

The screenshot shows the Visual Studio Code editor with a Python file named `simulasi1.py` open. The code defines a function `implies(p, q)` and uses `itertools.product` to iterate through all combinations of `p` and `q` (True/False). The output is printed in a formatted table.

```
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":<5}{"q":<5}{"(p and q)":<10}{"(p or q)":<10}{"(not p)":<5}{"implies(p, q)":<8}"')
```

The screenshot shows the terminal window in VS Code, displaying the output of the Python script. The output is a table showing the results of the logic simulation for all combinations of `p` and `q`.

```
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 False True
True False False True False False
False True False True True True
False False False False True True
PS C:\Users\ASUS>
```

The screenshot shows the Visual Studio Code editor with a Python file named `pembuktian2.py` open. The file contains a script to prove that if  $n^2$  is odd, then  $n$  is also odd, using a proof by contradiction. The script defines a function `is_odd(n)` and a function `proof_by_contradiction()`. The `proof_by_contradiction()` function iterates through numbers from 1 to 101, checking if  $n^2$  is odd while  $n$  is even. If such a case is found, it prints a contradiction. If not, it prints that the statement is true.

```
1 #Jika n² ganjil, maka n juga ganjil.
2 #Kita buktikan secara logika bahwa kebalikannya menghasilkan kontradiksi.
3
4 def is_odd(n):
5     return n % 2 != 0 # Lebih akurat: ganjil jika sisa bagi 2 bukan 0
6
7 def proof_by_contradiction():
8     """
9     Membuktikan pernyataan: "Jika n² ganjil, maka n juga ganjil."
10    Dengan proof by contradiction: Asumsikan kebalikannya benar, yaitu ada n genap dengan n² ganjil.
11    Jika tidak ditemukan kontradiksi, maka pernyataan asli benar.
12    """
13    for n in range(1, 101): # Periksa hingga n=100 untuk lebih komprehensif
14        if is_odd(n**2) and not is_odd(n): # Jika n² ganjil tapi n genap
15            print(f"Kontradiksi ditemukan pada n={n}!")
16            return False
17    print("Tidak ada kontradiksi ditemukan → pernyataan benar (jika n² ganjil maka n ganjil).")
18    return True
19
20 # Jalankan fungsi
21 proof_by_contradiction()
22 import itertools
```

The screenshot shows the same Visual Studio Code editor, but now the `TERMINAL` panel is active, displaying the output of the script. The output confirms that no contradiction was found, supporting the statement that if  $n^2$  is odd, then  $n$  is odd.

```
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>
```

The screenshot shows a Visual Studio Code editor window with a Python file named `import itertools2.py` open. The code defines a function `implies(a, b)` that returns `not (a and not b)`. It also defines a function `check_tautology()` that uses `itertools.product` to iterate over all combinations of `p` and `q` (True/False) and checks if the implication `(p → q)` is always true. The output of the script is visible in the terminal panel below the editor.

```
1 import itertools
2
3 def implies(a, b):
4     """Fungsi implikasi: a → b adalah True jika tidak (a dan not b)"""
5     return not (a and not b)
6
7 def check_tautology():
8     tautology = True
9     print("Memeriksa apakah (p → q) → p adalah tautologi:")
10    for p, q in itertools.product([True, False], repeat=2):
11        result = implies(p and q, p)
12        print(f"p={p}, q={q}, (p → q) → p = {result}")
13        if not result:
14            tautology = False
15    print("\nKesimpulan:", "TAUTOLOGI" if tautology else "BUKAN tautologi")
16
17 check_tautology()
```

The terminal output shows the execution of the script, confirming that the expression is a tautology.

This screenshot shows the same VS Code editor window, but with the terminal panel active. The terminal displays the output of the Python script executed in the previous screenshot. It shows the results of the implication checks for all combinations of `p` and `q`, and concludes that the expression is a tautology.

```
PS C:\Users\ASUS> & C:/Users/ASUS/AppData/Local/Programs/Python/Python313/python.exe "c:/Users/ASUS/Documents/import iterto
ols2.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>
```