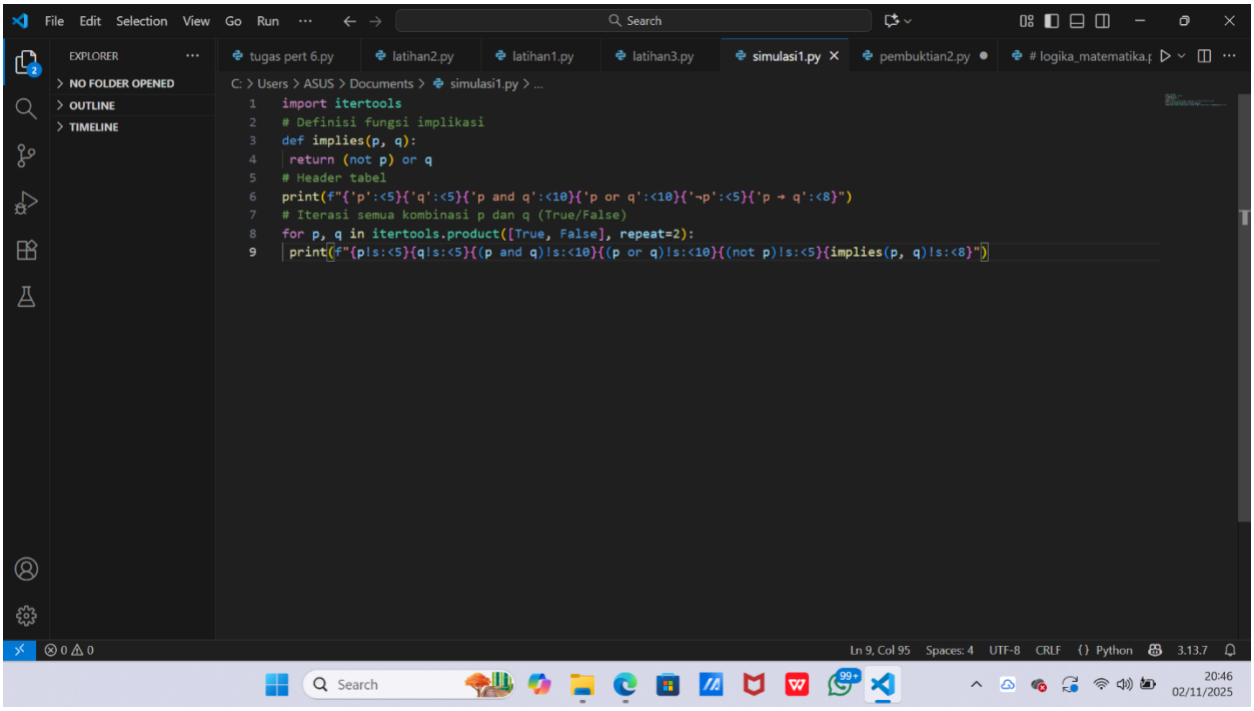
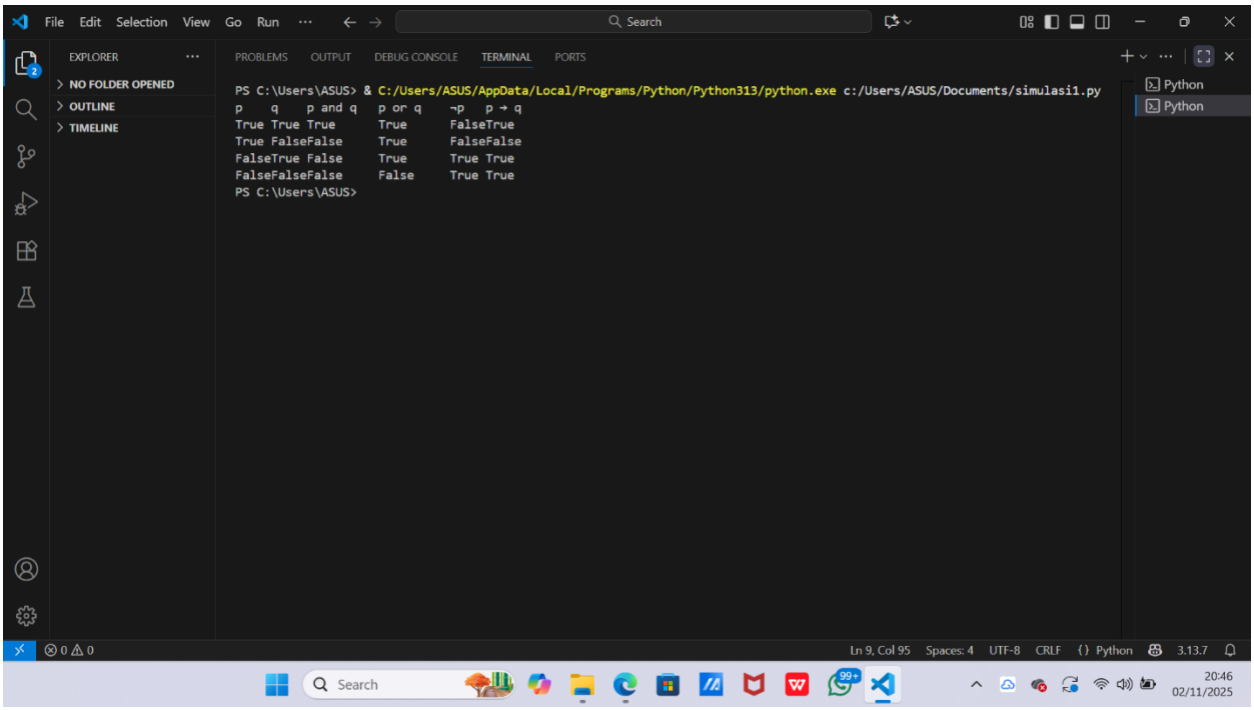
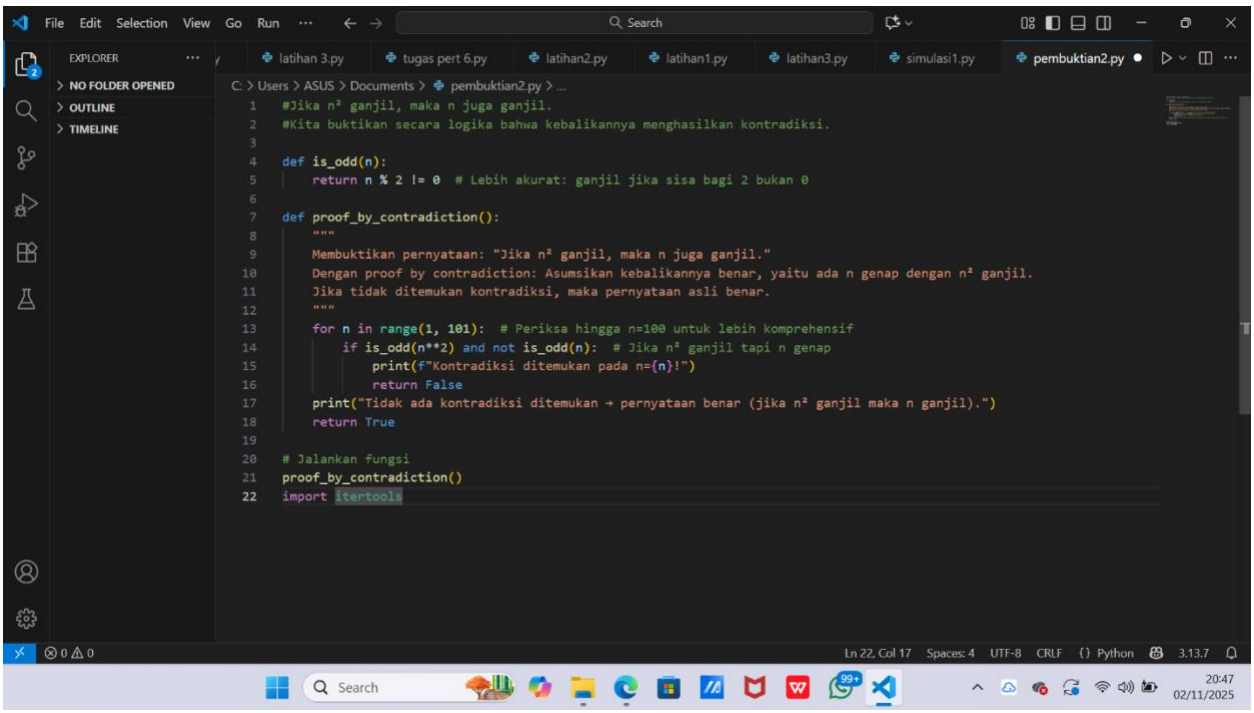
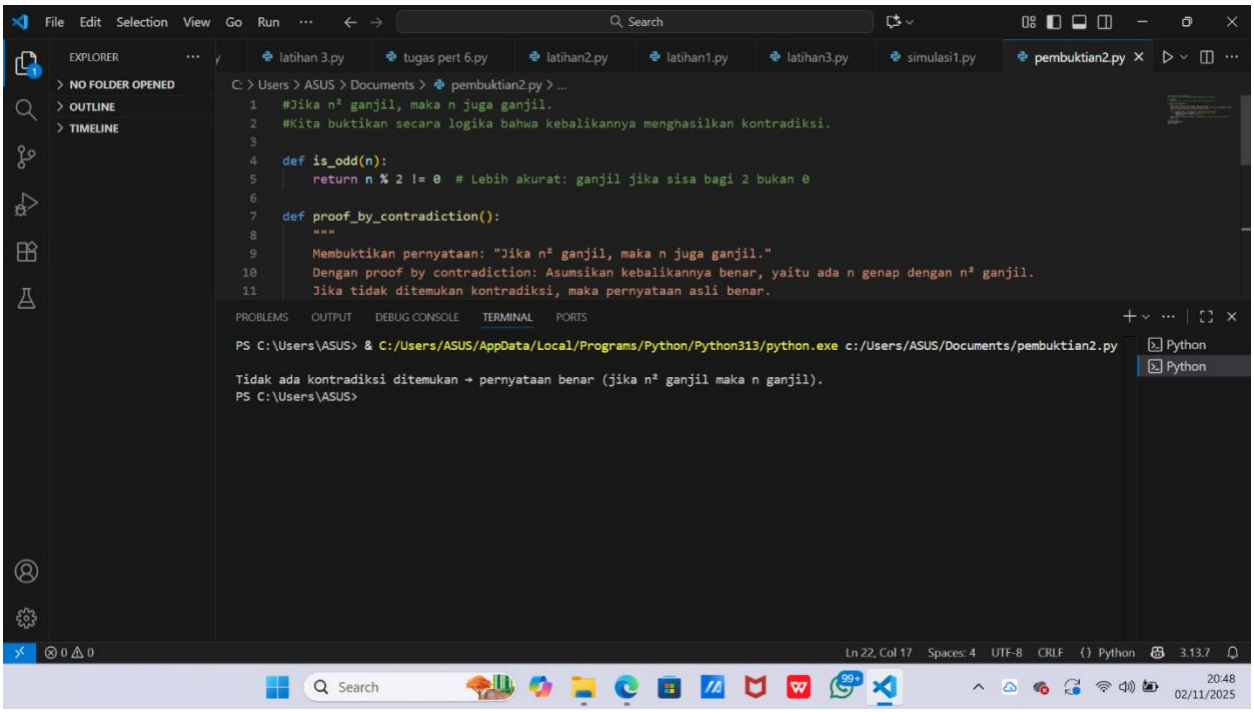
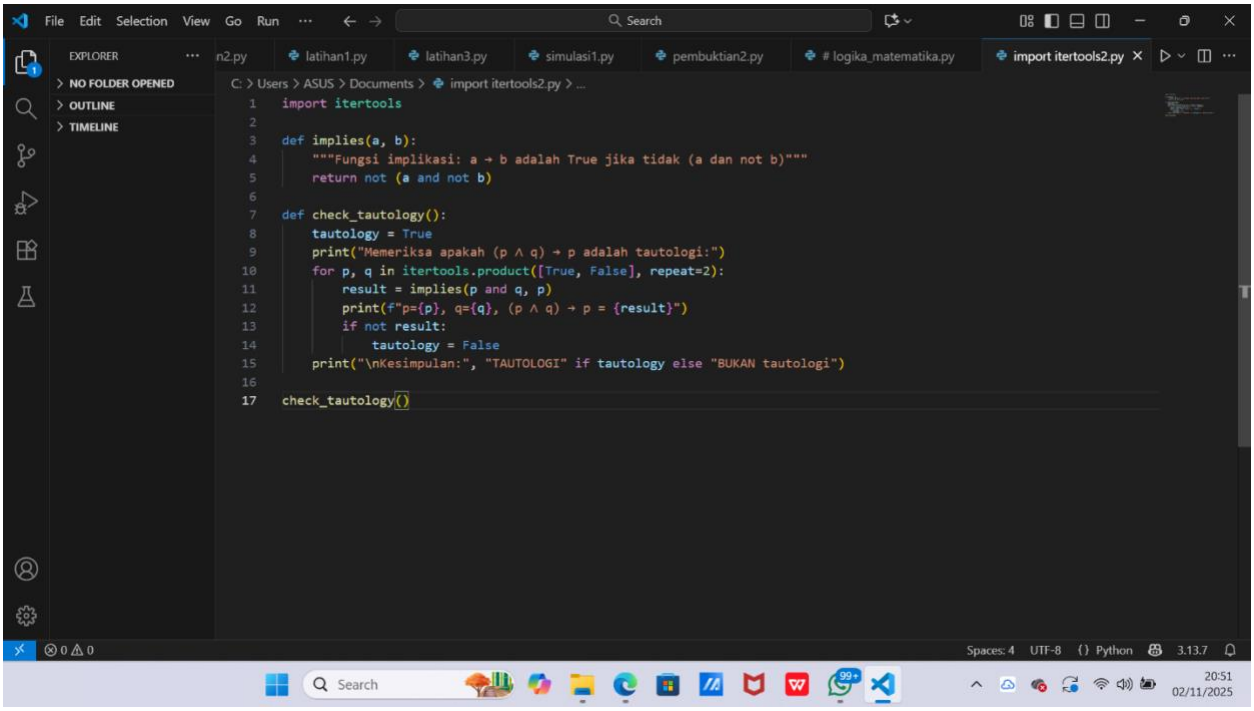
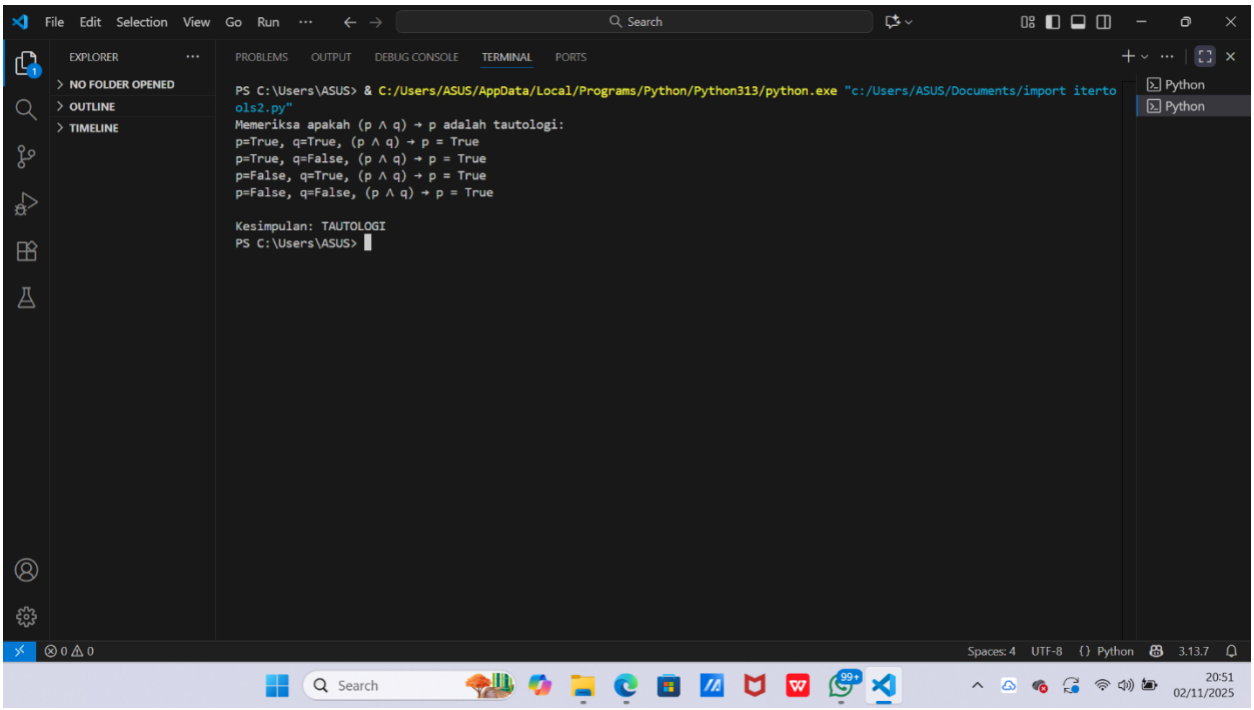

File Edit Selection View Go Run
Search
EXPLORER
NO FOLDER OPENED
OUTLINE
TIMELINE
tugas pert 6.py
latihan2.py
latihan1.py
latihan3.py
simulasi1.py
pembuktian2.py
logika_matematika.p
C: > Users > ASUS > Documents > simulasi1.py > ...
1 import itertools
2 # Definisi fungsi implikasi
3 def implies(p, q):
4 return (not p) or q
5 # Header tabel
6 print(f"{'p':<5}{'q':<5}{'p and q':<10}{'p or q':<10}{'¬p':<5}{'p → q':<8}")
7 # Iterasi semua kombinasi p dan q (True/False)
8 for p, q in itertools.product([True, False], repeat=2):
9 print(f"{p!s:<5}{q!s:<5}{(p and q)!s:<10}{(p or q)!s:<10}{(not p)!s:<5}{implies(p, q)!s:<8}")
Ln 9, Col 95 Spaces: 4 UTF-8 CRLF Python 3.13.7
Search
20:46
02/11/2025


File Edit Selection View Go Run
Search
EXPLORER
NO FOLDER OPENED
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ASUS> & C:/Users/ASUS/AppData/Local/Programs/Python/Python313/python.exe c:/Users/ASUS/Documents/simulasi1.py
p q p and q p or q ¬p p → q
True True True True FalseTrue
True FalseFalse True FalseFalse
FalseTrue False True True True
FalseFalseFalse False True True
PS C:\Users\ASUS>
Python
Python
Ln 9, Col 95 Spaces: 4 UTF-8 CRLF Python 3.13.7
Search
20:46
02/11/2025


File Edit Selection View Go Run
Search
EXPLORER
NO FOLDER OPENED
OUTLINE
TIMELINE
latihan 3.py
tugas pert 6.py
latihan2.py
latihan1.py
latihan3.py
simulasi1.py
pembuktian2.py
C: > Users > ASUS > Documents > pembuktian2.py > ...
#Jika n² ganjil, maka n juga ganjil.
#Kita buktikan secara logika bahwa kebalikannya menghasilkan kontradiksi.
def is_odd(n):
return n % 2 != 0 # Lebih akurat: ganjil jika sisa bagi 2 bukan 0
def proof_by_contradiction():
"""
Membuktikan pernyataan: "Jika n² ganjil, maka n juga ganjil."
Dengan proof by contradiction: Asumsikan kebalikannya benar, yaitu ada n genap dengan n² ganjil.
Jika tidak ditemukan kontradiksi, maka pernyataan asli benar.
"""
for n in range(1, 101): # Periksa hingga n=100 untuk lebih komprehensif
if is_odd(n**2) and not is_odd(n): # Jika n² ganjil tapi n genap
print(f"Kontradiksi ditemukan pada n={n}!")
return False
print("Tidak ada kontradiksi ditemukan → pernyataan benar (jika n² ganjil maka n ganjil).")
return True
# Jalankan fungsi
proof_by_contradiction()
import itertools
Ln 22, Col 17 Spaces: 4 UTF-8 CRLF Python 3.13.7
Search
20:47
02/11/2025


File Edit Selection View Go Run
Search
EXPLORER
NO FOLDER OPENED
OUTLINE
TIMELINE
latihan 3.py
tugas pert 6.py
latihan2.py
latihan1.py
latihan3.py
simulasi1.py
pembuktian2.py
C: > Users > ASUS > Documents > pembuktian2.py > ...
#Jika n² ganjil, maka n juga ganjil.
#Kita buktikan secara logika bahwa kebalikannya menghasilkan kontradiksi.
def is_odd(n):
return n % 2 != 0 # Lebih akurat: ganjil jika sisa bagi 2 bukan 0
def proof_by_contradiction():
"""
Membuktikan pernyataan: "Jika n² ganjil, maka n juga ganjil."
Dengan proof by contradiction: Asumsikan kebalikannya benar, yaitu ada n genap dengan n² ganjil.
Jika tidak ditemukan kontradiksi, maka pernyataan asli benar.
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ASUS> & C:/Users/ASUS/AppData/Local/Programs/Python/Python313/python.exe c:/Users/ASUS/Documents/pembuktian2.py
Tidak ada kontradiksi ditemukan → pernyataan benar (jika n² ganjil maka n ganjil).
PS C:\Users\ASUS>
Python
Python
Ln 22, Col 17 Spaces: 4 UTF-8 CRLF Python 3.13.7
Search
20:48
02/11/2025


File Edit Selection View Go Run
Search
EXPLORER
NO FOLDER OPENED
OUTLINE
TIMELINE
n2.py
latihan1.py
latihan3.py
simulasi1.py
pembuktian2.py
# logika_matematika.py
import itertools2.py
C: > Users > ASUS > Documents > import itertools2.py > ...
import itertools
def implies(a, b):
"""Fungsi implikasi: a → b adalah True jika tidak (a dan not b)"""
return not (a and not b)
def check_tautology():
tautology = True
print("Memeriksa apakah (p ∧ q) → p adalah tautologi:")
for p, q in itertools.product([True, False], repeat=2):
result = implies(p and q, p)
print(f"p={p}, q={q}, (p ∧ q) → p = {result}")
if not result:
tautology = False
print("\nKesimpulan:", "TAUTOLOGI" if tautology else "BUKAN tautologi")
check_tautology()
Spaces: 4 UTF-8 Python 3.13.7
Search
20:51
02/11/2025


File Edit Selection View Go Run
Search
EXPLORER
NO FOLDER OPENED
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\ASUS> & C:/Users/ASUS/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/ASUS/Documents/import itertools2.py"
Memeriksa apakah (p ∧ q) → p adalah tautologi:
p=True, q=True, (p ∧ q) → p = True
p=True, q=False, (p ∧ q) → p = True
p=False, q=True, (p ∧ q) → p = True
p=False, q=False, (p ∧ q) → p = True
Kesimpulan: TAUTOLOGI
PS C:\Users\ASUS>
Python
Python
Spaces: 4 UTF-8 Python 3.13.7
Search
20:51
02/11/2025