```python
import numpy as np
import skfuzzy as fuzz

# The process of obtaining input has been explained in Supplementary_Material_2.pdf

# Antecedent/Consequent for water quality monitoring

# Functions
PH = ctrl.Antecedent(np.arange(0, 14, 1), 'PH')
DO = ctrl.Antecedent(np.arange(0, 12, 1), 'DO')
EC = ctrl.Antecedent(np.arange(0, 1100, 50), 'EC')
ORP = ctrl.Antecedent(np.arange(0, 40, 2.5), 'ORP')
Temperature = ctrl.Antecedent(np.arange(0, 10, 1), 'Temperature')
Water_Quality = ctrl.Consequent(np.arange(0, 10, 1), 'Water_Quality')

# Generate Fuzzy membership function for antecedent

PH['NA'] = fuzz.trimf(PH.universe, (0, 0, 5))
PH['HAC'] = fuzz.trimf(PH.universe, (2, 4, 6))
PH['HAC'] = fuzz.trimf(PH.universe, (6.5, 7.6, 8.7))

DO['NA'] = fuzz.trimf(DO.universe, (0, 0, 2))
DO['HAC'] = fuzz.trimf(DO.universe, (5.3, 6, 6.7))

EC['NA'] = fuzz.trimf(EC.universe, (0, 0, 0))
EC['HAC'] = fuzz.trimf(EC.universe, (290, 400, 510))

ORP['NA'] = fuzz.trimf(ORP.universe, (0, 0, 2))
ORP['HAC'] = fuzz.trimf(ORP.universe, (500, 600, 700))

Temperature['NA'] = fuzz.trimf(Temperature.universe, (0, 0, 2))
Temperature['HAC'] = fuzz.trimf(Temperature.universe, (1.9, 5, 10))

# Fuzzy rule generation (for convenience only one rule has been shown here, rest of the rules can be referred from supplementary material and can be implemented)

# Controls

Water_Quality['HAC'] = fuzz.trimf(Water_Quality.universe, (0, 0, 4))
```

```python
# You can see how these look with viewfis for example

# Fuzzy rule generation (for convenience only one rule has been shown here, rest of the rules can be referred from supplementary material and can be implemented)

# Controls

Water_Quality['HAC'] = fuzz.trimf(Water_Quality.universe, (0, 0, 4))
```