1. FRACTION (Pecahan)

```
class Fraction:
    def __init__(self, numerator=0, denominator=1):
        # Konstruktor sudah diberikan, tidak perlu diubah

    def __str__(self):
        return f"{self._numerator}/{self._denominator}"

    def __add__(self, other):
        new_numerator = self._numerator * other._denominator +
other._numerator * self._denominator
        new_denominator = self._denominator * other._denominator
        return Fraction(new_numerator, new_denominator)

    def __sub__(self, other):
        new_numerator = self._numerator * other._denominator -
other._numerator * self._denominator
        new_denominator = self._denominator * other._denominator
        return Fraction(new_numerator, new_denominator)

    def __mul__(self, other):
        new_numerator = self._numerator * other._numerator
        new_denominator = self._denominator * other._denominator
        return Fraction(new_numerator, new_denominator)

    def __truediv__(self, other):
        if other._numerator == 0:
            raise ZeroDivisionError("Tidak dapat membagi dengan nol")
        new_numerator = self._numerator * other._denominator
        new_denominator = self._denominator * other._numerator
        return Fraction(new_numerator, new_denominator)

    def __eq__(self, other):
        return self._numerator * other._denominator == other._numerator *
self._denominator

    def __float__(self):
        return self._numerator / self._denominator

    def __int__(self):
        return int(self._numerator / self._denominator)
```

2. CLASS VARIABLE (Variabel Kelas)

```
class BankAccount:
    _next_account_number = 1000

    def __init__(self, initial_balance=0):
        self._balance = initial_balance
        self._account_number = BankAccount._next_account_number
        BankAccount._next_account_number += 1
```

```
    def get_account_number(self):
        return self._account_number

    def deposit(self, amount):
        self._balance += amount

    def withdraw(self, amount):
        if amount <= self._balance:
            self._balance -= amount
        else:
            print("Saldo tidak mencukupi")

    def get_balance(self):
        return self._balance

# Membuat beberapa rekening dan menampilkan nomor rekeningnya
for i in range(5):
    account = BankAccount()
    print(f"Nomor rekening baru: {account.get_account_number()}")
```

3. FAMILY (Keluarga)

```
class Family:
    def __init__(self, father, mother, *children):
        self.father = father
        self.mother = mother
        self.children = list(children)

    def __iter__(self):
        return iter([self.father, self.mother] + self.children)

    def add_child(self, child):
        self.children.append(child)

    def __str__(self):
        return f"Keluarga: Ayah={self.father}, Ibu={self.mother}, Anak-
anak={', '.join(self.children)}"

# Contoh penggunaan
keluarga = Family("Budi", "Ani", "Cici", "Dodi")
print(keluarga)

# Menambahkan anak baru
keluarga.add_child("Edi")
print(keluarga)

# Iterasi melalui anggota keluarga
print("Anggota keluarga:")
for anggota in keluarga:
    print(anggota)
```