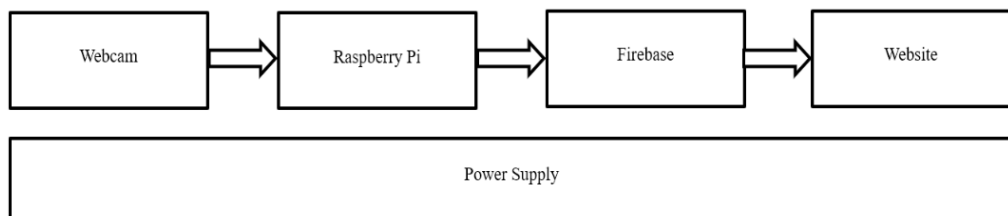


BAB III

METODE PENELITIAN

3.1. Blok Diagram

Dalam perancangan tugas akhir dengan judul “**Sistem Monitoring Tanaman Selada Berbasis pengolahan Citra Dengan *Internet of Things***” diawali dengan perancangan blok diagram dari sistem yang dirancang. Tujuan dibuatnya blok diagram sistem adalah untuk memudahkan pemahaman terkait input atau output dari sistem serta susunan komponen elektronika pada sistem yang dirancang. Berikut gambar mengenai blok diagram sistem yang dirancang:



Gambar 3. 1 Blok Diagram Sistem

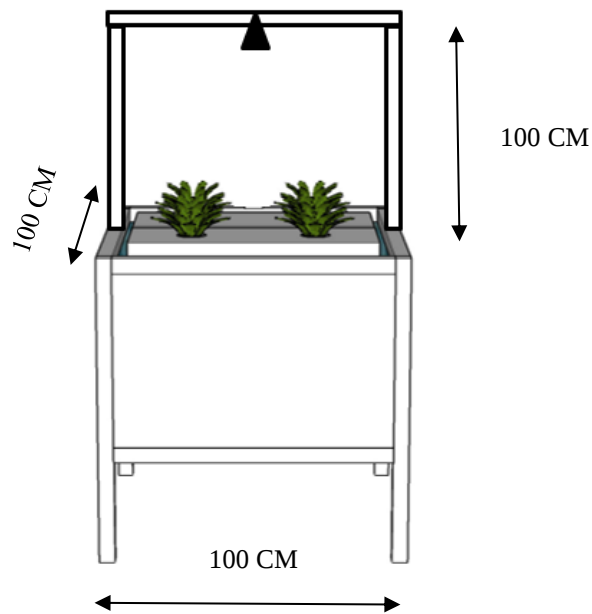
Pada blok diagram sistem gambar 3.1. diatas dibagi kedalam blok – blok diagram yang meliputi perangkat input, proses dan output. Adapun untuk penjelasannya sebagai berikut:

1. Webcam merupakan perangkat input yang berfungsi mengambil gambar dari tanaman yang sedang tumbuh untuk nantinya hasil gambar dikirim ke Raspberry Pi untuk diolah oleh program pengolahan citra.
2. Raspberry Pi sebagai pengendali utama alat tempat input gambar yang ditangkap oleh Webcam dan diolah oleh program pengolahan citra yang nantinya akan diukur luas dari hasil gambar tersebut. Selain itu juga akan menerima data serial berupa tingkat kelembapan oleh Arduino. Raspberry Pi akan mengirim data melalui wifi ke Firebase sebagai database data.
3. Firebase merupakan database data tempat data – data hasil pengolahan citra tadi dikirim. Data yang dikirim di Firebase nantinya akan dapat dilihat pada aplikasi Website.

Sistem diatas menggunakan Raspberry Pi sebagai pengendali utamanya dan webcam yang akan mengambil gambar yang nantinya akan diproses oleh Raspberry Pi. Hasil gambar yang telah diolah oleh Raspberry Pi dan diketahui tingkat kematangannya melalui luas tanaman akan dikirim melalui sinyal WiFi ke Firebase. Dari data yang ada pada Firebase tersebut akan ditampilkan melalui website.

3.2. Desain Alat

Gambar dibawah ini merupakan desain alat pada bagian tempat pertumbuhan tanaman yang dilengkapi dengan Webcam yang berfungsi sebagai sensor yang mendeteksi luas tanaman untuk mengetahui apakah tanaman sudah matang atau belum. Posisi Webcam diletakkan diatas alat dengan tujuan agar dapat melihat tanaman dari atas sehingga dapat didapat gambar tanaman yang bisa diukur luasnya.



Gambar 3. 2 Desain Alat

3.3. Spesifikasi Alat

- Spesifikasi:

Spesifikasi dari alat Sistem Monitoring Tanaman Selada Berbasis pengolahan Citra Dengan Internet of Things adalah sebagai berikut:

1. Dimensi alat 100 cm x 100 cm x 100cm.
2. Menggunakan daya 220 VAC

3. Menggunakan catu daya 5VDC
4. Menggunakan Raspberry Pi 3B+
5. Menggunakan Sensor Kelembaban
6. Menggunakan Webcam
7. Batas tray yang dapat ditampung kolam sebanyak 3 tray dengan dimensi 14cm x 85cm.
8. 3 Tanaman disetiap tray.

- **Fitur:**

Fitur dari alat Sistem Monitoring Tanaman Selada Berbasis pengolahan Citra Dengan Internet of Things adalah sebagai berikut:

1. Mengetahui tingkat kematangan dari tanaman selada dari perhitungan luas oleh penangkapan citra.
2. Mengetahui data pertumbuhan tanaman selada dari website secara online.

3.4. Teknik Fabrikasi

Pada pembuatan alat Sistem Monitoring Tanaman Selada Berbasis pengolahan Citra Dengan Internet of Things dibagi menjadi beberapa tahapan dalam pembuatannya yang dijelaskan sebagai berikut:

3.4.1. Perancangan Perangkat Keras

a) Kolam Penanaman

Pada kolam penanaman dibuat menggunakan besi siku berlubang yang telah dipotong menggunakan gerinda. Yang selanjutnya dilakukan pengeboran untuk membuat lubang yang digunakan untuk memasang baut. Lalu potongan besi siku berlubang dirangkai menjadi kolam seperti pada gambar 3.5 kemudian diberikan terpal agar air tidak keluar dari kolam.



Gambar 3. 3 Pemotongan besi siku dengan gerinda



Gambar 3. 4 Melubangi besi siku dan triplek dengan bor

b) Tempat mounting webcam

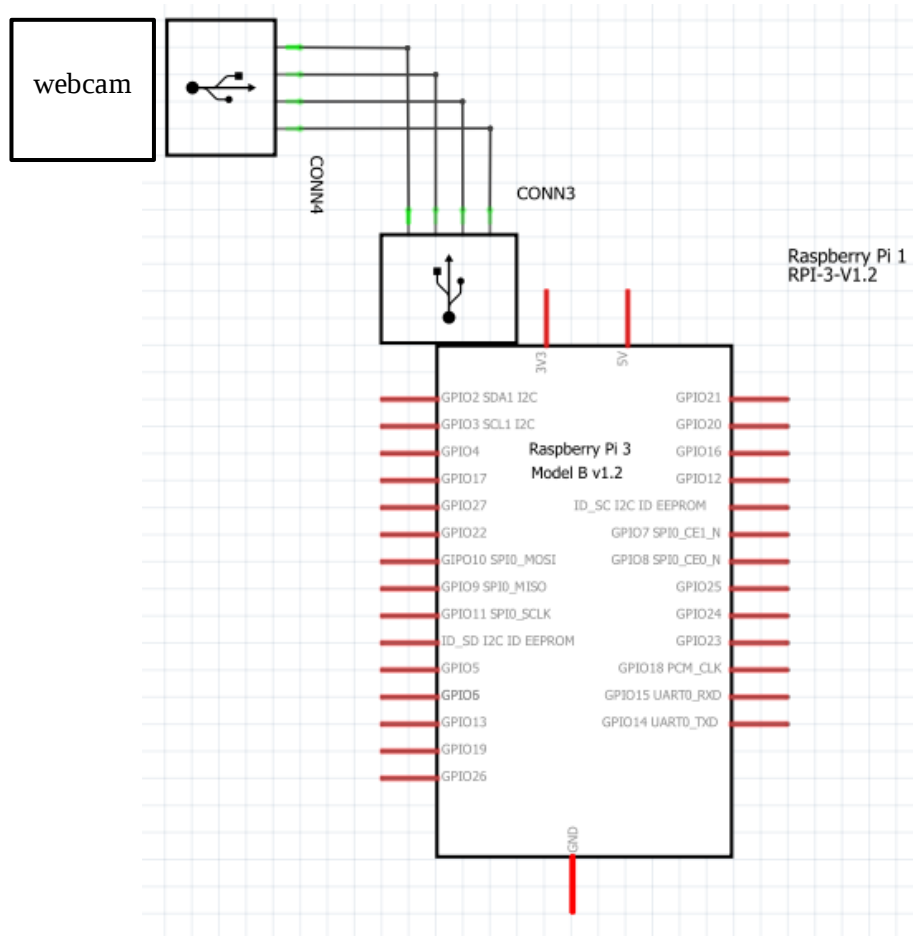
Untuk tempat mounting webcam dibuat juga menggunakan besi siku berlubang yang dipotong menggunakan gerinda dan dibentuk sesuai dengan gambar 3.5 yang kemudian dipasang pada kolam penanaman.



Gambar 3. 5 Meemasang webcam pada tempat mounting webcam

3.4.2. Perancangan Elektrikal

a. Diagram Skematik Sistem



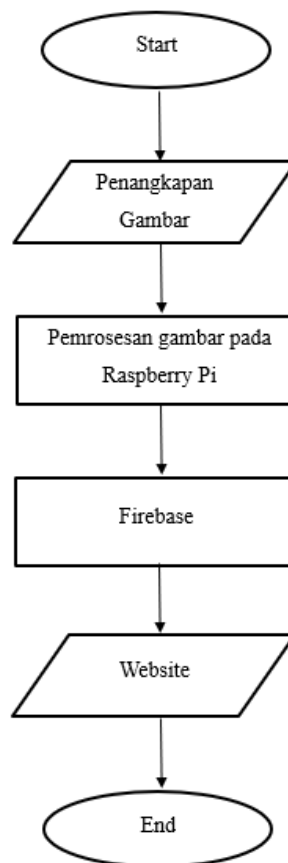
Gambar 3. 6 Diagram Skematik Sistem

Perancangan diagram skematik bertujuan memudahkan pembacaan rangkaian elektronik yang digunakan dalam sistem kendali penanaman hidroponik. Diagram skematik ini mencakup semua komponen elektronik yang terdapat pada sistem yang kemudian dirangkai dalam sebuah box kendali. Raspberry Pi akan menerima data dari pembacaan webcam lewat USB dan data tersebut akan diolah oleh Raspberry Pi dan nantinya akan diunggah ke Firebase.

3.4.3. Perancangan Perangkat Lunak

Pada perancangan perangkat lunak akan dijelaskan tentang flowchart dari sistem yang dibuat dan coding yang digunakan.

- a. Flowchart program tingkat kematangan tumbuhan



Gambar 3. 7 Flowchart program tingkat kematangan tumbuhan

Pada flowchart diatas merupakan urutan program kerja alat yang dimulai dengan pembacaan gambar oleh webcam yang dilakukan dengan coding dibawah.

Tabel 3. 1. Tabel penjelasan coding pengambilan gambar

<pre>from cv2 import * import cv2 import time</pre>	Pada coding ini berfungsi untuk memanggil library yang akan digunakan pada coding.
<pre>cam = cv2.VideoCapture(0) result, image = cam.read() if result: cv2.imwrite("/home /pi/Selected/1.png ", image)</pre>	Pada coding ini berfungsi untuk memanggil fitur pengambilan gambar dari OpenCV dan menyimpan di folder yang telah ditentukan.
<pre>else: print("no image") time.sleep(2) cam.release() cv2.destroyAllWindows()</pre>	Pada coding ini menjelaskan jika terjadi error atau gambar tidak terambil dan program akan tertutup.

Dari hasil gambar yang ditangkap oleh webcam dari coding diatas kemudian akan dilakukan pendeteksian sekaligus pemotongan bagian gambar yang berwarna hijau untuk menentukan bagian dari tanaman selada menggunakan model deteksi Canny dengan coding seperti dibawah.

Tabel 3. 2. Tabel pemotongan gambar

<code>import cv2 import numpy as np</code>	Pada coding ini berfungsi untuk memanggil library.
<code>img=cv2.imread('/home/pi/Selected/1.png')</code>	Untuk mengambil gambar dari folder.
<code>crop_img = img[5:430, 95:585]</code>	Melakukan pemotongan gambar.
<code>rgb=cv2.cvtColor(crop_img, cv2.COLOR_BGR2RGB)</code>	Merubah gambar dari BGR ke RGB.
<code>hsv=cv2.cvtColor(rgb, cv2.COLOR_RGB2HSV)</code>	Merubah gambar dari RGB ke HSV.
<code>low_green=np.array([30, 60, 50], dtype=np.uint8) high_green=np.array([70, 255, 255], dtype=np.uint8)</code>	Batas atas dan batas bawah warna hijau yang akan dideteksi.
<code>mask=cv2.inRange(hsv, low_green, high_green) result=cv2.bitwise_and(crop_img, crop_img, mask=mask)</code>	Melakukan masking untuk menyisakan hasil gambar yang sesuai perintah sebelumnya.
<code>noiseless=cv2.fastNlMeansDenoisingColo</code>	Melakukan pembersihan gambar dari noise.

<pre>red(result, None, 20, 20, 7, 21) ret, final = cv2.threshold(mask, 0, 255, cv2.THRESH_BINARY)</pre>	
<pre>edged = cv2.Canny(final, 10, 250) contours, hierarchy = cv2.findContours(fi nal.copy(), cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NO NE) idx = 0 for c in contours: x, y, w, h = cv2.boundingRect(c) if w>30 and h>30: idx+=1 new_img=final[y:y+h, x:x+w] #cropping images</pre>	Melakukan deteksi tepi Canny sesuai dengan nilai yang dimasukkan.
<pre>cv2.imwrite("/home/ pi/Selected/cropped /"+str(idx) +</pre>	Memotong gambar dan menyimpannya pada folder yang telah ditentukan.

<pre> '.png', new_img) cv2.imshow("Original Image",image) cv2.imshow("Canny Edge",edged) cv2.waitKey(0) print('>> Objects Cropped Successfully!') print(">> Check out 'cropped' Directory") </pre>	
--	--

Dari hasil coding diatas didapat 9 gambar dari masing-masing tanaman yang kemudian hasil dari pembacaan tersebut didapat tingkat kematangannya. Data tingkat kematangan, tingkat kelembapan dan juga waktu akan diunggah ke Realtime Database Firebase dengan coding berikut.

Tabel 3. 3 Tabel coding upload data firebase

<pre> import cv2 from datetime import datetime import pytz import numpy as np import pyrebase import serial import time </pre>	Untuk import library.
<pre> config = { "apiKey": "AIzaSyCHx0Vrunry- Mdm7PFY7itdywJXPfzj </pre>	Merupakan setingan yang diperlukan untuk mengunggah data ke firebase.

<pre>XTU", "authDomain": "monitoringtanaman- aae71.firebaseio.c om", "databaseURL": "https://monitoring tanaman-aae71- default-rtdb.asia- southeast1.firebaseio database.app", "projectId": "monitoringtanaman- aae71", "storageBucket": "monitoringtanaman- aae71.appspot.com", "messagingSenderId" : "974883709571", "appId": "1:974883709571:web :067171baedfdd0861a b1b6" }</pre>	
<pre>firebase = pyrebase.initialize _app(config) Jam = pytz.timezone("Asia /Pontianak") Waktu =</pre>	Merupakan coding untuk mendapatkan data waktu.

<pre> datetime.now(Jam) hasil = Waktu.strftime("%H: %M:%S") </pre>	
<pre> img1=cv2.imread('/h ome/pi/Selected/cro pped/1.png' , cv2.IMREAD_GRAYSCAL E) pixel_1 = np.sum(img1 == 255) jumlah_1 = pixel_1/12000 luas_1 = jumlah_1*100 desimal_luas_1 = round(luas_1) print('Luas Tanaman 1:', desimal_luas_1, "%") img2=cv2.imread('/h ome/pi/Selected/cro pped/2.png' , cv2.IMREAD_GRAYSCAL E) pixel_2 = np.sum(img2 == 255) jumlah_2 = pixel_2/12000 luas_2 = jumlah_2*100 </pre>	Coding untuk membaca jumlah pixel dari hasil pemotongan gambar pada coding sebelumnya.

<pre> desimal_luas_2 = round(luas_2) print('Luas Tanaman 2:', desimal_luas_2, "%") img3=cv2.imread('/h ome/pi/Selected/cro pped/3.png' , cv2.IMREAD_GRAYSCAL E) pixel_3 = np.sum(img3 == 255) jumlah_3 = pixel_3/12000 luas_3 = jumlah_3*100 desimal_luas_3 = round(luas_3) print('Luas Tanaman 3:', desimal_luas_3, "%") img4=cv2.imread('/h ome/pi/Selected/cro pped/4.png' , cv2.IMREAD_GRAYSCAL E) pixel_4 = np.sum(img4 == 255) </pre>	
---	--

jumlah_4	=	
pixel_4/12000		
luas_4	=	
jumlah_4*100		
desimal_luas_4	=	
round(luas_4)		
print('Luas Tanaman		
4:',		
desimal_luas_4,		
"%")		
img5=cv2.imread('/h		
ome/pi/Selected/cro		
pped/5.png'	,	
cv2.IMREAD_GRAYSCAL		
E)		
pixel_5	=	
np.sum(img5 == 255)		
jumlah_5	=	
pixel_5/12000		
luas_5	=	
jumlah_5*100		
desimal_luas_5	=	
round(luas_5)		
print('Luas Tanaman		
5:',		
desimal_luas_5,		
"%")		
img6=cv2.imread('/h		
ome/pi/Selected/cro		
pped/6.png'	,	

<pre> cv2.IMREAD_GRAYSCALE) pixel_6 = np.sum(img6 == 255) jumlah_6 = pixel_6/12000 luas_6 = jumlah_6*100 desimal_luas_6 = round(luas_6) print('Luas Tanaman 6:', desimal_luas_6, "%") img7=cv2.imread('/home/pi/Selected/cropped/7.png' , cv2.IMREAD_GRAYSCALE) pixel_7 = np.sum(img7 == 255) jumlah_7 = pixel_7/12000 luas_7 = jumlah_7*100 desimal_luas_7 = round(luas_7) print('Luas Tanaman 7:', desimal_luas_7, "%") </pre>	
---	--

<pre> img8=cv2.imread('/home/pi/Selected/cropped/8.png' , cv2.IMREAD_GRAYSCALE) pixel_8 = np.sum(img8 == 255) jumlah_8 = pixel_8/12000 luas_8 = jumlah_8*100 desimal_luas_8 = round(luas_8) print('Luas Tanaman 8:', desimal_luas_8, "%") img9=cv2.imread('/home/pi/Selected/cropped/9.png' , cv2.IMREAD_GRAYSCALE) pixel_9 = np.sum(img9 == 255) jumlah_9 = pixel_9/12000 luas_9 = jumlah_9*100 desimal_luas_9 = round(luas_9) </pre>	
---	--

<pre>print('Luas Tanaman 9:', desimal_luas_9, "%") print('Waktu: ' , hasil)</pre>	
<pre>database = firebase.database() storage = firebase.storage() database.child("Tan aman").child("Tanam an 1").child("Kematang an").set(desimal_lu as_1) database.child("Tan aman").child("Tanam an 1").child("Waktu"). set(hasil) database.child("Tan aman").child("Tanam an 1").child("Kelembap an").set(line) database.child("Tan aman").child("Tanam an 2").child("Kematang an").set(desimal_lu</pre>	Merupakan perintah untuk mengunggah data hasil pembacaan pixel ke realtime database firebase.

<pre> as_2) database.child("Tan aman").child("Tanam an 2").child("Waktu"). set(hasil) database.child("Tan aman").child("Tanam an 2").child("Kelembap an").set(line) database.child("Tan aman").child("Tanam an 3").child("Kematang an").set(desimal_lu as_3) database.child("Tan aman").child("Tanam an 3").child("Waktu"). set(hasil) database.child("Tan aman").child("Tanam an 3").child("Kelembap an").set(line) database.child("Tan aman").child("Tanam an 4").child("Kematang an").set(desimal_lu </pre>	
--	--

<pre> as_4) database.child("Tan aman").child("Tanam an 4").child("Waktu"). set(hasil) database.child("Tan aman").child("Tanam an 4").child("Kelembap an").set(line) database.child("Tan aman").child("Tanam an 5").child("Kematang an").set(desimal_lu as_5) database.child("Tan aman").child("Tanam an 5").child("Waktu"). set(hasil) database.child("Tan aman").child("Tanam an 5").child("Kelembap an").set(line) database.child("Tan aman").child("Tanam an 6").child("Kematang an").set(desimal_lu </pre>	
--	--

<pre> as_6) database.child("Tan aman").child("Tanam an 6").child("Waktu"). set(hasil) database.child("Tan aman").child("Tanam an 6").child("Kelembap an").set(line) database.child("Tan aman").child("Tanam an 7").child("Kematang an").set(desimal_lu as_7) database.child("Tan aman").child("Tanam an 7").child("Waktu"). set(hasil) database.child("Tan aman").child("Tanam an 7").child("Kelembap an").set(line) database.child("Tan aman").child("Tanam an 8").child("Kematang an").set(desimal_lu </pre>	
--	--

<pre> as_8) database.child("Tan aman").child("Tanam an 8").child("Waktu"). set(hasil) database.child("Tan aman").child("Tanam an 8").child("Kelembap an").set(line) database.child("Tan aman").child("Tanam an 9").child("Kematang an").set(desimal_lu as_9) database.child("Tan aman").child("Tanam an 9").child("Waktu"). set(hasil) database.child("Tan aman").child("Tanam an 9").child("Kelembap an").set(line </pre>	
--	--

Dari coding diatas, data – data akan terunggah pada Realtime Database Firebase dan hasil datanya akan dapat dilihat pada website **monitoringtanamanselada.web.app**.