Python Cheat Sheet

Tanggal: 27 November 2025

1. Variabel dan Tipe Data

Python adalah bahasa dengan dynamic typing - Anda tidak perlu mendeklarasikan tipe variabel secara eksplisit.

Deklarasi Variabel

```
# String
nama = "Ahmad Budiman"
alamat = 'Jakarta Selatan'

# Integer
umur = 28
saldo = 1500000

# Float
bunga = 5.75
nilai_tukar = 15750.50

# Boolean
aktif = True
terverifikasi = False
```

Tipe Data Collection

```
# List (mutable, ordered)
rekening = ["1234567890", "0987654321"]
rekening.append("1111222233")

# Tuple (immutable, ordered)
koordinat = (106.8456, -6.2088)

# Dictionary (key-value pairs)
nasabah = {
    "nama": "Siti Rahayu",
    "no_rekening": "1234567890",
    "saldo": 5000000
}

# Set (unique values)
kategori = {"retail", "corporate", "priority"}
```

2. Operator

Operator Aritmatika

Operator	Deskripsi	Contoh
+	Penjumlahan	10 + 5 = 15
-	Pengurangan	10 - 5 = 5
*	Perkalian	10 * 5 = 50
/	Pembagian	10 / 3 = 3.33
//	Pembagian Bulat	10 // 3 = 3
%	Modulus	10 % 3 = 1
**	Pangkat	2 ** 10 = 1024

Operator Perbandingan

```
x == y    # Sama dengan
x != y    # Tidak sama dengan
x > y     # Lebih besar
x  y     # Lebih kecil
x >= y    # Lebih besar atau sama
x = y    # Lebih kecil atau sama
```

Operator Logika

```
and   # True jika kedua operand True
or    # True jika salah satu True
not   # Membalik nilai boolean

# Contoh
if saldo >= 1000000 and status == "aktif":
    print("Eligible")
```

3. Struktur Kontrol

Kondisional (if-elif-else)

```
saldo = 5000000

if saldo >= 10000000:
    tier = "Priority"
    bunga = 4.5
elif saldo >= 5000000:
    tier = "Gold"
    bunga = 3.5
elif saldo >= 1000000:
    tier = "Silver"
    bunga = 2.5
else:
    tier = "Basic"
    bunga = 1.5

print(f"Nasabah {tier}, bunga {bunga}%")
```

Perulangan For

```
# Iterasi list
transaksi = [50000, 100000, 75000]
total = 0
for nominal in transaksi:
    total += nominal
print(f"Total: Rp {total:,}")

# Range
for i in range(1, 6):
    print(f"Transaksi ke-{i}")

# Dictionary
for key, value in nasabah.items():
    print(f"{key}: {value}")
```

Perulangan While

```
attempt = 0
max_attempts = 3

while attempt  max_attempts:
    pin = input("Masukkan PIN: ")
    if verify_pin(pin):
        print("PIN benar!")
        break
    attempt += 1
else:
    print("Akun diblokir")
```

4. Fungsi

Definisi Fungsi Dasar

```
def hitung_bunga(pokok, rate, bulan):
    """
    Menghitung bunga deposito.

    Args:
        pokok: Jumlah pokok (Rp)
        rate: Suku bunga (%/tahun)
        bulan: Jangka waktu

    Returns:
        Jumlah bunga
    """
    bunga_bulanan = rate / 100 / 12
    return pokok * bunga_bulanan * bulan

# Penggunaan
bunga = hitung_bunga(100000000, 5.5, 12)
print(f"Bunga: Rp {bunga:,.0f}")
```

Lambda Functions

```
# Format currency
format_rupiah = lambda x: f"Rp {x:,.0f}"

# Sorting dengan key
transaksi = [
    {"tanggal": "2024-01-15", "nominal": 500000},
    {"tanggal": "2024-01-10", "nominal": 750000},
]

sorted_trx = sorted(
    transaksi,
    key=lambda x: x["nominal"],
    reverse=True
)
```

5. String dan Error Handling

String Methods

```
nama = "  Ahmad Budiman  "

nama.strip()      # "Ahmad Budiman"
nama.upper()      # "AHMAD BUDIMAN"
nama.lower()      # "ahmad budiman"
nama.title()      # "Ahmad Budiman"

email = "ahmad@bank.co.id"
email.startswith("ahmad")  # True
email.endswith(".co.id")   # True
"@" in email               # True
```

F-strings

```
nama = "Siti Rahayu"
saldo = 15750000

print(f"Nama: {nama}")
print(f"Saldo: Rp {saldo:,}")
print(f"Persentase: {0.0575:.2%}")
```

Error Handling

```
def tarik_tunai(rekening, nominal):
    try:
        saldo = get_saldo(rekening)
        if nominal > saldo:
            raise ValueError("Saldo tidak cukup")

        update_saldo(rekening, saldo - nominal)
        return {"status": "sukses"}

    except ValueError as e:
        return {"status": "gagal", "pesan": str(e)}
    except ConnectionError:
        return {"status": "gagal", "pesan": "DB error"}
    finally:
        log_transaksi(rekening, nominal)
```