Filter Jakarta Timur

In [4]:
```
# ======================================================
# Analisis Zona Kritis Termal - Rawamangun
# Sesuai metodologi naskah IST (Indeks Stres Termal)
# VERSI 5.2 - Investigasi Nama Kota/Kabupaten
# ======================================================

import os
import sys
import geopandas as gpd
import pandas as pd
import numpy as np
import rasterio
import matplotlib.pyplot as plt
from shapely.geometry import box
from rasterstats import zonal_stats
from scipy.stats import pearsonr

# --- 1. Paths (Sesuaikan direktori Anda) ---
# Path ke direktori tempat file TIF (LST & NDVI) disimpan
data_dir = '/Users/admin/Documents/project pulogebang/proyek_pulogebang'
# Path ke file GeoJSON batas administrasi
gadm_geojson = os.path.expanduser('~/Downloads/gadm41_IDN_4.json')

# Membuat path lengkap untuk file raster dari direktori data
lst_path = os.path.join(data_dir, 'LST.TIF')   # LST hasil konversi dari Lands
ndvi_path = os.path.join(data_dir, 'NDVI.TIF') # NDVI hasil olahan dari Band 5


# --- Pengecekan Keberadaan File ---
def check_file_exists(filepath, filename):
    if not os.path.exists(filepath):
        print(f"--- ERROR ---")
        print(f"File '{filename}' tidak ditemukan di lokasi yang diharapkan.")
        print(f"Lokasi yang dicari: {filepath}")
        print("Pastikan path direktori dan nama file sudah benar.")
        sys.exit()

print("Memeriksa file input...")
check_file_exists(lst_path, 'LST.TIF')
check_file_exists(ndvi_path, 'NDVI.TIF')
check_file_exists(gadm_geojson, 'gadm41_IDN_4.json')
print("Semua file input ditemukan. Melanjutkan analisis...")
print("-" * 30)


# --- 2. Load & filter Rawamangun ---
try:
    gdf = gpd.read_file(gadm_geojson)
```

```
    # --- [PENTING] BAGIAN INVESTIGASI NAMA KOTA/KABUPATEN ---
    # Bagian ini sekarang aktif. Jalankan skrip untuk melihat daftar nama Kota
    # Setelah Anda menemukan nama yang benar, ikuti instruksi di output.
    available_cities = sorted(gdf['NAME_2'].unique())
    print("Daftar Kota/Kabupaten (NAME_2) yang tersedia di file GeoJSON:")
    print(available_cities)
    print("\n--- INSTRUKSI ---")
    print("1. Cari nama yang benar untuk 'Kota Jakarta Timur' dari daftar di a
    print("2. Ganti nilai variabel 'nama_kota_target' di bawah dengan nama yar
    print("3. Beri tanda pagar (#) pada 5 baris di atas (dari 'available_citie
    print("4. Jalankan kembali seluruh skrip.")
    sys.exit() # Menghentikan skrip setelah menampilkan daftar


    # GANTI 'kota jakarta timur' DENGAN NAMA YANG BENAR DARI DAFTAR DI ATAS
    nama_kota_target = 'kota jakarta timur'
    nama_kelurahan_target = 'rawamangun'

    rawamangun = gdf[
        (gdf['NAME_2'].str.lower() == nama_kota_target) &
        (gdf['NAME_4'].str.lower() == nama_kelurahan_target)
    ]
    if rawamangun.empty:
        raise ValueError(f"Kelurahan '{nama_kelurahan_target}' di '{nama_kota_
except Exception as e:
    print(f"Gagal memuat atau memfilter file GeoJSON: {e}")
    sys.exit()


# --- 3. Samakan CRS dan Buat Grid ---
try:
    with rasterio.open(lst_path) as src:
        raster_crs = src.crs
    rawamangun_reproj = rawamangun.to_crs(raster_crs)

    xmin, ymin, xmax, ymax = rawamangun_reproj.total_bounds
    grid_size = 300
    cols = np.arange(xmin, xmax + grid_size, grid_size)
    rows = np.arange(ymin, ymax + grid_size, grid_size)

    polygons = [box(cols[x], rows[y], cols[x+1], rows[y+1])
                for x in range(len(cols)-1) for y in range(len(rows)-1)]
    grid = gpd.GeoDataFrame({'geometry': polygons}, crs=raster_crs)
    grid_clipped = gpd.overlay(grid, rawamangun_reproj, how='intersection')

    if grid_clipped.empty:
        print("Tidak ada sel grid yang valid setelah dipotong dengan batas wil
        sys.exit()

    grid_clipped['cell_id'] = range(len(grid_clipped))

    print(f"Grid 300x300m untuk Rawamangun berhasil dibuat dengan total {len(g
```

```
except Exception as e:
    print(f"Terjadi error saat memproses data raster atau membuat grid: {e}")
    sys.exit()


# --- 4. Analisis Zonal Statistik (LST & NDVI) ---
stats_lst = zonal_stats(grid_clipped, lst_path, stats="mean", all_touched=True
stats_ndvi = zonal_stats(grid_clipped, ndvi_path, stats="mean", all_touched=Tr

df = pd.DataFrame({
    'cell_id': grid_clipped['cell_id'],
    'lst_mean': [s['mean'] for s in stats_lst],
    'ndvi_mean': [s['mean'] for s in stats_ndvi]
}).dropna()

# --- Pengecekan Jumlah Data Valid ---
if len(df) < 2:
    print(f"\n--- PERINGATAN ---")
    print(f"Jumlah sel grid dengan data yang valid hanya {len(df)}.")
    print("Analisis statistik (korelasi, klasifikasi) tidak dapat dilanjutkan
    print("Ini kemungkinan disebabkan oleh wilayah studi yang terlalu kecil at
    sys.exit()

# --- 5. Hitung Indeks Stres Termal (IST) ---
mu_lst = df['lst_mean'].mean()
sigma_lst = df['lst_mean'].std()
df['IST'] = (df['lst_mean'] - mu_lst) / sigma_lst

# --- 6. Klasifikasi Zona Termal ---
# Metode 1: Kuartil LST
q1, q3 = df['lst_mean'].quantile([0.25, 0.75])
def classify_lst(lst):
    if lst <= q1: return 'Aman'
    elif lst <= q3: return 'Waspada'
    else: return 'Kritis'
df['zona_LST'] = df['lst_mean'].apply(classify_lst)

# Metode 2: Ambang Batas IST
def classify_ist(ist):
    if ist > 1.5: return 'Kritis'
    elif ist > 0: return 'Waspada'
    else: return 'Aman'
df['zona_IST'] = df['IST'].apply(classify_ist)

grid_classified = grid_clipped.merge(df, on='cell_id')

# --- 7. Analisis Korelasi LST–NDVI ---
r, p = pearsonr(df['lst_mean'], df['ndvi_mean'])
print(f"Korelasi LST–NDVI di Rawamangun: r = {r:.4f}, p-value = {p:.4f}")

# --- 8. Visualisasi Peta ---
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(24, 12))
color_map = {'Aman': 'blue', 'Waspada': 'yellow', 'Kritis': 'red'}
```

```
custom_cmap = plt.matplotlib.colors.ListedColormap(color_map.values())

# Peta 1: Klasifikasi LST (Kuartil)
grid_classified.plot(column='zona_LST', ax=ax1, legend=True,
                     cmap=custom_cmap,
                     legend_kwds={'title': 'Zona Termal (LST - Kuartil)', 'loc
rawamangun_reproj.boundary.plot(ax=ax1, color='black', linewidth=1)
ax1.set_title('Peta Klasifikasi LST Rata-Rata (Kuartil) - Rawamangun', fontsiz
ax1.set_xlabel("Koordinat Timur (meter)")
ax1.set_ylabel("Koordinat Utara (meter)")

# Peta 2: Klasifikasi IST (Ambang Batas)
grid_classified.plot(column='zona_IST', ax=ax2, legend=True,
                     cmap=custom_cmap,
                     legend_kwds={'title': 'Zona Termal (IST - Ambang Batas)',
rawamangun_reproj.boundary.plot(ax=ax2, color='black', linewidth=1)
ax2.set_title('Peta Klasifikasi IST (Ambang Batas Statistik) - Rawamangun', fo
ax2.set_xlabel("Koordinat Timur (meter)")

# Simpan hasil visualisasi
output_path = os.path.expanduser("~/Downloads/peta_klasifikasi_rawamangun.png"
plt.savefig(output_path, dpi=300, bbox_inches='tight')
plt.show()

print(f"\nAnalisis selesai. Peta untuk Rawamangun tersimpan di: {output_path}"
```

```
Memeriksa file input...
Semua file input ditemukan. Melanjutkan analisis...
------------------------------
Daftar Kota/Kabupaten (NAME_2) yang tersedia di file GeoJSON:
['AcehBarat', 'AcehBaratDaya', 'AcehBesar', 'AcehJaya', 'AcehSelatan', 'AcehSin
gkil', 'AcehTamiang', 'AcehTengah', 'AcehTenggara', 'AcehTimur', 'AcehUtara',
'Agam', 'Alor', 'Ambon', 'Asahan', 'Asmat', 'Badung', 'Balangan', 'Balikpapan',
'BandaAceh', 'BandarLampung', 'Bandung', 'BandungBarat', 'Banggai', 'BanggaiKep
ulauan', 'Bangka', 'BangkaBarat', 'BangkaSelatan', 'BangkaTengah', 'Bangkalan',
'Bangli', 'Banjar', 'BanjarBaru', 'Banjarmasin', 'Banjarnegara', 'Bantaeng', 'B
antul', 'BanyuAsin', 'Banyumas', 'Banyuwangi', 'BaritoKuala', 'BaritoSelatan',
'BaritoTimur', 'BaritoUtara', 'Barru', 'Batam', 'Batang', 'BatangHari', 'Batu',
'BatuBara', 'Bau-Bau', 'Bekasi', 'Belitung', 'BelitungTimur', 'Belu', 'BenerMer
iah', 'Bengkalis', 'Bengkayang', 'Bengkulu', 'BengkuluSelatan', 'BengkuluTenga
h', 'BengkuluUtara', 'Berau', 'BiakNumfor', 'Bima', 'Bintan', 'Bireuen', 'Bitun
g', 'Blitar', 'Blora', 'Boalemo', 'Bogor', 'Bojonegoro', 'BolaangMongondow', 'B
olaangMongondowSelatan', 'BolaangMongondowTimur', 'BolaangMongondowUtara', 'Bom
bana', 'Bondowoso', 'Bone', 'BoneBolango', 'Bontang', 'BovenDigoel', 'Boyolal
i', 'Brebes', 'Bukittinggi', 'Buleleng', 'Bulukumba', 'Bulungan', 'Bungo', 'Buo
l', 'Buru', 'BuruSelatan', 'Buton', 'ButonUtara', 'Ciamis', 'Cianjur', 'Cilaca
p', 'Cilegon', 'Cimahi', 'Cirebon', 'Dairi', 'Danau', 'DanauLimboto', 'Deiyai',
'DeliSerdang', 'Demak', 'Denpasar', 'Depok', 'Dharmasraya', 'Dogiyai', 'Dompu',
'Donggala', 'Dumai', 'EmpatLawang', 'Ende', 'Enrekang', 'Fakfak', 'FloresTimu
r', 'Garut', 'GayoLues', 'Gianyar', 'Gorontalo', 'GorontaloUtara', 'Gowa', 'Gre
sik', 'Grobogan', 'GunungKidul', 'GunungMas', 'Gunungsitoli', 'HalmaheraBarat',
'HalmaheraSelatan', 'HalmaheraTengah', 'HalmaheraTimur', 'HalmaheraUtara', 'Hul
uSungaiSelatan', 'HuluSungaiTengah', 'HuluSungaiUtara', 'HumbangHasundutan', 'I
ndragiriHilir', 'IndragiriHulu', 'Indramayu', 'IntanJaya', 'JakartaBarat', 'Jak
artaPusat', 'JakartaSelatan', 'JakartaTimur', 'JakartaUtara', 'Jambi', 'Jayapur
a', 'Jayawijaya', 'Jember', 'Jembrana', 'Jeneponto', 'Jepara', 'Jombang', 'Kaim
ana', 'Kampar', 'Kapuas', 'KapuasHulu', 'Karanganyar', 'Karangasem', 'Karawan
g', 'Karimun', 'Karo', 'Katingan', 'Kaur', 'KayongUtara', 'Kebumen', 'Kediri',
'Keerom', 'Kendal', 'Kendari', 'Kepahiang', 'KepulauanAnambas', 'KepulauanAru',
'KepulauanMentawai', 'KepulauanMeranti', 'KepulauanSangihe', 'KepulauanSelaya
r', 'KepulauanSeribu', 'KepulauanSula', 'KepulauanTalaud', 'KepulauanYapen', 'K
erinci', 'Ketapang', 'Klaten', 'Klungkung', 'Kolaka', 'KolakaUtara', 'Konawe',
'KonaweSelatan', 'KonaweUtara', 'KotaBandung', 'KotaBaru', 'KotaBekasi', 'KotaB
ima', 'KotaBinjai', 'KotaBlitar', 'KotaBogor', 'KotaCirebon', 'KotaGorontalo',
'KotaJayapura', 'KotaKediri', 'KotaKupang', 'KotaMadiun', 'KotaMagelang', 'Kota
Malang', 'KotaMedan', 'KotaMojokerto', 'KotaPasuruan', 'KotaPekalongan', 'KotaP
ontianak', 'KotaProbolinggo', 'KotaSemarang', 'KotaSerang', 'KotaSolok', 'KotaS
orong', 'KotaSukabumi', 'KotaTangerang', 'KotaTanjungbalai', 'KotaTasikmalaya',
'KotaTegal', 'KotaYogyakarta', 'Kotamobagu', 'KotawaringinBarat', 'Kotawaringin
Timur', 'KuantanSingingi', 'KubuRaya', 'Kudus', 'KulonProgo', 'Kuningan', 'Kupa
ng', 'KutaiBarat', 'KutaiKartanegara', 'KutaiTimur', 'Labuhanbatu', 'Labuhanbat
uSelatan', 'LabuhanbatuUtara', 'Lahat', 'LakeToba', 'Lamandau', 'Lamongan', 'La
mpungBarat', 'LampungSelatan', 'LampungTengah', 'LampungTimur', 'LampungUtara',
'Landak', 'Langkat', 'Langsa', 'LannyJaya', 'Lebak', 'Lebong', 'Lembata', 'Lhok
seumawe', 'LimaPuluhKota', 'Lingga', 'LombokBarat', 'LombokTengah', 'LombokTimu
r', 'LombokUtara', 'Lubuklinggau', 'Lumajang', 'Luwu', 'LuwuTimur', 'LuwuUtar
a', 'Madiun', 'Magelang', 'Magetan', 'Majalengka', 'Majene', 'Makassar', 'Malan
g', 'Malinau', 'MalukuBaratDaya', 'MalukuTengah', 'MalukuTenggara', 'MalukuTeng
garaBarat', 'Mamasa', 'MamberamoRaya', 'MamberamoTengah', 'Mamuju', 'MamujuUtar
a', 'Manado', 'MandailingNatal', 'Manggarai', 'ManggaraiBarat', 'ManggaraiTimu
```

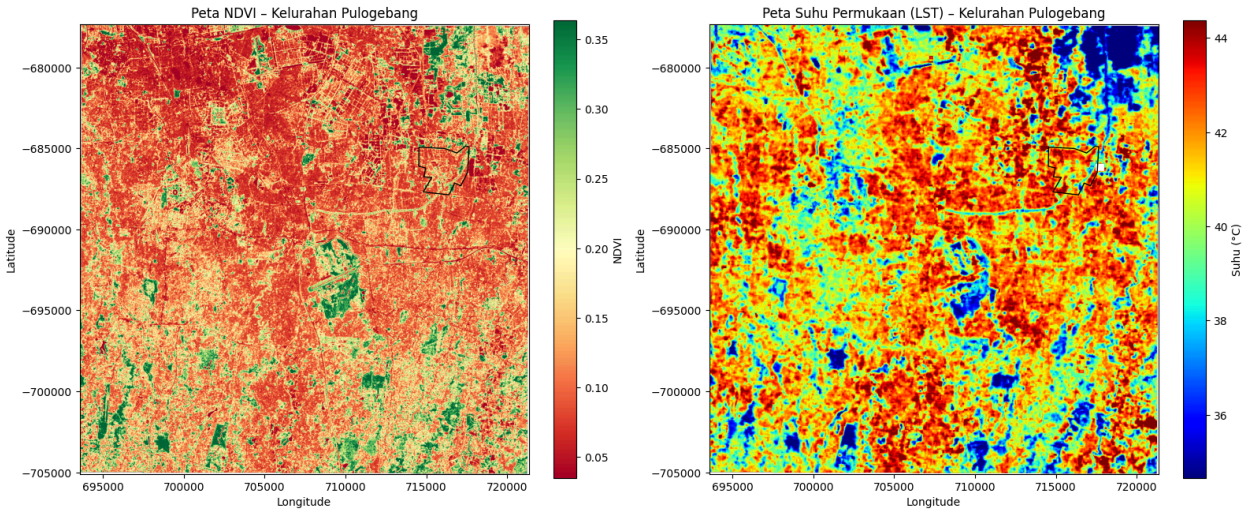
```
r', 'Manokwari', 'Mappi', 'Maros', 'Mataram', 'Maybrat', 'Melawi', 'Merangin',
'Merauke', 'Mesuji', 'Metro', 'Mimika', 'Minahasa', 'MinahasaSelatan', 'Minahas
aTenggara', 'MinahasaUtara', 'Mojokerto', 'Morowali', 'MuaraEnim', 'MuaroJamb
i', 'Mukomuko', 'Muna', 'MurungRaya', 'MusiBanyuasin', 'MusiRawas', 'Nabire',
'NaganRaya', 'Nagekeo', 'Natuna', 'Nduga', 'Ngada', 'Nganjuk', 'Ngawi', 'Nias',
'NiasBarat', 'NiasSelatan', 'NiasUtara', 'Nunukan', 'OganIlir', 'OganKomeringIl
ir', 'OganKomeringUlu', 'OganKomeringUluSelatan', 'OganKomeringUluTimur', 'Paci
tan', 'Padang', 'PadangLawas', 'PadangLawasUtara', 'PadangPanjang', 'PadangPari
aman', 'Padangsidimpuan', 'PagarAlam', 'PakpakBarat', 'PalangkaRaya', 'Palemban
g', 'Palopo', 'Palu', 'Pamekasan', 'Pandeglang', 'PangkajeneDanKepulauan', 'Pan
gkalpinang', 'Paniai', 'Parepare', 'Pariaman', 'ParigiMoutong', 'Pasaman', 'Pas
amanBarat', 'Paser', 'Pasuruan', 'Pati', 'Payakumbuh', 'PegununganBintang', 'Pe
kalongan', 'Pekanbaru', 'Pelalawan', 'Pemalang', 'Pematangsiantar', 'PenajamPas
erUtara', 'Pesawaran', 'PesisirSelatan', 'Pidie', 'PidieJaya', 'Pinrang', 'Pohu
wato', 'PolewaliMandar', 'Ponorogo', 'Pontianak', 'Poso', 'Prabumulih', 'Prings
ewu', 'Probolinggo', 'PulangPisau', 'PulauMorotai', 'Puncak', 'PuncakJaya', 'Pu
rbalingga', 'Purwakarta', 'Purworejo', 'RajaAmpat', 'RejangLebong', 'Rembang',
'RokanHilir', 'RokanHulu', 'RoteNdao', 'Sabang', 'SabuRaijua', 'Salatiga', 'Sam
arinda', 'Sambas', 'Samosir', 'Sampang', 'Sanggau', 'Sarmi', 'Sarolangun', 'Saw
ahlunto', 'Sekadau', 'Seluma', 'Semarang', 'SeramBagianBarat', 'SeramBagianTimu
r', 'Serang', 'SerdangBedagai', 'Seruyan', 'Siak', 'SiauTagulandangBiaro', 'Sib
olga', 'SidenrengRappang', 'Sidoarjo', 'Sigi', 'Sijunjung', 'Sikka', 'Simalungu
n', 'Simeulue', 'Singkawang', 'Sinjai', 'Sintang', 'Situbondo', 'Sleman', 'Solo
k', 'SolokSelatan', 'Soppeng', 'Sorong', 'SorongSelatan', 'Sragen', 'Subang',
'Subulussalam', 'Sukabumi', 'Sukamara', 'Sukoharjo', 'SumbaBarat', 'SumbaBaratD
aya', 'SumbaTengah', 'SumbaTimur', 'Sumbawa', 'SumbawaBarat', 'Sumedang', 'Sume
nep', 'SungaiPenuh', 'Supiori', 'Surabaya', 'Surakarta', 'Tabalong', 'Tabanan',
'Takalar', 'Tambrauw', 'TanaTidung', 'TanaToraja', 'TanahBumbu', 'TanahDatar',
'TanahLaut', 'Tangerang', 'TangerangSelatan', 'Tanggamus', 'TanjungJabungB', 'T
anjungJabungT', 'Tanjungpinang', 'TapanuliSelatan', 'TapanuliTengah', 'Tapanuli
Utara', 'Tapin', 'Tarakan', 'Tasikmalaya', 'Tebingtinggi', 'Tebo', 'Tegal', 'Te
lukBintuni', 'TelukWondama', 'Temanggung', 'Ternate', 'TidoreKepulauan', 'Timor
TengahSelatan', 'TimorTengahUtara', 'TobaSamosir', 'TojoUna-Una', 'Toli-Toli',
'Tolikara', 'Tomohon', 'TorajaUtara', 'Trenggalek', 'Tual', 'Tuban', 'TulangBaw
angBarat', 'Tulangbawang', 'Tulungagung', 'WadukCirata', 'WadukKedungombo', 'Wa
jo', 'Wakatobi', 'Waropen', 'WayKanan', 'Wonogiri', 'Wonosobo', 'Yahukimo', 'Ya
limo']

--- INSTRUKSI ---
1. Cari nama yang benar untuk 'Kota Jakarta Timur' dari daftar di atas.
2. Ganti nilai variabel 'nama_kota_target' di bawah dengan nama yang benar.
3. Beri tanda pagar (#) pada 5 baris di atas (dari 'available_cities' hingga 's
ys.exit()') untuk menonaktifkan investigasi.
4. Jalankan kembali seluruh skrip.
An exception has occurred, use %tb to see the full traceback.

SystemExit
/Users/admin/.pyenv/versions/3.13.2/lib/python3.13/site-packages/IPython/core/i
nteractiveshell.py:3557: UserWarning: To exit: use 'exit', 'quit', or Ctrl-D.
  warn("To exit: use 'exit', 'quit', or Ctrl-D.", stacklevel=1)
```

Gambar Peta Jakarta No Clipping

In [1]:

```
import os
import numpy as np
import rasterio
from rasterio.plot import show
import geopandas as gpd
import matplotlib.pyplot as plt

# ------------------------------
# 1. Konfigurasi Path Data
# ------------------------------
work_dir    = '/Users/admin/Documents/project pulogebang/proyek_pulogebang'
ndvi_tif    = os.path.join(work_dir, 'NDVI.TIF')
lst_tif     = os.path.join(work_dir, 'LST.TIF')
# GADM level-4 GeoJSON di Downloads yang memuat batas Pulogebang
gadm_geojson= os.path.expanduser('~/Downloads/gadm41_IDN_4.json')

# ------------------------------
# 2. Load Boundary Pulogebang
# ------------------------------
gdf = gpd.read_file(gadm_geojson)
pulog = gdf[gdf['NAME_4'].str.lower() == 'pulogebang']
if pulog.empty:
    raise ValueError("Kelurahan 'Pulogebang' tidak ditemukan di GADM GeoJSON."

# ------------------------------
# 3. Baca Rasters & Hitung Stretch
# ------------------------------
with rasterio.open(ndvi_tif) as src_ndvi:
    ndvi = src_ndvi.read(1).astype('float32')
    transform = src_ndvi.transform
    crs       = src_ndvi.crs
    vmin_ndvi = np.nanpercentile(ndvi, 2)
    vmax_ndvi = np.nanpercentile(ndvi, 98)

with rasterio.open(lst_tif) as src_lst:
    lst = src_lst.read(1).astype('float32')
    vmin_lst = np.nanpercentile(lst, 2)
    vmax_lst = np.nanpercentile(lst, 98)

# Reproject boundary ke CRS raster
pulog = pulog.to_crs(crs)

# ------------------------------
# 4. Plot NDVI & LST dengan Overlay Boundary
# ------------------------------
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(16, 8))

# NDVI
show(ndvi, transform=transform, ax=ax1,
```

```
     cmap='RdYlGn', vmin=vmin_ndvi, vmax=vmax_ndvi)
pulog.boundary.plot(ax=ax1, edgecolor='black', linewidth=1)
ax1.set_title('Peta NDVI – Kelurahan Pulogebang')
ax1.set_xlabel('Longitude')
ax1.set_ylabel('Latitude')
ax1.axis('on')  # tampilkan koordinat

# LST
show(lst, transform=transform, ax=ax2,
     cmap='jet', vmin=vmin_lst, vmax=vmax_lst)
pulog.boundary.plot(ax=ax2, edgecolor='black', linewidth=1)
ax2.set_title('Peta Suhu Permukaan (LST) – Kelurahan Pulogebang')
ax2.set_xlabel('Longitude')
ax2.set_ylabel('Latitude')
ax2.axis('on')

# Colorbars
cax1 = fig.colorbar(ax1.images[0], ax=ax1, orientation='vertical', fraction=0.
cax1.set_label('NDVI')
cax2 = fig.colorbar(ax2.images[0], ax=ax2, orientation='vertical', fraction=0.
cax2.set_label('Suhu (°C)')

plt.tight_layout()
plt.show()
```

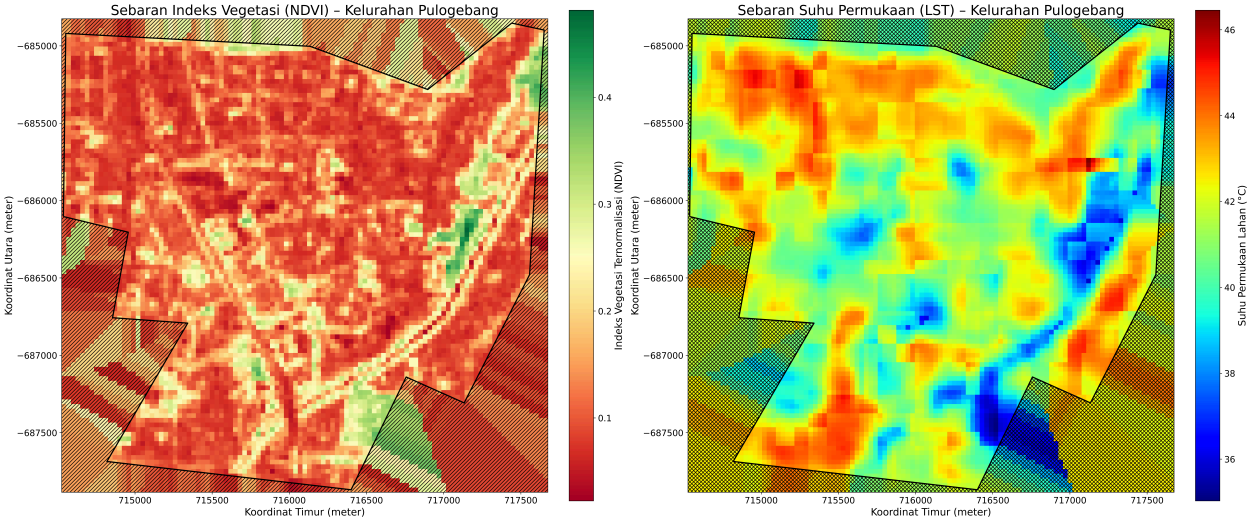


Clipping Pulogebang

In [2]:
```
# Import library yang dibutuhkan
import os
import numpy as np
import rasterio
import geopandas as gpd
import matplotlib.pyplot as plt
from rasterio.mask import mask
from rasterio.plot import show
from shapely.geometry import box
```

```
from scipy.interpolate import griddata

# 1. Paths
work_dir = '/Users/admin/Documents/project pulogebang/proyek_pulogebang'
ndvi_path = os.path.join(work_dir, 'NDVI.TIF')
lst_path = os.path.join(work_dir, 'LST.TIF')
gadm_geojson = os.path.expanduser('~/Downloads/gadm41_IDN_4.json')

# 2. Load & filter Pulogebang
gdf = gpd.read_file(gadm_geojson)
pulog = gdf[gdf['NAME_4'].str.lower() == 'pulogebang']
if pulog.empty:
    raise ValueError("Kelurahan 'Pulogebang' tidak ditemukan.")

# 3. Crop & mask rasters
def crop_mask(path):
    with rasterio.open(path) as src:
        geom = [pulog.to_crs(src.crs).geometry.values[0].__geo_interface__]
        arr, trans = mask(src, geom, crop=True, nodata=np.nan)
        return arr[0].astype('float32'), trans, src.crs

ndvi_crop, ndvi_trans, crs = crop_mask(ndvi_path)
lst_crop, lst_trans, _ = crop_mask(lst_path)

# 4. Fill NaN dengan nearest-neighbor
def fill_nan(arr):
    rows, cols = arr.shape
    X, Y = np.meshgrid(np.arange(cols), np.arange(rows))
    valid = ~np.isnan(arr)
    return griddata(
        (X[valid], Y[valid]),
        arr[valid],
        (X, Y),
        method='nearest'
    )

ndvi_filled = fill_nan(ndvi_crop)
lst_filled = fill_nan(lst_crop)

# 5. Buat poligon arsiran luar wilayah
h, w = ndvi_crop.shape
xmin, ymax = ndvi_trans * (0, 0)
xmax, ymin = ndvi_trans * (w, h)
bbox_poly = box(xmin, ymin, xmax, ymax)
pulog_poly = pulog.to_crs(crs).geometry.values[0]
outside_poly = bbox_poly.difference(pulog_poly)
outside_gdf = gpd.GeoDataFrame(geometry=[outside_poly], crs=crs)

# 6. Siapkan colormap transparan
cmap_ndvi = plt.cm.RdYlGn.copy(); cmap_ndvi.set_bad('none')
cmap_lst = plt.cm.jet.copy(); cmap_lst.set_bad('none')

# 7. Plotting dengan label besar
```

```
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(28, 14), dpi=600)

# --- NDVI ---
ndvi_extent = rasterio.plot.plotting_extent(ndvi_filled, ndvi_trans)
im1 = ax1.imshow(ndvi_filled, cmap=cmap_ndvi, extent=ndvi_extent, origin='uppe
outside_gdf.plot(ax=ax1, facecolor='none',
                 edgecolor='black', hatch='////',
                 linewidth=0, zorder=2)
pulog.to_crs(crs).boundary.plot(ax=ax1,
                                edgecolor='black', linewidth=2, zorder=3)

ax1.set_title('Sebaran Indeks Vegetasi (NDVI) – Kelurahan Pulogebang', fontsiz
ax1.set_xlabel('Koordinat Timur (meter)', fontsize=16)
ax1.set_ylabel('Koordinat Utara (meter)', fontsize=16)
cbar1 = plt.colorbar(im1, ax=ax1, fraction=0.046, pad=0.04)
cbar1.set_label('Indeks Vegetasi Ternormalisasi (NDVI)', fontsize=16)
cbar1.ax.tick_params(labelsize=14)
ax1.tick_params(axis='both', which='major', labelsize=14)

# --- LST ---
lst_extent = rasterio.plot.plotting_extent(lst_filled, lst_trans)
im2 = ax2.imshow(lst_filled, cmap=cmap_lst, extent=lst_extent, origin='upper',
outside_gdf.plot(ax=ax2, facecolor='none',
                 edgecolor='black', hatch='xxxx',
                 linewidth=0, zorder=2)
pulog.to_crs(crs).boundary.plot(ax=ax2,
                                edgecolor='black', linewidth=2, zorder=3)

ax2.set_title('Sebaran Suhu Permukaan (LST) – Kelurahan Pulogebang', fontsize=
ax2.set_xlabel('Koordinat Timur (meter)', fontsize=16)
ax2.set_ylabel('Koordinat Utara (meter)', fontsize=16)
cbar2 = plt.colorbar(im2, ax=ax2, fraction=0.046, pad=0.04)
cbar2.set_label('Suhu Permukaan Lahan (°C)', fontsize=16)
cbar2.ax.tick_params(labelsize=14)
ax2.tick_params(axis='both', which='major', labelsize=14)

# 8. Tampilkan output
plt.tight_layout()
plt.show()

print("Peta berhasil ditampilkan.")
```




```
Peta berhasil ditampilkan.
```

Klasifikasi Peta

In [3]:

```python
# Import library yang dibutuhkan
import os
import geopandas as gpd
import pandas as pd
import numpy as np
import rasterio
import matplotlib.pyplot as plt
from shapely.geometry import box

# Coba import rasterstats, jika belum ada, berikan pesan untuk menginstalnya
try:
    from rasterstats import zonal_stats
except ImportError:
    print("="*60)
    print("ERROR: Library 'rasterstats' tidak ditemukan.")
    print("Silakan install terlebih dahulu dengan menjalankan perintah ini di
    print("!pip install rasterstats")
    print("="*60)
    raise

print("Library berhasil di-import.")

# --- 1. Paths (Sesuaikan jika perlu) ---
work_dir = '/Users/admin/Documents/project pulogebang/proyek_pulogebang'
lst_path = os.path.join(work_dir, 'LST.TIF')
gadm_geojson = os.path.expanduser('~/Downloads/gadm41_IDN_4.json')

# --- 2. Load & filter Pulogebang ---
gdf = gpd.read_file(gadm_geojson)
pulog = gdf[gdf['NAME_4'].str.lower() == 'pulogebang']
if pulog.empty:
    raise ValueError("Kelurahan 'Pulogebang' tidak ditemukan.")
```

```python
# Samakan CRS Pulogebang dengan raster
with rasterio.open(lst_path) as src:
    raster_crs = src.crs
pulog_reproj = pulog.to_crs(raster_crs)
print("Batas wilayah Pulogebang berhasil dibaca dan disiapkan.")

# --- 3. Buat Grid 300x300 meter ---
xmin, ymin, xmax, ymax = pulog_reproj.total_bounds
grid_size = 300
cols = list(np.arange(xmin, xmax + grid_size, grid_size))
rows = list(np.arange(ymin, ymax + grid_size, grid_size))
polygons = [box(cols[x], rows[y], cols[x+1], rows[y+1]) for x in range(len(col
grid = gpd.GeoDataFrame({'geometry': polygons}, crs=raster_crs)
grid_clipped = gpd.overlay(grid, pulog_reproj, how='intersection')
grid_clipped['cell_id'] = range(len(grid_clipped))
print(f"Grid 300x300m berhasil dibuat dengan total {len(grid_clipped)} sel val

# --- 4. Analisis Zonal Statistik per Grid ---
stats_lst = zonal_stats(grid_clipped, lst_path, stats="mean", all_touched=True
print("Analisis zonal statistik per grid selesai.")

# --- 5. Buat DataFrame Hasil dan Hitung STL ---
df = pd.DataFrame(stats_lst)
df = df.rename(columns={'mean': 'lst_mean'})
df['cell_id'] = grid_clipped['cell_id']
df = df.dropna()

# Hitung Beban Termal Terstandarisasi (STL) menggunakan z-score
# Ini adalah metrik yang unggul dan sesuai dengan naskah Anda
mu_lst = df['lst_mean'].mean()
sigma_lst = df['lst_mean'].std()
df['stl'] = (df['lst_mean'] - mu_lst) / sigma_lst
print("Perhitungan 'Beban Termal Terstandarisasi (STL)' selesai.")

# --- 6. Klasifikasi Zona ---

# METODE 1: Klasifikasi berdasarkan LST Rata-rata (Metode Kuartil)
q1_lst = df['lst_mean'].quantile(0.25)
q3_lst = df['lst_mean'].quantile(0.75)
def classify_lst_quartile(lst):
    if lst <= q1_lst: return 'Aman'
    elif lst > q1_lst and lst <= q3_lst: return 'Waspada'
    else: return 'Kritis'
df['zona_lst_quartile'] = df['lst_mean'].apply(classify_lst_quartile)
print("Klasifikasi untuk LST Rata-Rata (Kuartil) selesai.")

# ============================================================================
# PERUBAHAN KUNCI: METODE KLASIFIKASI BARU UNTUK STL
# ============================================================================
# METODE 2: Klasifikasi berdasarkan Ambang Batas Statistik STL
# Metode ini akan menghasilkan peta yang berbeda dan lebih bermakna secara ilm
# Ia mengklasifikasikan sel berdasarkan seberapa signifikan deviasinya dari ra
```

```python
def classify_stl_threshold(stl_value):
    # Kritis: Sel yang suhunya lebih dari 1.5 standar deviasi di atas rata-rat
    if stl_value > 1.5:
        return 'Kritis'
    # Waspada: Sel yang suhunya di atas rata-rata, tetapi tidak seekstrem zona
    elif stl_value > 0 and stl_value <= 1.5:
        return 'Waspada'
    # Aman: Sel yang suhunya di bawah atau sama dengan rata-rata
    else:
        return 'Aman'

df['zona_stl_threshold'] = df['stl'].apply(classify_stl_threshold)
print("Klasifikasi baru berdasarkan Ambang Batas Statistik STL selesai.")
# ================================================================================

# Gabungkan hasil klasifikasi dengan data geospasial grid
grid_classified = grid_clipped.merge(df, on='cell_id')

# --- 7. Tampilkan Tabel Ringkasan ---
print("\n--- RINGKASAN KLASIFIKASI BERDASARKAN LST RATA-RATA (Kuartil) ---")
print(df.groupby('zona_lst_quartile')['lst_mean'].agg(['count', 'mean']))

print("\n--- RINGKASAN KLASIFIKASI BERDASARKAN AMBANG BATAS STATISTIK STL ---"
print(df.groupby('zona_stl_threshold')['stl'].agg(['count', 'mean']))

# --- 8. Visualisasi Perbandingan Peta ---
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(24, 12))
color_map = {'Aman': 'blue', 'Waspada': 'yellow', 'Kritis': 'red'}

# PETA 1: Berdasarkan LST Rata-rata (Metode Konvensional)
grid_classified.plot(column='zona_lst_quartile', ax=ax1, legend=True,
                     cmap=plt.matplotlib.colors.ListedColormap(color_map.value
                     categorical=True,
                     legend_kwds={'title': 'Zona Termal (LST - Kuartil)', 'loc
pulog_reproj.boundary.plot(ax=ax1, color='black', linewidth=1)
ax1.set_title('Peta Klasifikasi Berdasarkan LST Rata-Rata', fontsize=16)
ax1.set_xlabel('Koordinat Timur (meter)')
ax1.set_ylabel('Koordinat Utara (meter)')

# PETA 2: Berdasarkan STL (Ambang Batas)
grid_classified.plot(column='zona_stl_threshold', ax=ax2, legend=True,
                     cmap=plt.matplotlib.colors.ListedColormap(color_map.value
                     categorical=True,
                     legend_kwds={'title': 'Zona Termal (STL - Ambang Batas)',
pulog_reproj.boundary.plot(ax=ax2, color='black', linewidth=1)
ax2.set_title('Peta Klasifikasi Berdasarkan Ambang Batas Statistik STL', fonts
ax2.set_xlabel('Koordinat Timur (meter)')
ax2.set_ylabel('')

# Simpan otomatis ke Downloads
output_path = os.path.expanduser("~/Downloads/peta_klasifikasi_LST_vs_STL.png"
plt.savefig(output_path, dpi=300, bbox_inches='tight')
```

```
plt.tight_layout()
plt.show()

print(f"\nAnalisis dan visualisasi selesai. Gambar disimpan di: {output_path}"
```

```
Library berhasil di-import.
Batas wilayah Pulogebang berhasil dibaca dan disiapkan.
Grid 300x300m berhasil dibuat dengan total 98 sel valid.
Analisis zonal statistik per grid selesai.
Perhitungan 'Beban Termal Terstandarisasi (STL)' selesai.
Klasifikasi untuk LST Rata-Rata (Kuartil) selesai.
Klasifikasi baru berdasarkan Ambang Batas Statistik STL selesai.

--- RINGKASAN KLASIFIKASI BERDASARKAN LST RATA-RATA (Kuartil) ---
                    count       mean
zona_lst_quartile
Aman                   25  39.799679
Kritis                 24  43.135223
Waspada                48  41.640362

--- RINGKASAN KLASIFIKASI BERDASARKAN AMBANG BATAS STATISTIK STL ---
                     count      mean
zona_stl_threshold
Aman                    47 -0.807614
Kritis                   5  1.755219
Waspada                 45  0.648484
```


```
Analisis dan visualisasi selesai. Gambar disimpan di: /Users/admin/Downloads/peta_klasifikasi_LST_vs_STL.png
```

Scatter Plot

tabel data

In [5]:
```python
# Import library yang dibutuhkan
import os
import numpy as np
import geopandas as gpd
import rasterio
from shapely.geometry import box
from rasterstats import zonal_stats
import pandas as pd

print("Library berhasil di-import.")

# 1. Tentukan path data (sesuaikan jika perlu)
work_dir = '/Users/admin/Documents/project pulogebang/proyek_pulogebang'
ndvi_tif = os.path.join(work_dir, 'NDVI.TIF')
lst_tif = os.path.join(work_dir, 'LST.TIF')
gadm_json = os.path.expanduser('~/Downloads/gadm41_IDN_4.json')

# 2. Baca dan filter Kelurahan Pulogebang
try:
    gdf = gpd.read_file(gadm_json)
    gdf = gdf[gdf['NAME_4'].str.lower() == 'pulogebang']
    if gdf.empty:
        raise ValueError("Kelurahan 'Pulogebang' tidak ditemukan.")

    with rasterio.open(ndvi_tif) as src:
        raster_crs = src.crs
    pulog = gdf.to_crs(raster_crs)
    print("Batas wilayah Pulogebang berhasil dibaca dan disiapkan.")

    # 3. Buat grid 300x300 m di area Pulogebang
    grid_size = 300
    minx, miny, maxx, maxy = pulog.total_bounds
    xs = np.arange(minx, maxx + grid_size, grid_size)
    ys = np.arange(miny, maxy + grid_size, grid_size)

    cells = []
    for x in xs[:-1]:
        for y in ys[:-1]:
            cells.append(box(x, y, x + grid_size, y + grid_size))

    grid = gpd.GeoDataFrame({'geometry': cells}, crs=raster_crs)
    # PERBAIKAN: Mengganti unary_union yang usang dengan union_all()
    mask = grid.intersects(pulog.union_all())

    grid = grid[mask].reset_index(drop=True)
    grid['cell_id'] = grid.index
```

```
    # 4. Hitung zonal stats (mean NDVI & LST) per grid cell
    zs_ndvi = zonal_stats(grid, ndvi_tif, stats=['mean'])
    zs_lst = zonal_stats(grid, lst_tif, stats=['mean'])

    # 5. Kumpulkan hasil ke DataFrame
    records = []
    for idx, (nd, ls) in enumerate(zip(zs_ndvi, zs_lst)):
        records.append({
            'cell_id': grid.loc[idx, 'cell_id'],
            'ndvi_mean': nd.get('mean', np.nan),
            'lst_mean': ls.get('mean', np.nan)
        })

    df_cells = pd.DataFrame(records).dropna().reset_index(drop=True)

    # --- TAMPILKAN HASIL ---
    print("\n" + "="*40)
    print(f"JUMLAH TOTAL GRID SEL YANG DIANALISIS: {len(df_cells)}")
    print("="*40)

    print("\n--- TABEL DATA NDVI DAN LST RATA-RATA PER GRID ---")
    # Atur agar Pandas menampilkan semua baris
    pd.set_option('display.max_rows', None)
    print(df_cells)

except Exception as e:
    print(f"Terjadi error: {e}")
print(df_cells.describe())
```

```
Library berhasil di-import.
Batas wilayah Pulogebang berhasil dibaca dan disiapkan.

========================================
JUMLAH TOTAL GRID SEL YANG DIANALISIS: 97
========================================

--- TABEL DATA NDVI DAN LST RATA-RATA PER GRID ---
    cell_id  ndvi_mean   lst_mean
0         0   0.097376  43.149844
1         1   0.109588  41.993047
2         2   0.119811  42.208862
3         3   0.127947  41.487178
4         4   0.107018  42.397642
5         5   0.144600  41.838008
6         6   0.127351  42.514536
7         7   0.170429  42.045820
8         8   0.131596  41.738989
9         9   0.116728  42.291504
10       10   0.118037  42.108301
11       11   0.117272  42.591851
12       12   0.110879  42.526328
13       13   0.091829  41.985151
14       14   0.086610  44.104321
15       15   0.121198  42.392754
16       16   0.087107  43.797930
17       17   0.144928  41.776006
18       18   0.157875  40.341187
19       19   0.141912  41.555645
20       20   0.153365  40.642383
21       21   0.120428  41.179595
22       22   0.116388  42.870420
23       23   0.103081  43.372559
24       24   0.089130  44.179624
25       25   0.151281  41.628281
26       26   0.122345  42.482202
27       27   0.094737  44.035967
28       28   0.144569  42.162031
29       29   0.138929  42.217886
30       30   0.162706  40.581646
31       31   0.162333  39.182820
32       32   0.100625  41.217227
33       33   0.109806  42.816792
34       34   0.114892  43.279453
35       35   0.155450  41.728838
36       36   0.133951  40.949285
37       37   0.161111  40.896172
38       38   0.182908  40.817280
39       39   0.172269  40.274019
40       40   0.149423  41.129180
41       41   0.132689  41.315869
42       42   0.111162  41.100024
43       43   0.106165  42.447407
44       44   0.104926  43.395630
```

```
45       45   0.149152  41.462739
46       46   0.133075  41.988501
47       47   0.132484  40.973896
48       48   0.170072  41.393389
49       49   0.165167  40.441914
50       50   0.136607  41.078149
51       51   0.109589  41.659722
52       52   0.133636  41.151123
53       53   0.106859  42.268203
54       54   0.095718  42.795186
55       55   0.150066  40.907004
56       56   0.185758  39.984482
57       57   0.288719  37.044268
58       58   0.181986  39.875686
59       59   0.110413  41.469644
60       60   0.145453  41.021406
61       61   0.108062  41.043081
62       62   0.092277  41.076885
63       63   0.080948  42.683110
64       64   0.114589  41.912280
65       65   0.167616  40.827502
66       66   0.153737  41.482461
67       67   0.137105  40.747793
68       68   0.127059  40.730942
69       69   0.095654  41.947827
70       70   0.141582  40.839702
71       71   0.092885  42.899707
72       72   0.112520  42.678330
73       73   0.164545  40.809185
74       74   0.109618  42.902446
75       75   0.107841  43.283760
76       76   0.114906  42.209375
77       77   0.214961  38.825295
78       78   0.271386  38.605005
79       79   0.163818  41.905923
80       80   0.095542  43.289434
81       81   0.149408  41.641914
82       82   0.139014  41.345742
83       83   0.073715  43.940464
84       84   0.095928  42.975459
85       85   0.099288  43.057386
86       86   0.153209  39.524526
87       87   0.197597  38.984131
88       88   0.181297  40.407432
89       89   0.144629  41.954253
90       90   0.145270  41.530996
91       91   0.226754  37.044336
92       92   0.066670  39.025439
93       94   0.060144  40.626078
94       95   0.142987  39.282251
95       96   0.269143  38.034604
96       97   0.186678  41.154575
         cell_id  ndvi_mean   lst_mean
count  97.000000  97.000000  97.000000
```

```
mean   48.041237   0.135236  41.520541
std    28.214918   0.040747   1.414877
min     0.000000   0.060144  37.044268
25%    24.000000   0.108062  40.839702
50%    48.000000   0.132484  41.628281
75%    72.000000   0.153365  42.447407
max    97.000000   0.288719  44.179624
```

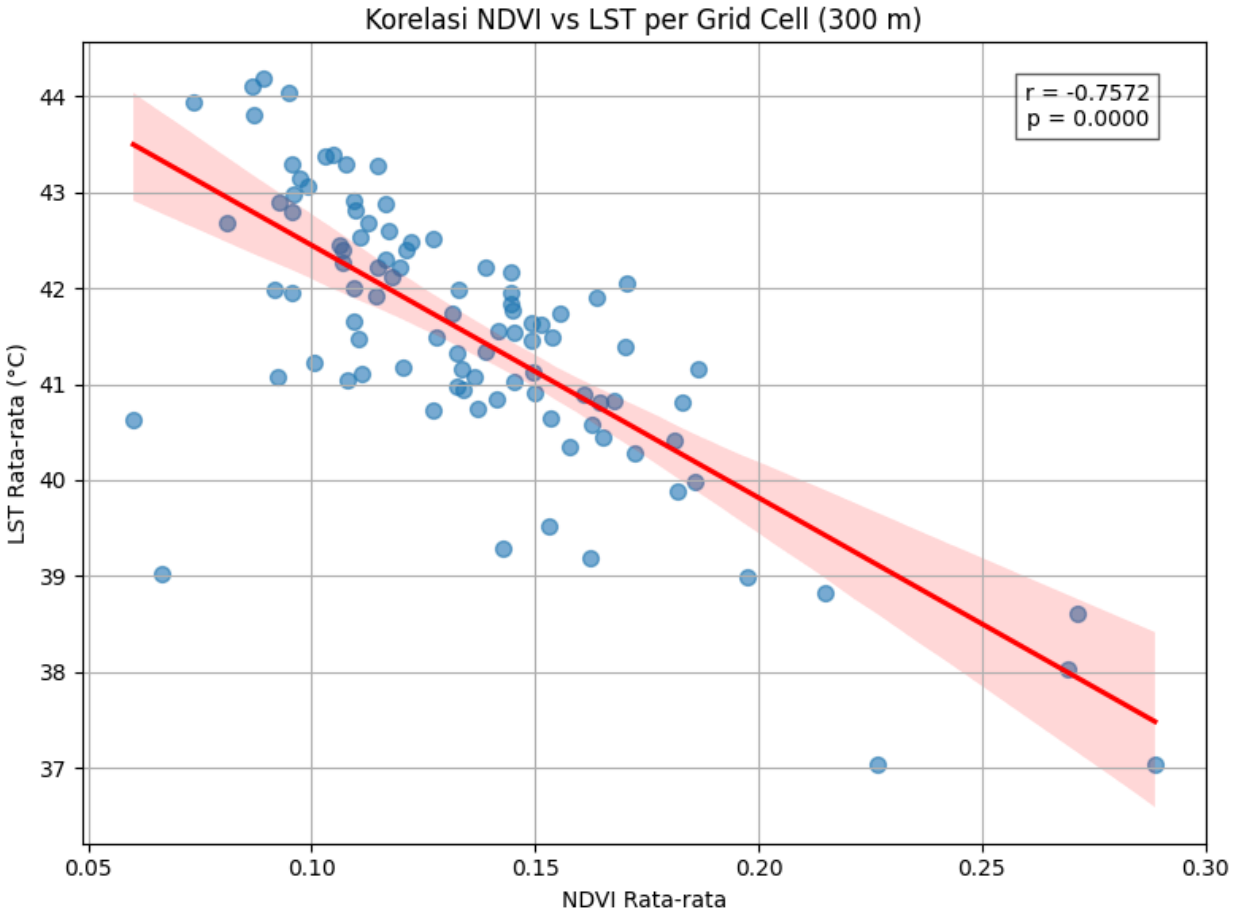
Grafik

In [6]:

```python
import os
import numpy as np
import geopandas as gpd
import rasterio
from shapely.geometry import box
from rasterstats import zonal_stats
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import pearsonr

# 1. Tentukan path data
work_dir   = '/Users/admin/Documents/project pulogebang/proyek_pulogebang'
ndvi_tif   = os.path.join(work_dir, 'NDVI.TIF')
lst_tif    = os.path.join(work_dir, 'LST.TIF')
gadm_json  = os.path.expanduser('~/Downloads/gadm41_IDN_4.json')

# 2. Baca dan filter Kelurahan Pulogebang
gdf = gpd.read_file(gadm_json)
gdf = gdf[gdf['NAME_4'].str.lower() == 'pulogebang']
if gdf.empty:
    raise ValueError("Kelurahan 'Pulogebang' tidak ditemukan di GADM GeoJSON."

with rasterio.open(ndvi_tif) as src:
    raster_crs = src.crs
pulog = gdf.to_crs(raster_crs)

# 3. Buat grid 300×300 m di area Pulogebang
grid_size = 300
minx, miny, maxx, maxy = pulog.total_bounds
xs = np.arange(minx, maxx + grid_size, grid_size)
ys = np.arange(miny, maxy + grid_size, grid_size)

cells = []
for x in xs[:-1]:
    for y in ys[:-1]:
        cells.append(box(x, y, x + grid_size, y + grid_size))

grid = gpd.GeoDataFrame({'geometry': cells}, crs=raster_crs)
mask = grid.intersects(pulog.union_all())  # ganti unary_union ke union_all
grid = grid[mask].reset_index(drop=True)
grid['cell_id'] = grid.index
```

```
print(f"Total grid cells in Pulogebang: {len(grid)}")

# 4. Hitung zonal stats (mean NDVI & LST) per grid cell
zs_ndvi = zonal_stats(grid, ndvi_tif, stats=['mean'])
zs_lst  = zonal_stats(grid, lst_tif,  stats=['mean'])

# 5. Kumpulkan hasil ke DataFrame
records = []
for idx, (nd, ls) in enumerate(zip(zs_ndvi, zs_lst)):
    records.append({
        'cell_id'   : grid.loc[idx, 'cell_id'],
        'ndvi_mean' : nd.get('mean', np.nan),
        'lst_mean'  : ls.get('mean', np.nan)
    })

df_cells = pd.DataFrame(records).dropna().reset_index(drop=True)
print(f"\nTotal grid cells processed: {len(df_cells)}")
print(df_cells.head())

# 6. Analisis korelasi dan visualisasi
r, p = pearsonr(df_cells['ndvi_mean'], df_cells['lst_mean'])
print(f"\nPearson r = {r:.4f}, p-value = {p:.4f}")
plt.figure(figsize=(8,6))
sns.regplot(
    x='ndvi_mean', y='lst_mean',
    data=df_cells,
    scatter_kws={'s':50,'alpha':0.6},
    line_kws={'color':'red'}
)
plt.title('Korelasi NDVI vs LST per Grid Cell (300 m)')
plt.xlabel('NDVI Rata-rata')
plt.ylabel('LST Rata-rata (°C)')
# Memindahkan label ke pojok kanan atas
plt.text(
    0.95, 0.95,  # Ubah posisi x dan y untuk memindahkan ke pojok kanan atas
    f"r = {r:.4f}\np = {p:.4f}",
    transform=plt.gca().transAxes,
    ha='right',  # Horizontal alignment ke kanan
    va='top',    # Vertical alignment ke atas
    bbox=dict(facecolor='white', alpha=0.6)
)
plt.grid(True)
plt.tight_layout()
# Simpan gambar ke folder Downloads
output_path = os.path.expanduser('~/Downloads/korelasi_ndvi_lst.png')
plt.savefig(output_path, dpi=300)  # dpi bisa disesuaikan untuk kualitas gamba
print(f"Gambar disimpan di: {output_path}")

plt.show()
```

```
Total grid cells in Pulogebang: 98

Total grid cells processed: 97
   cell_id  ndvi_mean   lst_mean
0        0   0.097376  43.149844
1        1   0.109588  41.993047
2        2   0.119811  42.208862
3        3   0.127947  41.487178
4        4   0.107018  42.397642

Pearson r = -0.7572, p-value = 0.0000
Gambar disimpan di: /Users/admin/Downloads/korelasi_ndvi_lst.png
```



In [ ]: