# 9. Perangkat Lunak Jaringan Kendali HVAC

Perangkat lunak jaringan kendali HVAC dikembangkan menggunakan bahasa Python, dengan arsitektur modular yang terdiri dari beberapa komponen utama:

* **Program Kontrol (Python)**: Logika kontrol utama diimplementasikan dalam Python, menggantikan ladder/structured text PLC, untuk mengatur urutan operasi, safety, dan fault handling.
* **HMI/SCADA**: Antarmuka operator berbasis Python Tkinter, digunakan untuk monitoring status sistem, alarm, dan kontrol manual perangkat HVAC.
* **Komunikasi**: Protokol komunikasi Modbus TCP/IP digunakan untuk pertukaran data antara modul kontrol Python, HMI, dan perangkat eksternal (jika ada).
* **Simulasi (Python)**:  
  Salah satu keunggulan utama pengembangan perangkat lunak HVAC ini adalah adanya modul simulasi berbasis Python. Modul simulasi ini memungkinkan seluruh logika kontrol, interaksi sensor, aktuator, serta respon sistem dapat diuji secara virtual sebelum implementasi fisik.  
  Simulasi dilakukan dengan membuat model digital dari seluruh perangkat (sensor, pompa, valve, alarm) dan lingkungan proses (misal: perubahan level, tekanan, dan kualitas air). Operator dapat mengatur nilai input sensor secara manual atau otomatis, lalu mengamati bagaimana sistem kontrol merespons kondisi tersebut.  
  Dengan simulasi Python, tim pengembang dapat:
  + Menguji dan memverifikasi algoritma kontrol tanpa risiko terhadap perangkat nyata.
  + Melatih operator menggunakan HMI virtual.
  + Melakukan troubleshooting dan analisis skenario kegagalan (fault simulation).
  + Mempercepat iterasi pengembangan, karena perubahan kode dapat langsung diuji pada lingkungan simulasi.
  + Menyediakan visualisasi proses secara real-time, sehingga memudahkan pemahaman alur kerja sistem HVAC. Simulasi ini juga terintegrasi dengan fitur data logging dan alarm, sehingga seluruh kejadian selama simulasi dapat dicatat dan dianalisis.
* **Data Logging**: Fitur pencatatan data proses (level, tekanan, alarm, status aktuator) untuk analisis performa dan troubleshooting.

## 9.1 Lampiran: Control System Plan and I/O Table

### 9.1.1 Control Philosophy

* Sistem HVAC Python berjalan otomatis penuh, dengan opsi override manual untuk semua pompa dan valve melalui HMI.
* Logika kontrol utama berbasis pembacaan sensor (level, flow, pressure, turbidity) yang diolah secara real-time.
* Alarm dihasilkan secara otomatis jika terjadi kondisi abnormal (misal: level rendah/tinggi, tekanan rendah, turbidity tinggi).
* Semua parameter kritis dimonitor dan dicatat ke file log.
* HMI lokal berbasis Tkinter untuk operator; pemantauan jarak jauh dapat diaktifkan jika diperlukan.
* **Pembaruan Arsitektur:** Pemisahan tegas antara sensor, logika, dan aktuator diimplementasikan dalam struktur kode Python, sesuai flowchart terbaru.

### 9.1.2 Main Control Logic

* **Intake Fan**: Aktif jika pre-treatment dan sistem HVAC siap, nonaktif jika level ground tank rendah atau alarm aktif.
* **Pre-treatment**: Siklus backwash otomatis berdasarkan tekanan diferensial atau timer.
* **HVAC High-Pressure Pump**: Aktif jika pre-treatment OK dan level ground tank cukup; nonaktif jika tekanan rendah/tinggi atau level ground tank rendah.
* **Post-treatment**: Disinfeksi berjalan paralel dengan output HVAC.
* **Transfer Pump to Roof Tank**: Aktif jika roof tank belum penuh dan ground tank masih ada air; nonaktif jika roof tank penuh atau ground tank rendah.
* **Alarm**: Setiap pembacaan sensor abnormal akan memicu alarm dan dapat menghentikan perangkat terkait.
* **Pemetaan Logika:** Semua nilai sensor dikirim ke modul logika Python, yang kemudian mengontrol aktuator sesuai algoritma.

### 9.1.3 I/O Table

Tabel I/O diimplementasikan dalam Python sebagai dictionary atau class, dengan pemetaan variabel berikut:

* **AI (Analog Input):** Pembacaan sensor level, tekanan, flow, turbidity (misal: LT-101, PT-101, TU-101).
* **DO (Digital Output):** Kontrol aktuator (pompa, valve, UV, alarm).
* **DI/DO:** Status sistem (System\_Running, Emergency\_Stop).
* **Internal Logic Variables:** Variabel boolean untuk status logika (PreTreatment\_OK, RO\_OK, Alarm, dst).

### 9.1.4 Control Logic Summary

#### System States:

1. **Emergency Stop**: Semua aktuator OFF, alarm ON, System\_Running = FALSE.
2. **System Stopped**: Semua aktuator OFF, System\_Running = FALSE.
3. **System Running**: Logika operasi normal aktif, System\_Running = TRUE.

#### Key Logic Conditions (Python Implementation):

* PreTreatment\_OK: TU\_101 < 5.0 NTU
* RO\_OK: PreTreatment\_OK and LT\_101 > 20% and 50 < PT\_101 < 70
* P\_101 (Intake): RO\_OK and LT\_101 < 95%
* P\_102 (RO Pump): RO\_OK
* P\_103 (Post-treatment): LT\_101 > 30%
* P\_104 (Transfer to Ground): RO\_OK and LT\_101 < 95%
* P\_105 (Pump to Rooftop): LT\_101 > 40% and LT\_102 < 95%
* P\_106 (Transfer to Roof): LT\_102 < 98%
* UV\_101: RO\_OK
* V\_101: P\_101 or P\_102
* PRV\_101: PT\_101 > 70
* Alarm: TU\_101 > 10 or PT\_101 < 45 or PT\_101 > 75 or LT\_101 < 10 or LT\_102 > 98

### 9.2 Notes

* Semua sinyal analog menggunakan range 4–20 mA (disimulasikan dalam Python).
* Semua pompa dan valve dapat dikontrol secara local/remote dan manual/auto via HMI.
* Sistem dapat dikembangkan untuk pemantauan jarak jauh, data logging lanjutan, dan diagnostik.
* **Pembaruan Arsitektur:** Tabel I/O dan pemetaan variabel Python telah disesuaikan dengan kebutuhan simulasi dan HMI.
* **Sinkronisasi:** Struktur variabel Python sepenuhnya sesuai dengan flowchart dan kebutuhan operasional HVAC.