**Flowchart Logika Penalaran Regulator Tekanan Kualitatif**

graph TD

A[Mulai] --> B{Terima Input Kualitatif};

B --> C[Inisialisasi Status Variabel<br>(Pin, Pout, Q, dll.)<br>Input Ditetapkan, Sisanya 'Tidak Diketahui'];

C --> D{Perulangan Propagasi<br>Ada Variabel Baru yang<br>Dapat Ditentukan?};

D -- Ya --> E{Periksa Setiap Aturan Logika<br>(Semua Grup Pi)};

E --> F{Jika semua variabel<br>dalam sebuah aturan<br>sudah diketahui<br>kecuali satu?};

F -- Ya --> G[Tentukan Status Baru<br>untuk Variabel yang Tidak Diketahui];

G --> H{Cek Kontradiksi:<br>Apakah Status Baru Berbeda<br>dengan Status yang Sudah Ada?};

H -- Ya --> I[Catat Kontradiksi];

H -- Tidak --> J[Update Status Variabel];

J --> E;

I --> J;

D -- Tidak --> K{Apakah Ada Kontradiksi yang Dicatat?};

K -- Ya --> L[Laporkan Variabel yang Kontradiksi<br>dan Jelaskan Alasannya];

K -- Tidak --> M[Laporkan Semua Status Akhir<br>Status Logis dan Konsisten];

L --> N[Selesai];

M --> N;

**Penjelasan Alur Logika:**

1. **Mulai:** Proses dimulai ketika Anda menjalankan program.
2. **Terima Input Kualitatif:** Algoritma menerima input kualitatif dari pengguna, yaitu status **Naik (I)**, **Turun (D)**, atau **Konstan (C)** untuk variabel-variabel seperti Pin dan Pout.
3. **Inisialisasi Status Variabel:** Semua variabel dalam sistem (seperti Q, A\_open, x, dll.) diatur ke status "Tidak Diketahui". Hanya variabel yang diinput oleh pengguna (Pin, Pout) yang memiliki status awal.
4. **Perulangan Propagasi:** Ini adalah inti dari algoritma. Proses akan berulang terus menerus selama masih ada variabel baru yang statusnya dapat ditentukan.
5. **Periksa Setiap Aturan Logika:** Di setiap perulangan, algoritma memeriksa setiap aturan Pi (seperti pi\_A1, pi\_A2, dll.).
6. **Cek Kondisi Penentuan:** Untuk setiap aturan, algoritma mencari satu-satunya variabel yang belum diketahui statusnya.
7. **Tentukan Status Baru:** Jika kondisi di atas terpenuhi, algoritma menggunakan aturan berbanding lurus dan terbalik untuk menentukan status baru dari variabel yang tidak diketahui tersebut.
8. **Cek Kontradiksi:** Sebelum menetapkan status baru, algoritma membandingkannya dengan status yang sudah ada (jika ada).
   * Jika status baru **bertentangan** (misalnya, Naik vs. Turun), algoritma mencatatnya sebagai kontradiksi.
   * Jika status baru **tidak bertentangan**, status variabel diperbarui.
9. **Laporkan Hasil:** Setelah perulangan selesai (tidak ada lagi variabel yang dapat ditentukan), algoritma memeriksa apakah ada kontradiksi yang dicatat.
   * **Tidak ada kontradiksi:** Logika konsisten. Algoritma melaporkan status akhir dari semua variabel.
   * **Ada kontradiksi:** Logika tidak konsisten. Algoritma melaporkan variabel yang mengalami kontradiksi dan berhenti.

**Flowchart Logika Penalaran Regulator Tekanan Kualitatif**

graph TD

A[Mulai] --> B{Terima Input Kualitatif};

B --> C[Inisialisasi Status Variabel<br>(Pin, Pout, Q, dll.)<br>Input Ditetapkan, Sisanya 'Tidak Diketahui'];

C --> D{Perulangan Propagasi<br>Ada Variabel Baru yang<br>Dapat Ditentukan?};

D -- Ya --> E{Periksa Setiap Aturan Logika<br>(Semua Grup Pi)};

E --> F{Jika semua variabel<br>dalam sebuah aturan<br>sudah diketahui<br>kecuali satu?};

F -- Ya --> G[Tentukan Status Baru<br>untuk Variabel yang Tidak Diketahui];

G --> H{Cek Kontradiksi:<br>Apakah Status Baru Berbeda<br>dengan Status yang Sudah Ada?};

H -- Ya --> I[Catat Kontradiksi];

H -- Tidak --> J[Update Status Variabel];

J --> E;

I --> J;

D -- Tidak --> K{Apakah Ada Kontradiksi yang Dicatat?};

K -- Ya --> L[Laporkan Variabel yang Kontradiksi<br>dan Jelaskan Alasannya];

K -- Tidak --> M[Laporkan Semua Status Akhir<br>Status Logis dan Konsisten];

L --> N[Selesai];

M --> N;

**Penjelasan Alur Logika:**

1. **Mulai:** Proses dimulai ketika Anda menjalankan program.
2. **Terima Input Kualitatif:** Algoritma menerima input kualitatif dari pengguna, yaitu status **Naik (I)**, **Turun (D)**, atau **Konstan (C)** untuk variabel-variabel seperti Pin dan Pout.
3. **Inisialisasi Status Variabel:** Semua variabel dalam sistem (seperti Q, A\_open, x, dll.) diatur ke status "Tidak Diketahui". Hanya variabel yang diinput oleh pengguna (Pin, Pout) yang memiliki status awal.
4. **Perulangan Propagasi:** Ini adalah inti dari algoritma. Proses akan berulang terus menerus selama masih ada variabel baru yang statusnya dapat ditentukan.
5. **Periksa Setiap Aturan Logika:** Di setiap perulangan, algoritma memeriksa setiap aturan Pi (seperti pi\_A1, pi\_A2, dll.).
6. **Cek Kondisi Penentuan:** Untuk setiap aturan, algoritma mencari satu-satunya variabel yang belum diketahui statusnya.
7. **Tentukan Status Baru:** Jika kondisi di atas terpenuhi, algoritma menggunakan aturan berbanding lurus dan terbalik untuk menentukan status baru dari variabel yang tidak diketahui tersebut.
8. **Cek Kontradiksi:** Sebelum menetapkan status baru, algoritma membandingkannya dengan status yang sudah ada (jika ada).
   * Jika status baru **bertentangan** (misalnya, Naik vs. Turun), algoritma mencatatnya sebagai kontradiksi.
   * Jika status baru **tidak bertentangan**, status variabel diperbarui.
9. **Laporkan Hasil:** Setelah perulangan selesai (tidak ada lagi variabel yang dapat ditentukan), algoritma memeriksa apakah ada kontradiksi yang dicatat.
   * **Tidak ada kontradiksi:** Logika konsisten. Algoritma melaporkan status akhir dari semua variabel.
   * **Ada kontradiksi:** Logika tidak konsisten. Algoritma melaporkan variabel yang mengalami kontradiksi dan berhenti.