```
C:\Users\usuario\Anaconda3\envs\tfq\python.exe C:/Users/usuario/qGAN/quantumGAN/performance_testing/
performance_get_output_generator.py
[0.22607422 0.4921875  0.16552734 0.11621094]
Timer unit: 1e-07 s

Total time: 0.90647 s
File: C:/Users/usuario/qGAN/quantumGAN/performance_testing/performance_get_output_generator.py
Function: get_output_V1 at line 29

Line #      Hits         Time  Per Hit   % Time  Line Contents
==============================================================
    29                                           def get_output_V1():
    30         1         25.0     25.0      0.0      real_keys = {"00", "10", "01", "11"}
    31
    32         1        680.0    680.0      0.0      quantum = QuantumRegister(sum(num_qubits), name="q")
    33         1       1554.0   1554.0      0.0      qc = QuantumCircuit(sum(num_qubits))
    34
    35         1       1328.0   1328.0      0.0      init_dist = qiskit.QuantumCircuit(sum(num_qubits))
    36         1         33.0     33.0      0.0      assert latent_space_noise.shape[0] == sum(num_qubits)
    37
    38         3         46.0     15.3      0.0      for num_qubit in range(sum(num_qubits)):
    39         2       2600.0   1300.0      0.0          init_dist.ry(latent_space_noise[num_qubit], num_qubit)
    40
    41         1         83.0     83.0      0.0      params = cast(np.ndarray, parameter_values)
    42
    43         1     135702.0 135702.0      1.5      qc.append(construct_circuit(params), quantum)
    44         1      66560.0  66560.0      0.7      final_circuit = qc.compose(init_dist, front=True)
    45         1       3578.0   3578.0      0.0      final_circuit.measure_all()
    46
    47         1    4027225.0 4027225.0     44.4      simulator = qiskit.Aer.get_backend("aer_simulator")
    48         1    4709456.0 4709456.0     52.0      final_circuit = qiskit.transpile(final_circuit, simulator)
    49         1     106141.0 106141.0      1.2      result = simulator.run(final_circuit, shots=shots).result()
    50         1       1805.0   1805.0      0.0      counts = result.get_counts(final_circuit)
    51
    52         1         12.0     12.0      0.0      try:
    53         1        126.0    126.0      0.0          pixels = np.array([counts["00"], counts["10"], counts["01"],
counts["11"]])
    54
    55                                               except KeyError:
    56                                                   # dealing with the keys that qiskit doesn't include in the
    57                                                   # dictionary because they don't get any measurements
    58
    59                                                   keys = counts.keys()
    60                                                   missing_keys = real_keys.difference(keys)
    61                                                   # we use sets to get the missing keys
    62                                                   for key_missing in missing_keys:
    63                                                       counts[key_missing] = 0
    64
    65                                                   pixels = np.array([counts["00"], counts["10"], counts["01"],
counts["11"]])
    66
    67         1        186.0    186.0      0.0      pixels = pixels / shots
    68         1       7563.0   7563.0      0.1      print(pixels)


Process finished with exit code 0

```