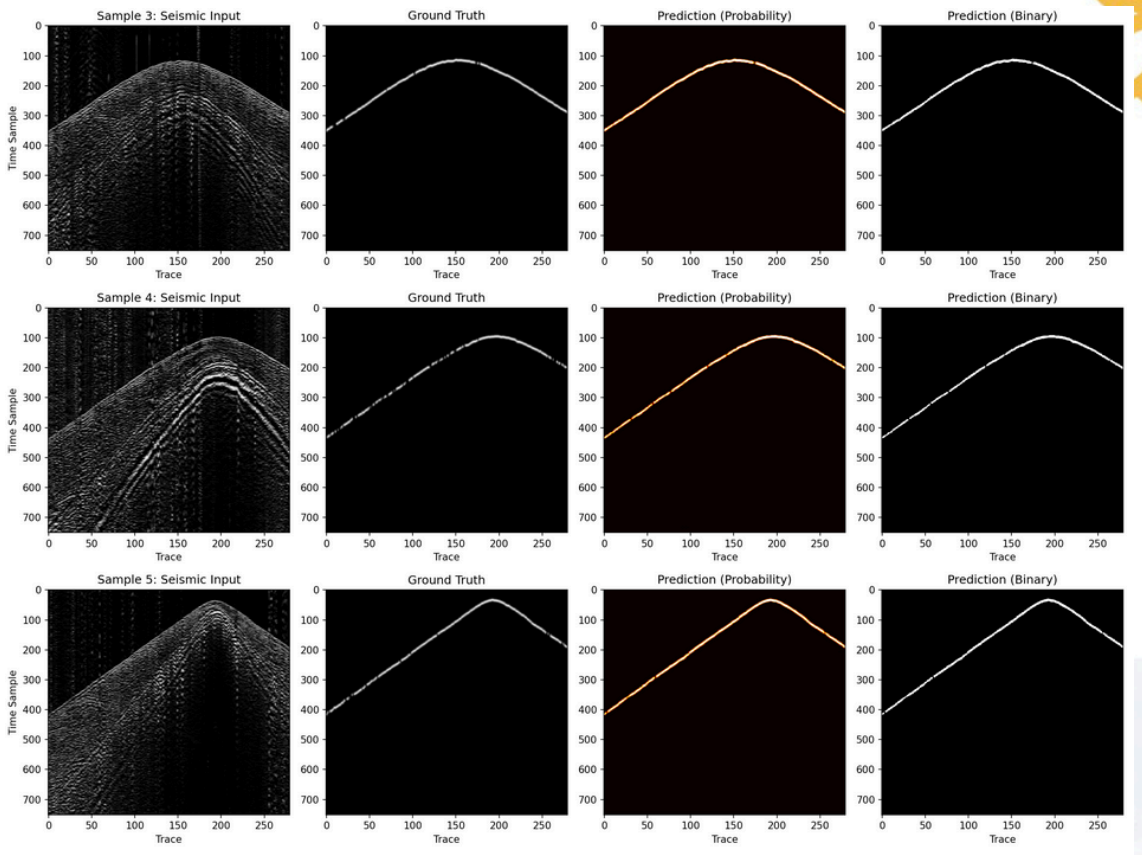
8. Callbacks

- ModelCheckpoint
- TensorBoard
- CSVLogger

```
Epoch 9: saving model to /content/drive/MyDrive/Seismic_Output_lagi/model_epoch009_val0.3182.h5
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legacy. We recommend using instead the native Keras f

Epoch 9: val_loss improved from 0.32498 to 0.31820, saving model to /content/drive/MyDrive/Seismic_Output_lagi/AKU_MAU_MODEL_YANG_DICE_BISA_GAK_SIHH_IHH.h5
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legacy. We recommend using instead the native Keras f

    Epoch 9 Summary:
    Train Loss      : 0.3158
    Train Accuracy  : 0.9736
    Val Loss        : 0.3182
    Val Accuracy    : 0.9734

270/270 ━━━━━━━━━━━━━━━━━━━━ 411s 2s/step - accuracy: 0.9737 - dice_coefficient: 0.6989 - loss: 0.3167 - precision: 0.9599 - recall: 0.2335 - val_accuracy: 0.9734 - val_dice_coefficient: 0.6972 -
Epoch 10/10

Epoch 10: saving model to /content/drive/MyDrive/Seismic_Output_lagi/model_epoch010_val0.3182.h5
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legacy. We recommend using instead the native Keras f

Epoch 10: val_loss did not improve from 0.31820

    Epoch 10 Summary:
    Train Loss      : 0.0000
    Train Accuracy  : 0.0000
    Val Loss        : 0.3182
    Val Accuracy    : 0.9734
```

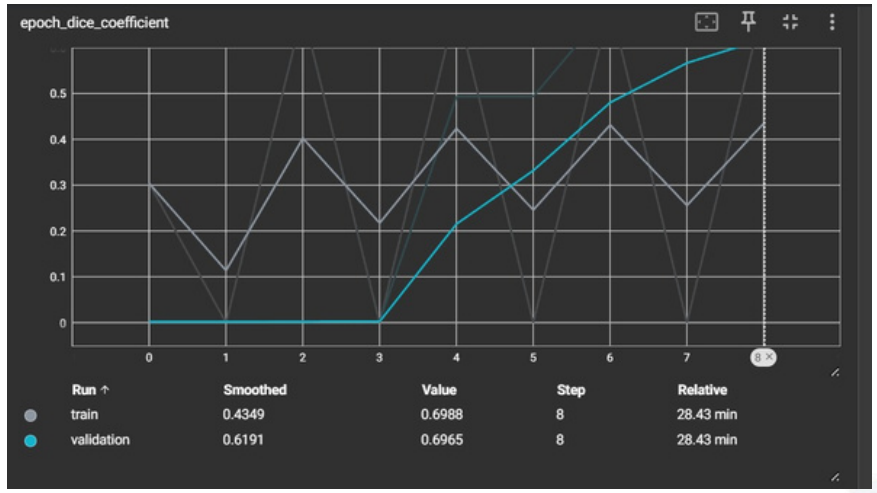
HASIL DAN EVALUASI

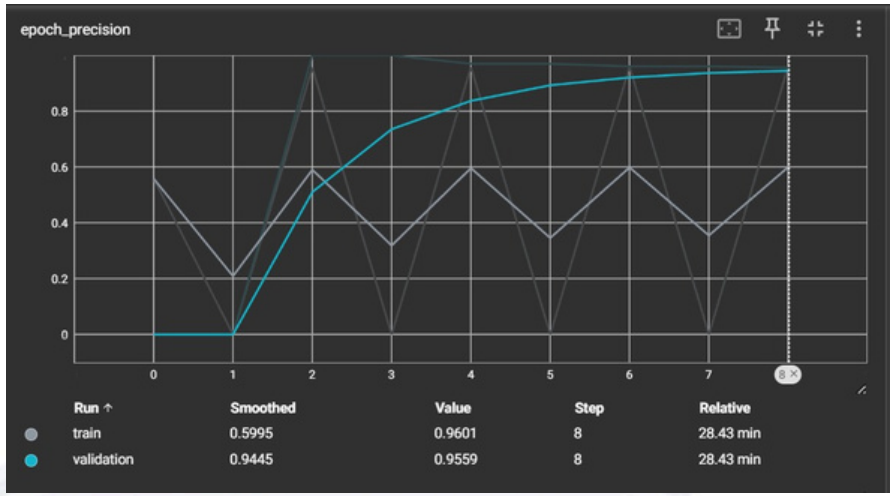


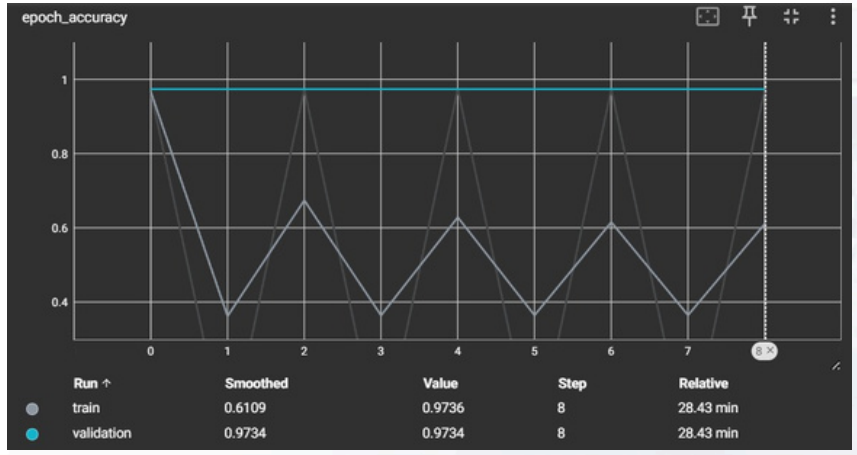


- Model menghasilkan probability map yang halus dan mengikuti pola first arrival dengan akurat.
- Setelah thresholding (0.5), garis first break prediksi hampir identik dengan ground truth pada berbagai bentuk gather (linear, hyperbolic).

HASIL DAN EVALUASI







Hasil Evaluasi yang diambil dari Tensorboard :

1. Berdasarkan epoch dice coefficient
2. Berdasarkan epoch accuracy
3. Berdasarkan epoch precision

HASIL DAN EVALUASI

epoch	accuracy	e_coefficie	loss	precision	recall	al_accurac	dice_coeffi	val_loss	al_precisio	val_recall
0	0.972135	0.425709	0.625997	0.82126	0.308862	0.97367	0.000552	1.04359	0	0
1	0	0	0	0	0	0.97367	0.000552	1.04359	0	0
2	0.973613	0.687294	0.328528	0.95704	0.23607	0.97367	7.71E-05	1.055189	0	0
3	0	0	0	0	0	0.97367	7.71E-05	1.055189	0	0
4	0.973625	0.694926	0.32079	0.958563	0.234361	0.973636	0.447005	0.579541	0.985408	0.087964
5	0	0	0	0	0	0.973636	0.447005	0.579541	0.985408	0.087964
6	0.973629	0.697833	0.317731	0.95915	0.23392	0.973432	0.689489	0.324978	0.960885	0.222631
7	0	0	0	0	0	0.973432	0.689489	0.324978	0.960885	0.222631
8	0.973637	0.699752	0.31575	0.960144	0.233467	0.973399	0.697215	0.318205	0.958702	0.238927
9	0	0	0	0	0	0.973399	0.697215	0.318205	0.958702	0.238927

Hasil Evaluasi Model yang diambil dari Tensor Log, dengan akurasi, dice, dan loss antara training dan validation yang tidak jauh. Model diperoleh dengan 10 epoch, dengan akurasi training epocj terakhir : 0.976 dan dice : 0.699 , lalu akurasi vaidation : 0.9734 dan dice : 0.6972

KESIMPULAN

Model CNN U-Net mampu mempelajari pola first break pada data seismik secara stabil dan konsisten, yang ditunjukkan oleh nilai loss training dan validasi yang seimbang serta nilai akurasi dan precision yang tinggi. Hal ini menandakan bahwa model efektif dalam mengenali area target first break.

Nilai Dice Coefficient sekitar 70% menunjukkan bahwa hasil segmentasi memiliki tingkat overlap yang cukup baik dengan ground truth. Meskipun sudah tergolong baik, performa ini masih dapat ditingkatkan melalui penambahan jumlah epoch dan data latih agar model lebih robust terhadap variasi sinyal.

Secara umum, model telah mampu mengikuti pola utama horizon seismik dan dapat dijadikan dasar yang baik untuk pengembangan metode automasi segmentasi first break selanjutnya.

ADVANCING HUMANITY

Thank you