

Tutorial Analisis Kelayakan IDP (Program E1)

1. Tujuan

Panduan ini membantu menganalisis kelayakan peserta IDP berdasarkan:

- Program E1 (EPP, SDP, MDP, ADP)
- Level peserta (NML, FML, MML, SML)
- Umur peserta

Hasil analisis:

- Status kelayakan peserta (Layak / Tidak Layak)
- Alasan kelayakan
- Summary jumlah peserta layak dan tidak layak per program
- Total peserta layak per program dan keseluruhan

File hasil disimpan dalam Excel: Detail Data, Summary, Summary Extra.

2. Persiapan

1. Python 3.9+ sudah terinstal
2. Library pandas dan openpyxl:
pip install pandas openpyxl
3. File Excel tersedia di folder D:\IDP FINAL\
 - MASTER DATA IDP 2025.xlsx
 - List Employee.xlsx
4. Editor Python siap digunakan (VS Code / Jupyter / Colab)
5. Pastikan file Excel ditutup sebelum menjalankan script.

3. Struktur Folder

D:\IDP FINAL\
├── MASTER DATA IDP 2025.xlsx
├── List Employee.xlsx
└── Output_Kelayakan_IDP_Sum.xlsx (file hasil nanti)

4. Codingan Lengkap

```

import pandas as pd

# ===== KONFIGURASI FILE =====
master_file = "D:\\IDP FINAL\\MASTER DATA IDP 2025.xlsx"
list_employee_file = "D:\\IDP FINAL\\List Employee.xlsx"
output_file = "Output_Kelayakan_IDP_Sum.xlsx"

# ===== BACA DATA =====
master_df = pd.read_excel(master_file)
employee_df = pd.read_excel(list_employee_file)

# ===== SAMAKAN KOLOM =====
master_df = master_df[["NIK", "NAME", "LEVEL", "E1"]]
employee_df = employee_df[["No. Induk Karyawan", "Nama Karyawan", "Posisi", "Umur"]]

employee_df = employee_df.rename(columns={
    "No. Induk Karyawan": "NIK",
    "Nama Karyawan": "NAME",
    "Posisi": "JABATAN",
    "Umur": "UMUR"
})

# ===== GABUNGKAN DATA =====
final_df = pd.merge(master_df, employee_df, on=["NIK", "NAME"], how="left")
final_df = final_df[["NIK", "NAME", "JABATAN", "LEVEL", "UMUR", "E1"]]

# ===== ATURAN KELAYAKAN (MATRIX FINAL) =====
def cek_kelayakan(row):
    level = str(row["LEVEL"]).upper().strip()
    umur = row["UMUR"]
    e1 = str(row["E1"]).upper().strip()

    # ===== EPP =====
    if e1 == "EPP":
        if ("NML" in level or "FML" in level):
            if umur <= 35:
                return "Layak", "Memenuhi syarat Band 4 (NML/FML) dan usia ≤ 35 tahun untuk EPP"
            else:
                return "Tidak Layak", "Usia melebihi batas (≤ 35 tahun) untuk EPP"
        else:
            return "Tidak Layak", f"EPP hanya untuk Band 4 (NML/FML), bukan {level}"

    # ===== SDP =====
    elif e1 == "SDP":
        if ("FML" in level or "MML" in level):
            if umur <= 40:
                return "Layak", "Memenuhi syarat Band 3 (FML/MML) dan usia ≤ 40 tahun untuk SDP"
            else:

```

```

        return "Tidak Layak", "Usia melebihi batas ( $\leq 40$  tahun) untuk SDP"
    else:
        return "Tidak Layak", f"SDP hanya untuk Band 3 (FML/MML), bukan {level}"

# ===== MDP =====
elif e1 == "MDP":
    if ("MML" in level or "SML" in level):
        if umur <= 45:
            return "Layak", "Memenuhi syarat Band 2 & 3 (MML/SML) dan usia  $\leq 45$  tahun untuk MDP"
        else:
            return "Tidak Layak", "Usia melebihi batas ( $\leq 45$  tahun) untuk MDP"
    else:
        return "Tidak Layak", f"MDP hanya untuk Band 2 & 3 (MML/SML), bukan {level}"

# ===== ADP =====
elif e1 == "ADP":
    if ("MML" in level or "SML" in level):
        if umur <= 45:
            return "Layak", "Memenuhi syarat Band 2 (MML/SML) dan usia  $\leq 45$  tahun untuk ADP"
        else:
            return "Tidak Layak", "Usia melebihi batas ( $\leq 45$  tahun) untuk ADP"
    else:
        return "Tidak Layak", f"ADP hanya untuk Band 2 (MML/SML), bukan {level}"

# ===== Program lain =====
else:
    return "Tidak Layak", f"Program {e1} tidak dikenali dalam matrix"

# Terapkan aturan ke dataframe
final_df[["KELAYAKAN", "ALASAN"]] = final_df.apply(
    lambda row: pd.Series(cek_kelayakan(row)), axis=1
)

# ===== BUAT SUMMARY =====
summary = final_df.groupby(["E1", "KELAYAKAN"]).size().reset_index(name="Jumlah")
summary_pivot = summary.pivot(index="E1", columns="KELAYAKAN",
    values="Jumlah").fillna(0)
summary_pivot["Total"] = summary_pivot.sum(axis=1)

# Hitung total layak semua program
total_layak = (final_df["KELAYAKAN"] == "Layak").sum()

# Hitung jumlah layak per program
layak_per_program = final_df[final_df["KELAYAKAN"] == "Layak"].groupby("E1").size()

# Buat dataframe tambahan untuk summary
extra_summary = pd.DataFrame({

```

```

    "Keterangan": ["Total Layak"] + [f"Layak {prog}" for prog in layak_per_program.index],
    "Jumlah": [total_layak] + layak_per_program.tolist()
})

# ===== SIMPAN OUTPUT KE EXCEL =====
with pd.ExcelWriter(output_file, engine="openpyxl") as writer:
    final_df.to_excel(writer, sheet_name="Detail Data", index=False)
    summary_pivot.to_excel(writer, sheet_name="Summary")
    extra_summary.to_excel(writer, sheet_name="Summary Extra", index=False)

print("✅ Proses selesai! File hasil disimpan di:", output_file)
print("📊 Total peserta yang Layak:", total_layak)

```

Noted: gunakan file path yang sudah lengkap E1, E2, E3

5. Menjalankan Script

1. Pastikan file Excel ditutup.
2. Jalankan sel codingan di Jupyter Notebook.
3. Buka file Output_Kelayakan_IDP_Sum.xlsx untuk melihat hasil.

6. Contoh Output

Detail Data:

NIK	NAME	JABATAN	LEVEL	UMUR	E1	KELAYAKAN	ALASAN
12345	Budi Santoso	Manager	FML	34	EPP	Layak	Memenuhi syarat Band 4 ...
23456	Siti Aminah	Supervisor	MML	42	SDP	Tidak Layak	Usia melebihi batas (≤ 40 tahun)

Summary Pivot:

E1 Layak Tidak Layak Total

EPP	12	3	15
SDP	15	2	17

Summary Extra:

Keterangan Jumlah

Total Layak 27

Layak EPP 12

Layak SDP 15

7. Tips Tambahan

- Nama kolom Excel harus sama dengan script (NIK, NAME, LEVEL, E1).
- Gunakan nama file dan sheet sesuai konfigurasi.
- Tutup file Excel sebelum menjalankan script.
- Jalankan script per bagian jika dataset besar agar mudah debug.

8. Kesimpulan

Dengan tutorial ini, pengguna dapat:

- Menentukan kelayakan peserta IDP secara sistematis.
- Mengetahui jumlah peserta layak dan tidak layak per program.
- Mengetahui total peserta layak keseluruhan.
- Menghasilkan file Excel siap digunakan untuk laporan, visualisasi, atau dashboard IDP.