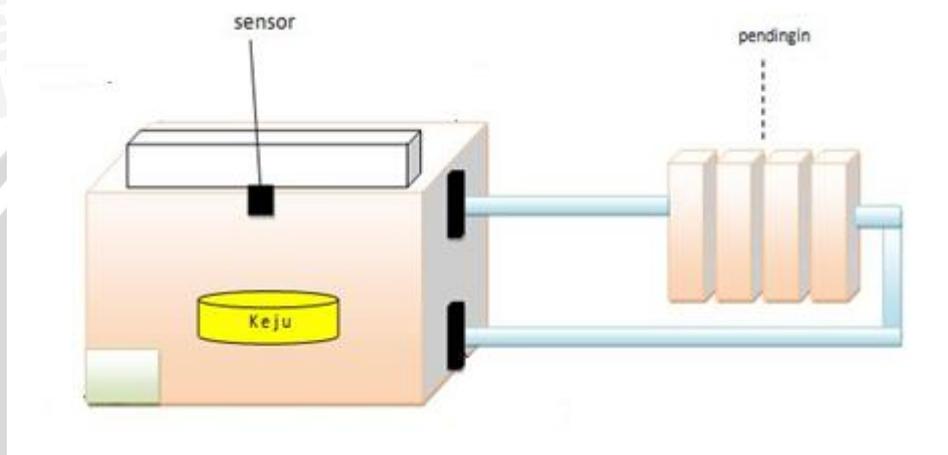


## BAB III

### PERANCANGAN SIMULASI SISTEM

#### 3.1. Perancangan Simulasi Sistem

Perancangan plant sistem dilakukan sebagai langkah awal sebelum terbentuknya suatu sistem, hal ini dimaksudkan agar sistem pengendalian suhu dapat berjalan sesuai deskripsi awal yang telah direncanakan. Plant sistem berupa pematang keju dan komponen-komponen pendukung lainnya yang digambarkan sebagai berikut:



**Gambar 3.1.** Sistem pematang keju

Sistem ini terdiri dari sensor yang membaca suhu di dalam box pematang keju. Sinyal dari sensor diteruskan ke mikrokontroller untuk diproses. Setelah memproses sinyal dari sensor, mikrokontroller mengeluarkan sinyal keluaran yang mengatur evaporator untuk mendinginkan udara dalam pematang keju.

#### 3.2. Pemodelan Sistem

Suhu luar yang masuk akan berfluktuasi dan dapat bekerja sebagai gangguan. Suhu yang masuk diubah oleh pendingin secara tiba-tiba dari suhu awal ( $\theta_i$ ) sampai dengan suhu akhir ( $\theta_0$ ) yang diinginkan. Sistem dari plan ini memiliki kapasitas termal ruangan ( $C_R$ ) dengan perubahan suhu yang dipengaruhi oleh sistem pendingin. Faktor yang lainnya adalah kapasitas termal keju ( $C_K$ ) dengan perubahan suhu yang dipengaruhi oleh sistem pendingin. maka persamaan diferensial untuk kasus ini adalah:

$$C_K \frac{d^2\theta}{dt} + C_R \frac{d\theta}{dt} = -h_i - h_o \quad (3.1)$$

Dimana laju kalor pendingin masuk  $h_i$  sebanding dengan suhu ( $\theta$ ) yang dibagi dengan tahanan termal ( $R$ ), sehingga persamaan ini dapat disubstitusikan ke dalam persamaan (3.1). hasil dari substitusi ini dapat dilihat dipersamaan (3.2).

$$C_k \frac{d^2\theta}{dt} + C_R \frac{d\theta}{dt} = -h_i - \frac{\theta}{R} \quad (3.2)$$

Persamaan (3.2) dikalikan dua ruas dengan variabel  $R$  sehingga hasilnya dapat dilihat di persamaan (3.3).

$$C_K R \frac{d^2\theta}{dt} + C_R R \frac{d\theta}{dt} + \theta = -h_i R \quad (3.3)$$

Dengan mentransformasikan laplace pada persamaan (3.3) akan didapat hasilnya dipersamaan (3.4).

$$\mathcal{L} \left[ -C_K R \frac{d^2\theta}{dt} - C_R R \frac{d\theta}{dt} - \theta \right] = \mathcal{L}[h_i R]$$

$$-C_K R s^2 \theta(s) - C_R R s \theta(s) - 1 \theta(s) = R H_i(s) \quad (3.4)$$

Hasil dari transformasi laplace persamaan (3.4) dibuat fungsi alihnya dengan perbandingan nilai output sistem dan input sistem. fungsi alih ini dapat dilihat pada persamaan (3.5)

$$\frac{H_i(s)}{\theta(s)} = \frac{R}{-C_K R s^2 - C_R R s - 1} \quad (3.5)$$

$$\frac{\left(\frac{1}{Gc}\right)}{\left((MC_K)\left(\frac{1}{Gc}\right)\right)s^2 - \left((MC_R)\left(\frac{1}{Gc}\right)\right)s - 1} \quad (3.6)$$

Box pada simulasi ini berukuran panjang 10 cm, lebar 10 cm , tinggi 10 cm. Massa keju berukuran 1 kg, volume ruang berukuran 10 liter, massa jenis udara bernilai  $1.293 \text{ kg/m}^3$ , massa jenis keju adalah  $1.29 \text{ kg/m}^3$ , kalor jenis keju adalah  $0.595 \text{ J/kg}^\circ\text{C}$  , kecepatan angin adalah  $4 \text{ m/s}$  , kalor jenis angina bernilai  $1 \text{ J/kg}^\circ\text{C}$  Untuk persamaan

diatas dimasukan nilai-nilai dari parameter yang sudah ditentukan. Sehingga model dari matematika sistem pematang keju adalah

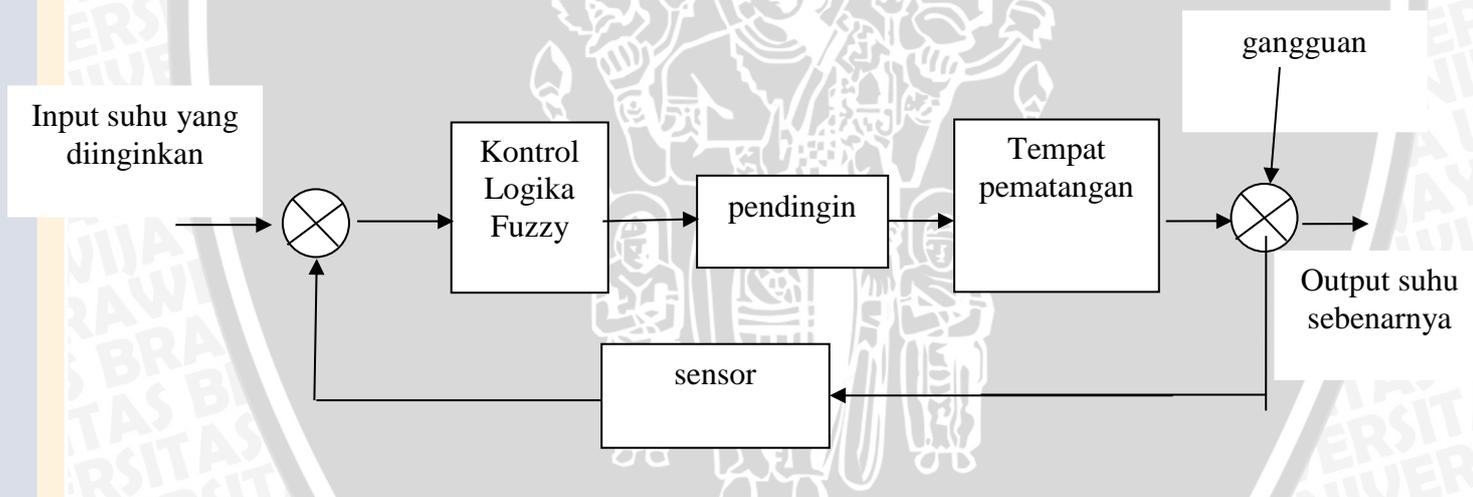
$$\frac{H_i(s)}{\theta(s)} = \frac{-0.25}{-2s^2 - 0.26s - 1}$$

Nilai kapasitas termal (C) dapat dilihat dari persamaan 2.3 Dalam kasus ini R sama dengan satu dibagi dengan laju udara dan kalor jenis (c) sehingga penyederhanaan persamaan 3.5 dengan langkah sebagai berikut

$$\frac{H_i(s)}{\theta(s)} = \frac{R}{(M \cdot c)Rs^2 - C_R Rs - 1} \quad (3.7)$$

### 3.3. Blok Diagram

Pemodelan untuk simulasi pematang keju secara keseluruhan dapat digambarkan melalui fungsi blok yang ada sehingga dapat dilihat pada gambar berikut.

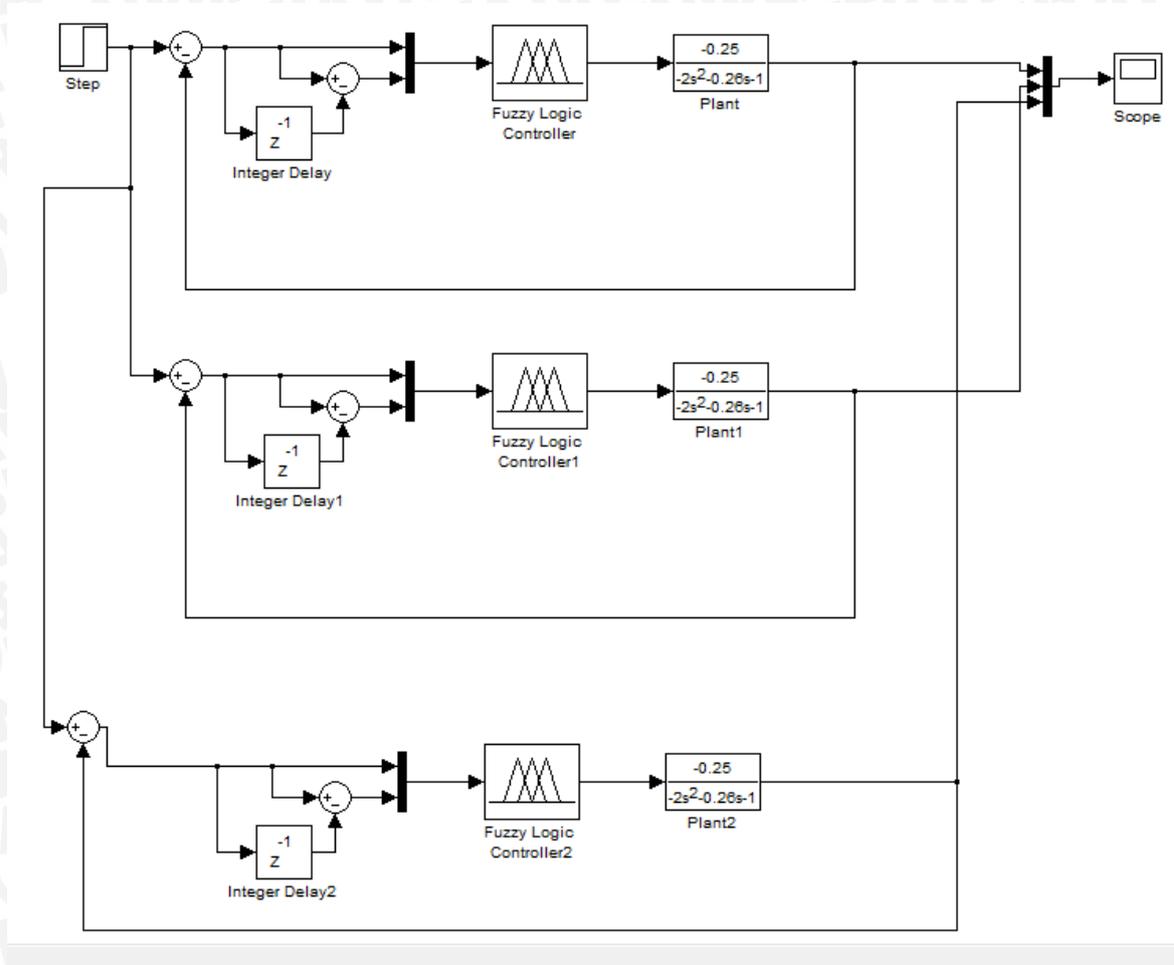


**Gambar 3.2.** Blok Diagram Sistem (Perancangan)

Aktuator akan aktif saat tombol ON yang berfungsi sebagai pendingin ruangan didalam box. Nilai dari perubahan suhu akan di *feedback* kembali untuk disamakan dengan nilai *set point* yang telah ditetapkan. Setelah suhu didapatkan, maka kontroler akan memberikan arus yang sesuai untuk mendinginkan suhu ruangan pada box. Dengan demikian suhu akan stabil dengan dengan masukan yang disetel diawal.

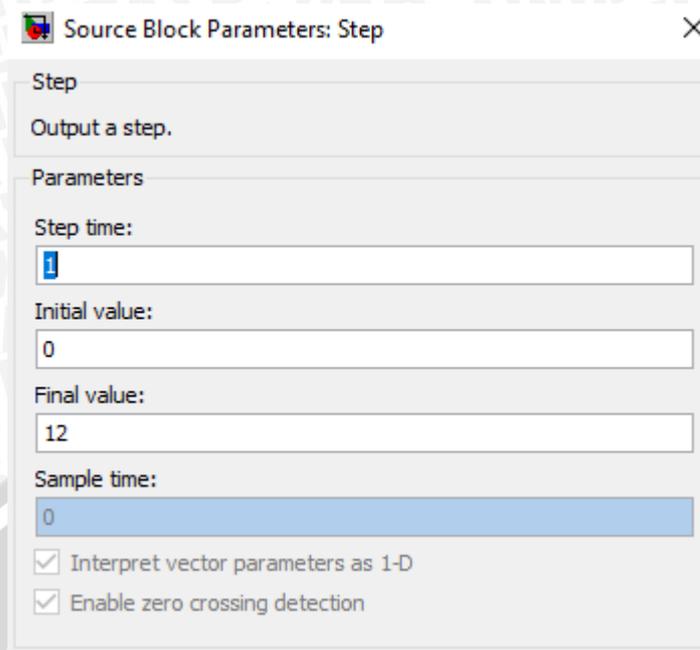
### 3.4. Simulasi dengan Simulink

Pemodelan untuk simulasi pematang keju secara keseluruhan dapat digambarkan melalui fungsi blok yang ada di Simulink sehingga dapat dilihat pada gambar berikut:



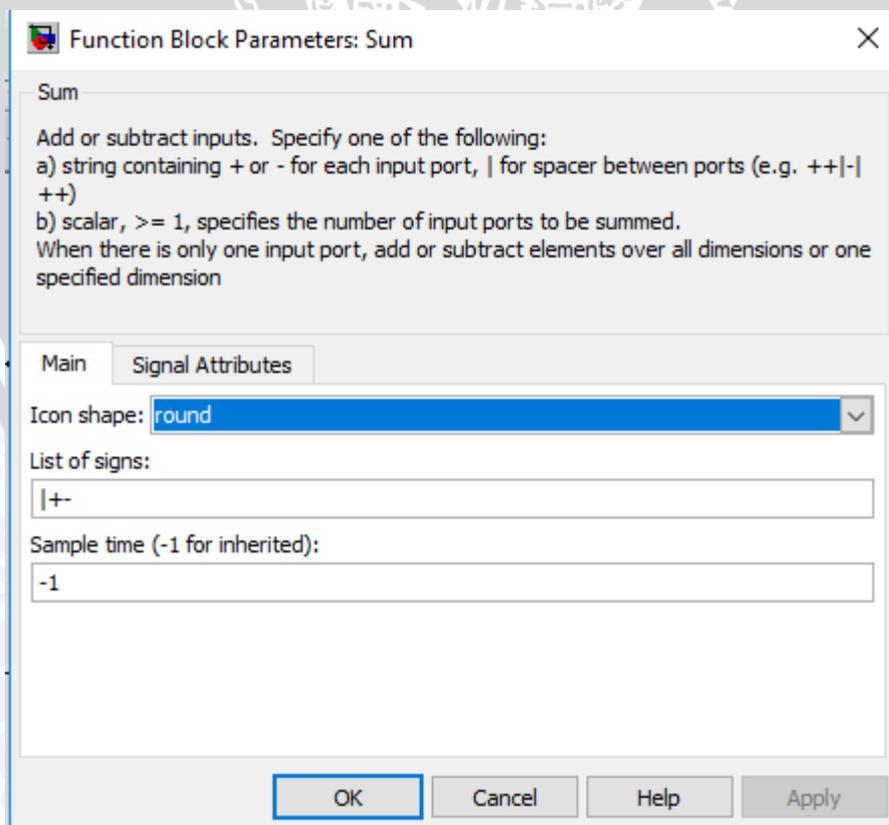
Gambar 3.3. Blok Simulasi Sistem

Dari gambar 3.3. Fungsi blok *Step* sebagai pembangkit sinyal dengan *set point* 12° C dan parameter yang lain dapat dilihat di gambar 3.4.



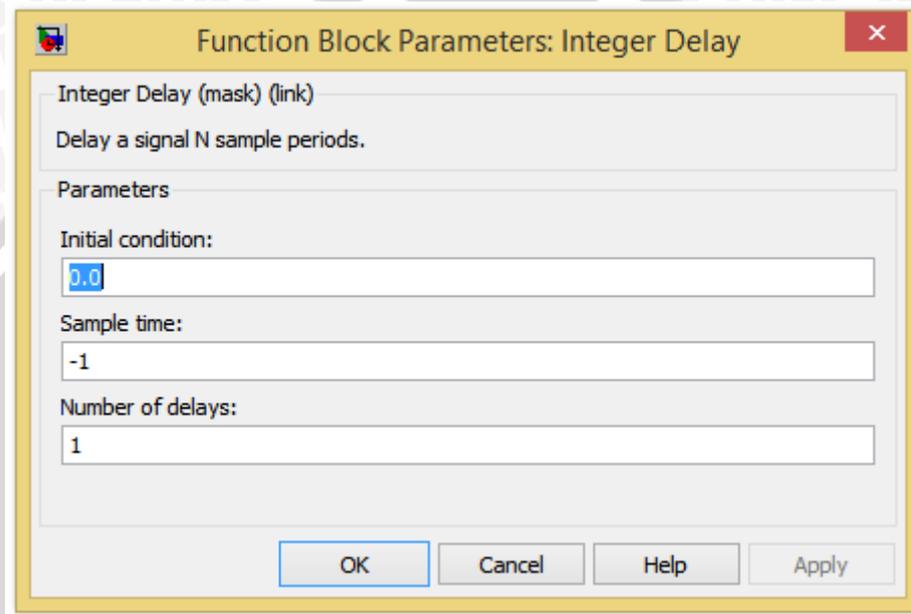
**Gambar 3.4** Parameter *block step*

Sinyal dari *block step* dijumlahkan dari hasil dari feedback dengan menggunakan *function block parameter: sum*. Dan nilai parameter nya dapat dilihat pada gambar 3.5.



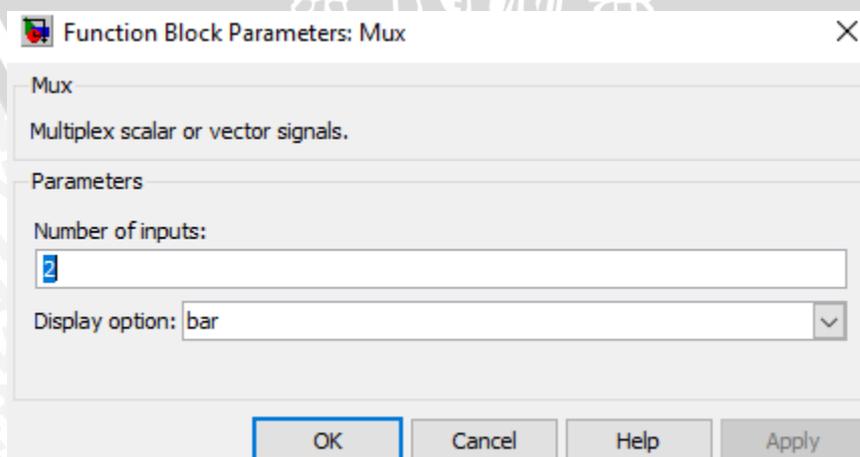
**Gambar 3.5.** Parameter *block sum*

Sinyal dari *block* penjumlahan akan dibagi menjadi dua bagian. Satu sinyal sebagai nilai *error* dan satu lagi sebagai nilai  $d(\text{error})$ . Nilai  $d(\text{error})$  adalah perubahan nilai *error* yang sebelumnya. Nilai *error* yang sebelumnya disimpan di *block memory* kemudian dijumlahkan dengan keadaan *error*. Hasil dari kedua sinyal ini masuk kedalam *function block parameter: Integer Delay* kemudian dijumlahkan dengan keadaan *error* dapat dilihat pada Gambar 3.6.



**Gambar 3.6.** *Function Block Parameters: Integer Delay*

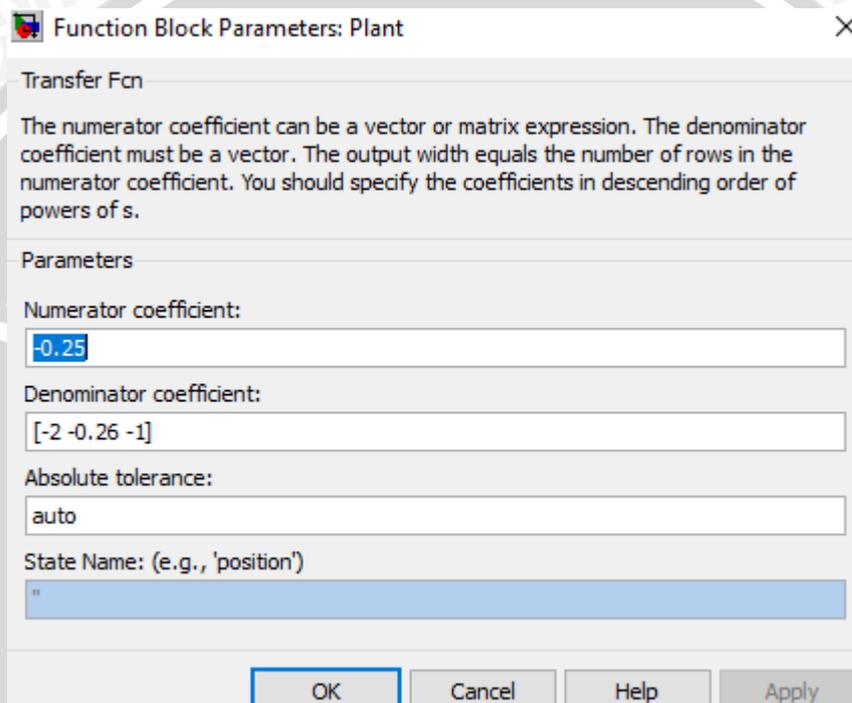
Dari dua sinyal masuk dijadikan menjadi satu keluaran sinyal digunakan *function block parameter: mux*. Parameter dari nilai dari *Function block parameter: mux* dapat dilihat pada Gambar 3.7.



**Gambar 3.7.** Parameter *block mux*

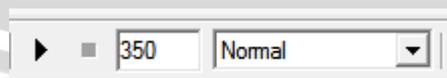
Sinyal dari *function block parameter: mux* digunakan sebagai input ke dalam *function block parameter: fuzzy logic controller* yang nantinya akan dikendalikan sehingga mendapatkan respon sistem yang diinginkan.

Sinyal hasil dari *function block parameter: fuzzy logic controller* akan diberikan untuk mengendalikan *function block parameter: plant* dengan nilai yang telah dimodelkan sebelumnya, sehingga didapat seperti gambar 3.8.



**Gambar 3.8.** *function block parameter: plant*

Sinyal akan dibalik ke *function block parameter: sum* untuk dijumlahkan dengan nilai *set point*. Sistem ini akan terus berjalan sampai nilai *simulation stop time* yang telah diset diawal. Nilai *simulation stop time* pada sistem dapat dilihat pada gambar 3.9.



**Gambar 3.9.** *simulation stop time*