Topik : 3.4. Bandingkan FL vs Centralized

Objective : Latih model centralized untuk dataset gabungan → bandingkan

Task : Buat tabel perbandingan akurasi & risiko privasi

Model centralized dengan Scikit-learn berarti kita melatih model machine learning secara terpusat (centralized learning), di mana semua data dari berbagai sumber dikumpulkan dulu dalam satu tempat/server, baru kemudian dipakai untuk training.

Gabungan CSV :

```
PS C:\KP\MATERI\3.4 Bandingkan FL vs Centralized> python read.py
      jumlah_tanggungan  penghasilan  kondisi_rumah  umur status_pekerjaan status_pernikahan riwayat_penyakit  status_gizi  tinggi_cm  berat_kg  layak_subsidi
0                     7      2168898  semi permanen    33            pns          menikah       tidak ada  kurang gizi        175        52              0
1                     3       905067  semi permanen    35       karyawan     belum menikah      hipertensi        lebih        154        81              1
2                     2      4070886          layak    52            pns          menikah       tidak ada       normal        157        64              1
3                     4      3039184  semi permanen    57          buruh     belum menikah             tbc       normal        186        94              1
4                     4      1449692          layak    53       mahasiswa     belum menikah        diabetes       normal        173        72              0
...                 ...          ...            ...   ...            ...              ...             ...          ...        ...       ...            ...
1495                  5      1358152  semi permanen    59          buruh       janda/duda        diabetes        lebih        187        87              1
1496                  2      4286261    tidak layak    32          petani       janda/duda        diabetes  kurang gizi        183        89              1
1497                  3      3318614          layak    47            pns       janda/duda            asma  kurang gizi        188        97              0
1498                  7      2291075    tidak layak    49          buruh          menikah        diabetes       normal        161        57              1
1499                  2      2829293    tidak layak    63       karyawan       janda/duda             tbc        lebih        174        78              1

[1500 rows x 11 columns]
PS C:\KP\MATERI\3.4 Bandingkan FL vs Centralized>
```

Centralized :

1. Load Dataset

```
11    df = pd.read_csv("gabungan.csv")
```

Semua data dari berbagai sumber (Dinsos, Dukcapil, Kemenkes) dikumpulkan jadi satu file CSV gabungan.

2. Preprocessing

```
# 2) Feature/label
y = df["layak_subsidi"]
X = df.drop(columns=["layak_subsidi"])

# 3) Definisikan kolom numerik & kategorikal
num_cols = ["jumlah_tanggungan", "penghasilan", "umur", "tinggi_cm", "berat_kg"]
cat_cols = ["kondisi_rumah", "status_pekerjaan", "status_pernikahan",
            "riwayat_penyakit", "status_gizi"]

#  jika ada NaN, isi sederhana
X[num_cols] = X[num_cols].fillna(0)
X[cat_cols] = X[cat_cols].fillna("unknown")

# 4) Preprocessor:
#    - Numerik: StandardScaler(with_mean=False) agar aman bila gabung sparse
#    - Kategorikal: OneHotEncoder(handle_unknown='ignore') biar robust
preprocessor = ColumnTransformer(
    transformers=[
        ("num", StandardScaler(with_mean=False), num_cols),
        ("cat", OneHotEncoder(handle_unknown="ignore"), cat_cols),
    ]
)

# 5) Model pipeline: preprocessor -> LogisticRegressionS
clf = Pipeline(steps=[
    ("prep", preprocessor),
    ("model", LogisticRegression(solver="liblinear", max_iter=200))
])
```

Handle missing value, scaling fitur numerik agar model lebih stabil.

3. Split Data

```
# 6) Split train/test
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)
```

Data dipisah jadi training (80%) dan testing (20%).

4. Train centralized model

```
# 7) Train
clf.fit(X_train, y_train)
```

Model dilatih dengan seluruh data

5. Evaluasi

```
# 8) Evaluasi
y_pred = clf.predict(X_test)
y_proba = clf.predict_proba(X_test)[:, 1]

print("Accuracy:", accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred, digits=4))
print("ROC-AUC:", roc_auc_score(y_test, y_proba))

cm = confusion_matrix(y_test, y_pred, labels=[0,1])
print("Confusion matrix [[TN FP],[FN TP]]:\n", cm)
```

```
PS C:\KP\MATERI\3.4 Bandingkan FL vs Centralized> python centralized.py
Accuracy: 0.51
              precision    recall  f1-score   support

           0     0.4960    0.4247    0.4576       146
           1     0.5200    0.5909    0.5532       154

    accuracy                         0.5100       300
   macro avg     0.5080    0.5078    0.5054       300
weighted avg     0.5083    0.5100    0.5067       300

ROC-AUC: 0.5273527842020993
Confusion matrix [[TN FP],[FN TP]]:
 [[62 84]
 [63 91]]
PS C:\KP\MATERI\3.4 Bandingkan FL vs Centralized>
```

Ukur performa model dengan accuracy, precision, recall, dan f1-score.

Hasil FL :

```
.
Skipping registering GPU devices...
Round 01 | loss=0.6943 | bin_acc=0.5484
Round 02 | loss=0.6862 | bin_acc=0.5884
Round 03 | loss=0.6786 | bin_acc=0.6051
Round 04 | loss=0.6714 | bin_acc=0.6182
Round 05 | loss=0.6645 | bin_acc=0.6527
Training selesai.
(venv) ezranahumury@DESKTOP-8003BIM:/mnt/c/KP/MATERI/Gabungan Materi 3.1 - 3.3$
```

Hasil Centralized :

```
PS C:\KP\MATERI\3.4 Bandingkan FL vs Centralized> python centralized.py
Accuracy: 0.51
              precision    recall  f1-score   support

           0     0.4960    0.4247    0.4576       146
           1     0.5200    0.5909    0.5532       154

    accuracy                         0.5100       300
   macro avg     0.5080    0.5078    0.5054       300
weighted avg     0.5083    0.5100    0.5067       300

ROC-AUC: 0.5273527842020993
Confusion matrix [[TN FP],[FN TP]]:
 [[62 84]
 [63 91]]
PS C:\KP\MATERI\3.4 Bandingkan FL vs Centralized>
```

Tabel Perbandingan FL vs Centralized

Aspek	Federated Learning (FL)	Centralized Learning
Accuracy	0.54 → 0.65 setelah 5 round (bin_acc)	±0.51 (ROC-AUC 0.52, hampir random)
Data	Data tetap di client, hanya model update yang dibagi	Semua data harus digabung di server
Privasi	Lebih aman: data mentah tidak keluar dari perangkat	Risiko tinggi: semua data dikirim & disimpan
Risiko Privasi	Rendah → karena raw data tetap lokal	Tinggi → data sensitif rentan bocor
Kelebihan	Aman untuk data sensitif (kesehatan, finansial, dsb.)	Sederhana, cepat, library banyak tersedia

1. **Performa Model**

Dari hasil percobaan, Federated Learning menunjukkan akurasi yang cenderung naik dari 0.54 → 0.65 selama 5 ronde training, sementara model Centralized (Scikit-learn Logistic Regression) hanya mencapai akurasi sekitar 0.51 dengan ROC-AUC mendekati 0.52 (hampir setara random guess).

- Rendahnya akurasi centralized lebih disebabkan karena dataset dummy kita diberi label acak, sehingga sulit bagi model menemukan pola yang bermakna.
- FL terlihat sedikit lebih stabil karena sifat agregasi model antar-klien membuat hasil tidak terlalu bergantung pada distribusi data tunggal.

2. **Privasi Data**

- Centralized learning mengharuskan semua data dikumpulkan ke satu server. Hal ini berisiko tinggi untuk privasi, terutama jika data berasal dari instansi berbeda (contoh: Dinsos, Dukcapil, Kemenkes). Data mentah bisa terekspos jika server diretas atau ada kebocoran.
- Federated Learning menjaga data tetap berada di perangkat atau institusi masing-masing. Hanya parameter model (gradien atau bobot terupdate) yang dikirim ke server pusat. Risiko kebocoran data mentah jauh lebih kecil, menjadikannya lebih cocok untuk kasus sensitif seperti kesehatan, keuangan, atau data kependudukan.