

PURWARUPA PERANGKAT IOT UNTUK SMART GREENHOUSE BERBASIS MIKROKONTROLER

Aghi Wardani¹, Dr. Kemas Muslim Lhaksana²

^{1,2}Fakultas Informatika, Universitas Telkom, Bandung

¹aghiwardani@students.telkomuniversity.ac.id, ²kemasmuslim@telkomuniversity.ac.id

Abstrak

Internet of Things (IoT) merupakan suatu konsep yang bertujuan untuk memperluas manfaat dari konektivitas Internet yang tersambung secara terus menerus. IoT bisa dimanfaatkan pada *greenhouse* untuk mengendalikan peralatan elektronik seperti *air cooler* dan pompa air yang dapat dioperasikan secara otomatis. Selain itu dengan memonitoring secara *real time* terhadap suhu udara, kelembaban udara, kelembaban tanah, dan intensitas cahaya yang terdapat di dalam *greenhouse*, berbagai tanaman di dalam *greenhouse* dapat tumbuh dengan optimal. Sistem yang digunakan dalam tugas akhir ini menggunakan NodeMCU ESP8266 sebagai pusat kontrol dan menggunakan DHT11, Soil Moisture, dan LDR sebagai sensor untuk mengukur IoT suhu udara, kelembaban udara, kelembaban tanah, dan intensitas cahaya di dalam *greenhouse*. Sebagai kendali di dalam *greenhouse* terdapat dua output kendali, yaitu *air cooler* dan pompa air. NodeMCU akan membaca suhu udara, kelembaban udara, kelembaban tanah, dan intensitas cahaya yang dikirim dari modul DHT11 yang akan menentukan apakah *air cooler* dan pompa air akan menyala atau tidak. Semua output yang diterima oleh NodeMCU akan dikirim ke server yang sebelumnya diproses pada halaman web yang dibuat menggunakan bahasa PHP. Selanjutnya pada saat output sudah diterima pada alamat web yang dituju maka output tersebut akan dikirim ke database MySQL. Sehingga web dapat menampilkan suhu udara, kelembaban udara, kelembaban tanah, dan intensitas cahaya *greenhouse* secara *real-time*.

Kata kunci : IoT (Internet of Things), Greenhouse, NodeMCU, DHT11, Kendali, Monitoring

Abstract

Internet of Things (IoT) is a concept that aims to expand the benefits of continuously connected Internet connectivity. IoT can be used in *greenhouse* to control electronic equipment such as water cooler and water pump which can be operated automatically. In addition, by real time monitoring on air temperature, air humidity, soil moisture, and light intensity contained in the *greenhouse*, plants in the *greenhouse* can grow optimally. The system used in this final project uses NodeMCU ESP8266 as the control center and uses DHT11, Soil Moisture, and LDR as sensors to measure air temperature, humidity, soil moisture, and light intensity in the *greenhouse*. as the control inside the *greenhouse* contains 2 output controls namely air cooler and water pump. NodeMCU will read air temperature, air humidity, soil moisture, and light intensity sent from DHT11 module which will determine whether water cooler and water pump will be on. All output received by NodeMCU will be sent to the server that was previously processed on web pages created using PHP language. Furthermore, when the output is received at the destination web address then the output will be sent to the MySQL database. So the Web can display air temperature, air humidity, soil moisture, and light intensity *greenhouse* in real-time.

Keywords: IoT (Internet of Things), Greenhouse, NodeMCU, DHT11, Control, Monitoring

1. Pendahuluan

Latar Belakang

Perkembangan kemajuan teknologi saat ini sudah sangat berkembang dengan pesat, tidak dapat dipungkiri kemajuan teknologi yang sedemikian cepat harus bisa dimanfaatkan, dipelajari serta diterapkan dalam kehidupan sehari-hari. Salah satu kemajuan yang bisa dirasakan adalah di bidang kendali, saat ini dengan adanya teknologi jaringan komputer yang sudah tumbuh pesat masalah hambatan jarak dan waktu dapat dipecahkan dengan solusi teknologi contohnya adalah penggunaan sistem komputer yang diterapkan dalam kehidupan sehari-hari. Penggunaan sistem komputer akan membuat kinerja dalam segi waktu mejadi lebih efektif.

Internet of Things (IoT) merupakan teknologi yang sedang banyak dibicarakan saat ini, karena terdapat penelitian yang mengungkapkan bahwa teknologi berbasis IoT akan menjadi satu hal yang besar dimasa yang akan datang dapat identifikasi segala sesuatu yang terhubung ke internet termasuk dalam teknologi berbasis IoT. Dengan kata lain, semua benda fisik atau pun virtual yang dapat dimonitor dan dikendalikan dari jarak jauh menggunakan internet termasuk teknologi IoT. IoT juga dapat digunakan untuk hal lain seperti pengambilan data dari satu tempat dengan menggunakan sensor dan juga akses jarak jauh untuk mengendalikan benda lain di satu tempat.

Greenhouse merupakan konstruksi yang berfungsi untuk menghindari dan memanipulasi kondisi lingkungan agar tercipta kondisi lingkungan yang dikehendaki dalam pemeliharaan tanaman yang nantinya tanaman akan lebih terkontrol dan pertumbuhan akan lebih maksimal dibandingkan dengan tanaman yang dibudidayakan diluar *greenhouse*. Namun pengendalian dan *monitoring* suhu dan kelembaban *greenhouse* masih menggunakan cara manual sehingga kualitas yang dihasilkan masih belum optimal. Tugas akhir ini akan merancang dan membangun sistem kendali *Smart Greenhouse* yang secara otomatis dapat mengendalikan suhu dan kelembaban udara serta dapat dimonitoring secara jarak jauh namun sistem pada Tugas akhir ini akan berfokus pada pengontrolan dan *monitoring* suhu udara, kelembaban udara, kelembaban tanah, dan intensitas cahaya pada *Smart greenhouse*.

Topik dan Batasannya

Berdasarkan latar belakang yang sudah dijelaskan di atas, permasalahan yang dapat diangkat, yaitu bagaimana merancang sebuah alat pengendali dan pemonitoring suhu udara, kelembaban udara, kelembaban tanah, dan intensitas cahaya berbasis *Internet of Things* menggunakan NodeMCU ESP8266 dengan sensor DHT11, Soil Moisture, dan LDR kemudian bagaimana cara untuk memonitoring suhu udara, kelembaban udara, kelembaban tanah, dan intensitas cahaya yang mampu menampilkan secara realtime berbasis Web.

Adapun lingkup yang menjadi batasan TA ini yaitu:

1. Mikrokontroler yang digunakan adalah NodeMCU ESP8266.
2. Alat sensor yang digunakan adalah DHT11, Soil Moisture, dan LDR kondisi yang di ukur adalah suhu, kelembaban udara, kelembaban tanah, dan Intensitas cahaya.
3. Sistem tidak melihat hambatan komunikasi dan keamanan jaringan.
4. *Greenhouse* yang digunakan merupakan prorotype dengan ukuran p x l x t = 25 cm x 16 cm x 16 cm.
5. Terdapat 2 output pengendali yaitu *Air cooler* & pompa air.

Tujuan

Adapun tujuan dari pembuatan TA ini yaitu Merancang dan membangun Sistem kendali dan *monitoring* menggunakan mikrokontroler NodeMCU ESP8266 dan sensor DHT11, Soil Moisture, dan LDR, berbasis *Internet of Things* (IoT) yang dapat memonitoring suhu udara, kelembaban udara, kelembaban tanah, dan intensitas cahaya secara cepat, tepat dan mampu mengendalikan suhu udara, kelembaban udara menggunakan *air cooler* dan melakukan penyiraman secara otomatis.

2. Studi Terkait

Sistem *Greenhouse*

Greenhouse dirancang dalam bentuk yang berbeda untuk kondisi iklim yang berbeda. Satu tanaman memiliki syarat-syarat kondisi tertentu yang merupakan kondisi yang membantu tanaman tersebut untuk tumbuh subur dan lebih produktif. Penyesuaian iklim di dalam *greenhouse* seharusnya dapat dioptimalkan melalui sistem yang dapat membuatnya sama dengan iklim yang dibutuhkan dalam menanam tanaman tersebut. [3,6]

Temperatur / Suhu

Temperatur atau suhu sangat berpengaruh pada pertumbuhan tanaman, beberapa proses pada tanaman yang dipengaruhi oleh suhu adalah proses transpirasi, proses foto sintesis, serta proses respirasi pada tanaman. Saat temperatur atau suhu tersebut dapat dijaga dengan baik maka pertumbuhan tanaman akan lebih maksimal. [3,6]

Kelembaban udara dan tanah

Kelembaban udara & tanah sangat erat kaitannya dengan kadar air yang dibutuhkan oleh tanaman, semakin lembab kondisinya pertumbuhan tanaman akan lebih baik dan maksimal. Kelembaban berbanding terbalik dengan temperatur/suhu, semakin tinggi suhunya semakin kecil nilai kelembabannya begitupun sebaliknya. Pengaruh kelembaban pada tanaman hampir sama seperti suhu karena pada dasarnya tanaman sangat membutuhkan air. [3,6]

Intensitas Cahaya

Intensitas cahaya adalah banyaknya energi yang diterima oleh suatu tanaman per satuan luas dan per satuan waktu ($\text{kal}/\text{cm}^2/\text{hari}$). Pengertian intensitas disini sudah termasuk di dalamnya lama penyinaran, yaitu lama matahari bersinar dalam satu hari, karena satuan waktunya menggunakan hari. Intensitas cahaya dan lamanya penyinaran mempengaruhi sifat tanaman. Pengaruh intensitas cahaya terhadap pertumbuhan dan perkembangan tanaman sejauh mana berhubungan erat dengan proses Dalam proses ini energi cahaya diperlukan untuk berlangsungnya penyatuan CO_2 dan air untuk membentuk karbohidrat. Semakin besar jumlah energi yang tersedia akan memperbesar jumlah hasil fotosintesis sampai dengan maksimum.

Mikrokontroler NodeMCU ESP8266

NodeMCU ESP8266 merupakan sebuah papan mikrokontroler yang berbasis modul WiFi ESP8266 yang dikemas ke dalam modul yang dikhususkan untuk mengakses modul sensor maupun modul mikrokontroler lainnya, yang dapat dikendalikan atau monitoring melalui internet.[4,9]

Sensor Suhu dan Kelembaban (DHT11)

Sensor DHT11 adalah sensor suhu dan kelembaban dimana keluaran dari sensor ini berupa sinyal digital. Sensor ini memerlukan catu daya sebesar 3V hingga 5V DC. Resolusi dari DHT11 untuk suhu adalah 8 bit, akurasi minimum ± 1 °C dan akurasi maksimum ± 2 °C dengan rentang pengukuran suhu dari 0 °C sampai dengan 50 °C. Adapun kelembaban udara yang dapat diukur berkisar antara 20 hingga 90% dengan tingkat presisi $\pm 5\%$. [2,8,10]

Sensor Kelembaban Tanah (Soil Moisture Sensor)

Sensor kelembaban tanah terdiri dari dua probe yang digunakan untuk mengukur kandungan volumetrik air. Kedua probe memungkinkan arus melewati tanah dan kemudian mendapat nilai resistansi untuk mengukur nilai kelembaban. Ketika ada lebih banyak air, tanah akan melakukan lebih banyak listrik yang berarti bahwa akan ada lebih sedikit perlawanan. Oleh karena itu, tingkat kelembapannya akan lebih tinggi. Tanah kering melakukan listrik dengan buruk, sehingga ketika akan ada lebih sedikit air, maka tanah akan melakukan lebih sedikit listrik yang berarti akan ada lebih banyak perlawanan. Oleh karena itu, tingkat kelembapannya akan lebih rendah.

Sensor Cahaya (LDR)

Light Dependent Resistor atau disingkat dengan LDR adalah jenis resistor yang nilai hambatan atau nilai resistansinya tergantung pada intensitas cahaya yang diterimanya. Nilai hambatan LDR akan menurun pada saat cahaya terang dan nilai hambatannya akan menjadi tinggi jika dalam kondisi gelap. Dengan kata lain, fungsi LDR (*Light Dependent Resistor*) adalah untuk menghantarkan arus listrik jika menerima sejumlah intensitas cahaya (Kondisi Terang) dan menghambat arus listrik dalam kondisi gelap. [11]

Komponen pengendali

Greenhouse ini memiliki 2 komponen utama sebagai pengendali untuk kondisi di dalam *greenhouse*, kedua komponen tersebut adalah *air cooler* dan pompa air. *Air cooler* merupakan kipas angin yang bisa mendinginkan ruangan yang panas menjadi sejuk. Pompa air merupakan alat yang digunakan menyiramkan air pada tanaman sehingga kelembaban tanah di sekitar tanaman menjadi basah/lembab.

Web Server

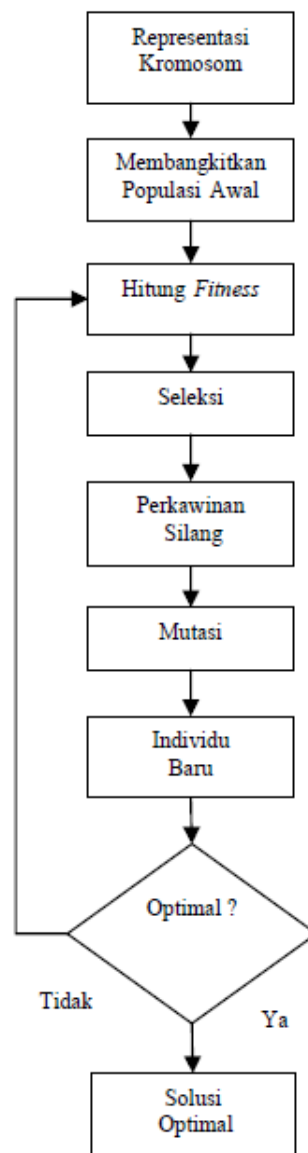
Web server merupakan perangkat lunak yang menyediakan layanan akses kepada pengguna melalui protokol komunikasi HTTP atau HTTPS atas berkas-berkas yang terdapat pada suatu situs web. Perangkat lunak ini di konfigurasi di dalam sebuah komputer yang nantinya akan menjadi komputer server. Komputer yang di rekomendasikan untuk menjadi server web ini harus mempunyai spesifikasi khusus dan computer nantinya ini dikhususkan untuk menaruh datawebsite.

Genetic Algorithm (GA)

Algoritma genetika adalah algoritma komputasi yang diinspirasi teori evolusi yang kemudian diadopsi menjadi algoritma komputasi untuk mencari solusi suatu permasalahan dengan cara yang lebih "alamiah". Salah satu aplikasi algoritma genetika adalah pada permasalahan optimasi kombinasi, yaitu mendapatkan suatu nilai solusi optimal terhadap suatu permasalahan yang mempunyai banyak kemungkinan solusi.

Pada algoritma ini, teknik pencarian dilakukan sekaligus atas sejumlah solusi yang mungkin dikenal dengan istilah populasi. Individu yang terdapat dalam satu populasi disebut dengan istilah kromosom. Kromosom ini merupakan suatu solusi yang masih berbentuk simbol. Populasi awal dibangun secara acak, sedangkan populasi selanjutnya merupakan hasil evolusi kromosom-kromosom melalui iterasi yang disebut dengan istilah generasi. Pada setiap generasi kromosom-kromosom tersebut dievaluasi tingkat keberhasilan nilai solusinya terhadap masalah yang ingin diselesaikan menggunakan ukuran yang disebut dengan fitness. Untuk memilih kromosom yang tetap dipertahankan untuk generasi selanjutnya dilakukan proses yang disebut dengan seleksi. Proses seleksi kromosom menggunakan konsep aturan evolusi yang telah disebutkan sebelumnya yaitu kromosom yang mempunyai nilai fitness tinggi akan memiliki peluang lebih besar untuk terpilih lagi pada generasi selanjutnya. kromosom-kromosom baru yang disebut dengan offspring, dibentuk dengan cara melakukan perkawinan antar kromosom-kromosom dalam satu generasi yang disebut sebagai proses perkawinan silang. Mekanisme perubahan susunan unsur penyusun makhluk hidup akibat adanya faktor alam yang disebut dengan mutasi direpresentasikan sebagai proses berubahnya satu atau lebih nilai gen dalam kromosom dengan suatu nilai acak. Setelah beberapa generasi akan dihasilkan kromosom-kromosom yang nilai gen-gennya konvergen ke suatu nilai tertentu yang

merupakan solusi terbaik yang dihasilkan oleh algoritma genetika terhadap permasalahan yang ingin diselesaikan. [12,13]

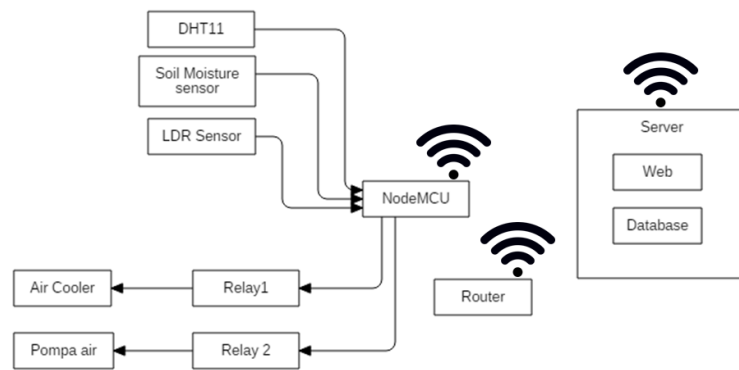


Gambar 1 Diagram Alur *Genetic Algorithms* sederhana.

3. Sistem yang Dibangun

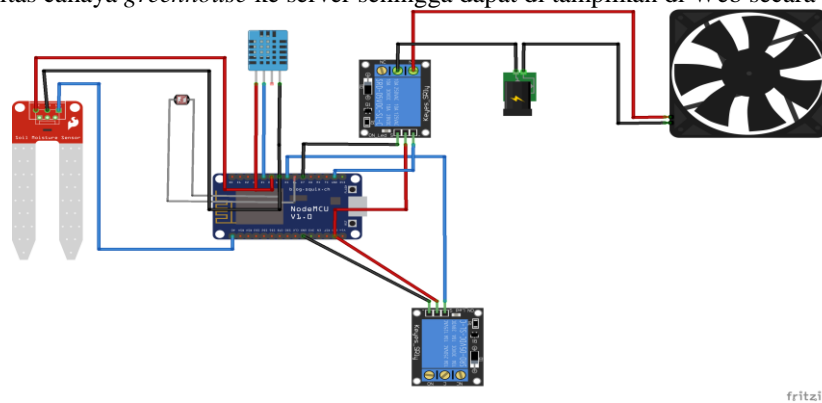
Perancangan sistem *Smart greenhouse* dibagi menjadi dua bagian utama, yaitu:

- Perancangan *hardware* yang terdiri dari mikrokontroler dan perangkat pendukung seperti, sensor suhu udara dan kelembaban udara (DHT11), sensor kelembaban tanah (soil moisture sensor), dan sensor intensitas cahaya (LDR) serta perangkat pengendali seperti air cooler dan pompa air.
- Perancangan *software* yang terdiri dari pembuatan Web.



Gambar 2 Diagram blok sistem

Perancangan *hardware* meliputi NodeMCU, DHT11, soil moisture sensor, LDR, *air cooler*, pompa air dan design Smart greenhouse. NodeMCU di sini berfungsi sebagai pusat kontrol dan *client* dan pada saat awal diaktifkan akan mencari koneksi router. NodeMCU akan membaca suhu udara, kelembaban udara, kelembaban tanah, dan intensitas cahaya yang dikirim dari modul sensor yang akan menentukan apakah *air cooler* dan pompa air akan menyala atau tidak, Serta NodeMCU akan mengirim nilai suhu udara, kelembaban udara, kelembaban tanah, dan intensitas cahaya *greenhouse* ke server sehingga dapat di tampilkan di Web secara *real time*.



Gambar 3 Rangkaian *Smart Greenhouse* dengan NodeMCU

Semua *output* yang diterima oleh NodeMCU akan dikirim ke server yang sebelumnya diproses pada halaman Web yang dibuat menggunakan bahasa PHP. Selanjutnya pada saat *output* sudah diterima pada alamat web yang dituju maka *output* tersebut akan dikirm ke database MySQL. Sehingga Web dapat menampilkan suhu dan kelembaban *greenhouse* secara *real time*.

Dalam penelitian ini terdapat data yang telah disiapkan untuk kebutuhan mencari nilai optimal kecepatan kipas dengan *genetic algorithm*. Nilai optimal yang akan dicari adalah kecepatan kipas dengan faktor yang mempengaruhinya yaitu suhu dan kelembaban udara. Data ini didapatkan dari situs bmgk.go.id, dikarenakan kecepatan angin di dalam ruangan yang dihasilkan oleh kipas tidak dapat mencapai 6 knot maka kecepatan angin dalam data dirubah menjadi nilai tertinggi adalah 3 dan terendah 1, dan data menjadi tidak asli atau dummy.

Bulan	Tahun	kelembapan	angin	temperatur
JANUARI	2005	83	3	23,3
FEB		85	3	23,1
MARET		84	2	23,6
APRIL		83	2	23,7
MEI		82	2	23,8
JUNI		85	2	23,4
JULI		80	2	22,8

AGUSTUS		77	2	23,4
SEPT		79	2	23,6
OKT		81	2	23,5
NOV		81	2	23,4
DES		84	2	23,2
JANUARI	2006	85	3	23,1
FEB		83	3	23,5
MARET		82	3	23,9
APRIL		83	3	23,5
MEI		80	2	23,3
JUNI		77	3	22,7
JULI		78	3	23
AGUSTUS		76	3	22,6
SEPT		73	3	23,6
OKT		72	3	24,4
NOV		78	2	24,8
DES		88	1	23,2

Tabel 1 Tabel sample data

Representasi Kromosom dari data yang digunakan

$$ax + by = z$$

x = suhu udara

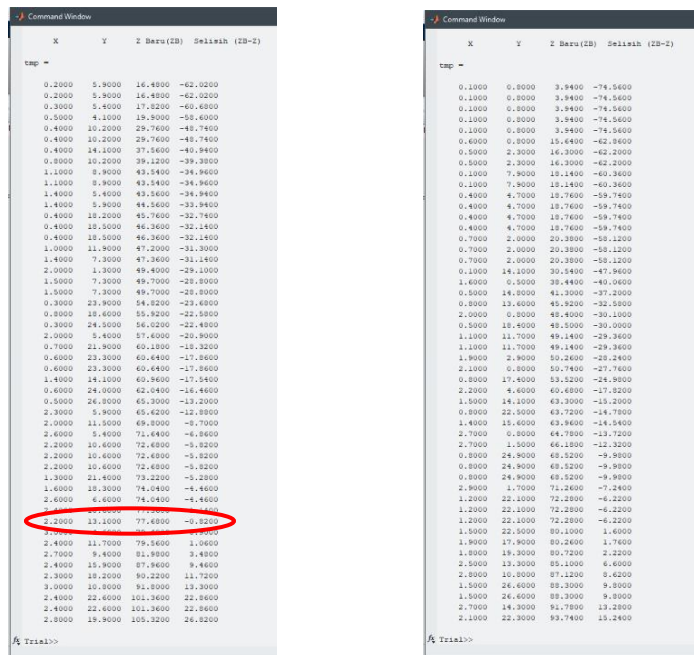
y = kecepatan angin

z = kelembaban udara

a = gen suhu udara

b = gen kecepatan angin

setelah mendapatkan fungsi kromosomnya, kemudian di hitung menggunakan algoritma GA dengan MATLAB, dari dua kali percobaan didapat hasil.



Gambar 4 Hasil Iterasi GA menggunakan MATLAB

Didapatkan nilai Kromosom terbaik

$$[2,2 ; 13,1]$$

Karena yang akan dicari adalah nilai kecepatan angin maka fungsi yang ada dirubah menjadi:

$$\frac{z - ax}{b} = y$$

$$\frac{z - 2,2x}{13,1} = y$$

Sebagai ujicoba apakah nilai kromosom yang di cari sudah benar maka dilakukan ujicoba dengan memasukan nilai x dan z dengan rata-rata suhu udara dan rata-rata kelembaban udara.

$$\frac{78,5208333 - 2,2(23,4333333)}{13,1} = 2$$

Dari percobaan diatas didapat nilai kecepatan udara = 2 sudah sama dengan nilai rata-rata kecepatan udara, maka dapat disimpulkan bahwa nilai kromosom yang menjadi solusi optimal sudah tepat.

4. Evaluasi

Pada Bab ini *prototype* yang telah di bangun kemudian dilakukan pengujian dan analisis guna untuk mengetahui kinerja sistem *Smart Greenhouse* dengan menggunakan *Client Server* melalui komunikasi *Wireless*. Pengujian berupa pengolahan *hardware* dan *software* yang telah terintegrasi satu sama lain.

4.1 Hasil Pengujian

4.1.1 Pengujian NodeMCU ESP8266

NodeMCU ESP8266 merupakan pengendali utama dari *hardware* yang di buat. Pengujian terhadap NodeMCU ESP8266 ini yaitu untuk mengetahui apakah mikrokontroler ini dapat digunakan dengan baik atau tidak. Cara menguji *hardware* ini yaitu dengan memeriksa setiap pin *input* dan *output* yang terdapat pada NodeMCU ESP8266 yang sebelumnya telah dipasang program pada setiap pin yang digunakan. Hasil pengujian menunjukkan bahwa NodeMCU yang sebelumnya telah diprogram dan disambungkan dengan *hardware* lainnya dapat berjalan

4.1.2 Pengujian Sensor

Pada pengujian sensor dilakukan dengan beberapa cara yaitu untuk pengujian DHT11 dilakukan dengan cara memberikan udara panas kepada DHT11 sehingga udara disekitar DHT11 meningkat. Untuk soil moisture sensor dengan cara membaca kelembaban tanah pada tanah basah dan kering yang di tempatkan pada botol plastik. Dan pengujian LDR sensor dilakukan pengujian dengan cara menyinari sensor dengan senter. Hasil pengujian dari tiap sensor ini menunjukkan bahwa DHT11, soil moisture sensor, dan LDR sensor dapat berfungsi dengan baik dan dapat mengukur suhu udara, kelembaban udara, kelembaban tanah, dan intensitas cahaya dengan benar.



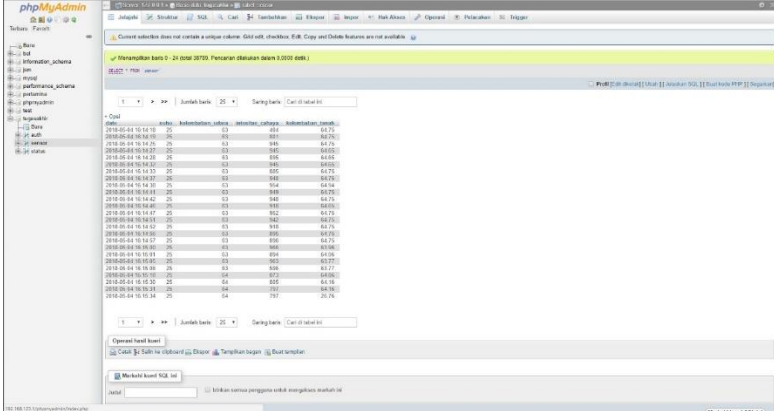
Gambar 5 Hasil pengujian Sensor

4.1.3 Pengujian Controller

Pada pengujian *controller* dilakukan pengujian dengan cara memberikan perintah manual kepada mikrokontroler sehingga kipas dan pompa air dapat dioperasikan secara manual, Hasil pengujian ini menunjukkan bahwa *controller* yang digunakan dapat berfungsi dengan baik.

4.1.4 Pengujian Pengiriman Data dari NodeMCU ke server

Pengujian Pengiriman data dari NodeMCU meliputi data dari sensor suhu dan kelembaban udara DHT11, kelembaban tanah (soil moisture sensor), dan intensitas cahaya (LDR). data dari sensor telah berhasil disimpan ke dalam database yang ditunjukkan oleh gambar:



id	date	suhu	kelembaban_udara	intensitas_cahaya	kelembaban_tanah
2018-08-01 14:14	2018-08-01 14:14:25	33	68	846	64.75
2018-08-01 14:16	2018-08-01 14:16:25	32	69	846	64.75
2018-08-01 14:18	2018-08-01 14:18:25	33	68	846	64.75
2018-08-01 14:20	2018-08-01 14:20:25	33	68	846	64.75
2018-08-01 14:22	2018-08-01 14:22:25	33	68	846	64.75
2018-08-01 14:24	2018-08-01 14:24:25	33	68	846	64.75
2018-08-01 14:26	2018-08-01 14:26:25	33	68	846	64.75
2018-08-01 14:28	2018-08-01 14:28:25	33	68	846	64.75
2018-08-01 14:30	2018-08-01 14:30:25	33	68	846	64.75
2018-08-01 14:32	2018-08-01 14:32:25	33	68	846	64.75
2018-08-01 14:34	2018-08-01 14:34:25	33	68	846	64.75
2018-08-01 14:36	2018-08-01 14:36:25	33	68	846	64.75
2018-08-01 14:38	2018-08-01 14:38:25	33	68	846	64.75
2018-08-01 14:40	2018-08-01 14:40:25	33	68	846	64.75
2018-08-01 14:42	2018-08-01 14:42:25	33	68	846	64.75
2018-08-01 14:44	2018-08-01 14:44:25	33	68	846	64.75
2018-08-01 14:46	2018-08-01 14:46:25	33	68	846	64.75
2018-08-01 14:48	2018-08-01 14:48:25	33	68	846	64.75
2018-08-01 14:50	2018-08-01 14:50:25	33	68	846	64.75
2018-08-01 14:52	2018-08-01 14:52:25	33	68	846	64.75
2018-08-01 14:54	2018-08-01 14:54:25	33	68	846	64.75
2018-08-01 14:56	2018-08-01 14:56:25	33	68	846	64.75
2018-08-01 14:58	2018-08-01 14:58:25	33	68	846	64.75
2018-08-01 15:00	2018-08-01 15:00:25	33	68	846	64.75

Gambar 6 Record sensor pada database

Pada gambar dapat dilihat status dari kondisi sensor yang berubah ketika mendeteksi perubahan keadaan. Untuk kondisi kipas tidak dikirim ke database namun kipas langsung aktif ketika suhu lebih besar sama dengan $30^{\circ}C$. Data yang telah dikirim akan ditampilkan dalam bentuk halaman Web dan akan menampilkan data-data yang telah disimpan ke dalam database agar pengguna dapat mengetahui keadaan greenhouse dari setiap kondisi yang ditunjukkan seperti pada gambar:

4.1.5 Pengujian Pengiriman Data dari server ke NodeMCU

Pengujian penerimaan data dari server ke NodeMCU dilakukan untuk mengetahui keberhasilan pengontrolan *Air Cooler* dan pompa air, sistem *controlling* dapat di atur secara otomatis ataupun manual Jika pengguna memilih mode otomatis maka kipas dan pompa air akan beroperasi secara otomatis sesuai kondisi yang sudah di tetapkan, jika pengguna memilih mode manual maka pengguna dapat mengoperasikan kipas dan pompa air secara manual dengan kondisi yang dihendaki oleh pengguna. Hasil pengujian ini menunjukkan bahwa NodeMCU sudah dapat menerima data perintah yang telah dikirimkan dari server dengan baik.

4.2 Analisis Hasil Pengujian

Setelah seluruh komponen terintegrasi secara menyeluruh satu sama lain dan terkoneksi terhadap server. Kemudian dilakukan pengujian secara keseluruhan apakah sistem dapat bekerja dengan baik dan telah tersinkronisasi satu dengan yang lainnya. Dari pengujian keseluruhan ini didapatkan hasil bahwa kipas yang digunakan sebagai kontroler dapat bekerja dengan baik dan dapat beroperasi secara otomatis dengan cara membaca nilai suhu dan kelembaban udara pada database, bila mana nilai suhu dan kelembaban udara melebihi nilai yang telah ditetapkan, kipas akan bekerja sehingga dapat membuat suhu dan kelembaban udara di dalam *greenhouse* menjadi lebih stabil. Pompa air pada purwarupa *Smart greenhouse* ini pun dapat bekerja dengan baik sehingga dapat menyirami tanaman sesuai waktu yang telah ditentukan.

Kekurangan dari *prototype Smart greenhouse* ini adalah tidak dapat mengatasi suhu dan kelembaban udara yang rendah karena kurangnya kontroler pengendali.

5. Kesimpulan

Berdasarkan hasil analisis pengujian yang dilakukan maka didapatkan kesimpulan bahwa sistem dan *hardware* dapat bekerja dengan baik tetapi masih memiliki banyak kekurangan seperti kurangnya kontroler yang dapat mengatur intensitas cahaya, kontroler yang dapat menaikkan suhu udara dan mengatur kelembaban udara secara tepat karena sistem yang telah dibangun tidak dapat mengatasi kondisi ekstrim seperti suhu dan kelembaban yang sangat rendah. Alasan ini dikarenakan kurangnya pin pada nodemcu untuk dihubungkan kepada kontroler. Untuk mengatasi masalah ini disarankan menggunakan lebih dari 1 nodeMCU sehingga banyaknya pin dapat bertambah dan dapat menggunakan lebih banyak kontroler.

Daftar Pustaka

- [1] Kadir, A. (2015). Buku Pintar Pemrograman Arduino, Yogyakarta, Indonesia.
- [2] Kadir, A. (2015). Arduino, Panduan Mempelajari Aneka Proyek Berbasis Mikrokontroler, Yogyakarta, Indonesia.
- [3] Abraham Lono, L. (2016). Smart Greenhouse Berbasis Mikrokontroler Arduino Mega 2650 REV 3, Universitas Sanata Dharma, Indonesia.
- [4] Fajar Wicaksono, M. (2017). Implementasi Modul Wifi NodeMCU ESP8266 Untuk Smart Home. UNIKOM, Indonesia.
- [5] Grunert, J. How to Use a Greenhouse, [Online], Available at: http://garden.lovetoknow.com/wiki/How_to_Use_a_Greenhouse [Accessed, 28 Oktober 2017].
- [6] ROL Staff. How to Keep a Greenhouse Going, [Online], Available at: <https://www.rodaleorganiclife.com/garden/tending-greenhouse> [Accessed, 28 Oktober 2017].
- [7] Moore, Sarah. What Is the Ideal Humidity Level for a Greenhouse? [Online], Available at: <http://homeguides.sfgate.com/ideal-humidity-level-greenhouse-77050.html> [Accessed, 28 Oktober 2017].
- [8] Ajie, Saptaji. (2016). Mengukur Suhu dan Kelembaban Udara Dengan Sensor DHT11 dan Arduino. [Online]. Available at: <http://saptaji.com/2016/08/10/mengukur-suhu-dan-kelembaban-udara-dengan-sensor-dht11-dan-arduino/> [Accessed, 29 Oktober 2017].
- [9] Jimmy, Mada. (2016). NodeMCU ESP8266 dan papan informasi berbasis MAX7219 [Online]. Available at: <http://www.madajimmy.com/artikel/proyek/69-node-mcu-esp8266-dan-papan-informasi-berbasis-max7219-.html> [Accessed, 29 Oktober 2017].
- [10] Nyebarilmu. (2017). Monitoring sensor DHT11 dengan NodeMCU ESP8266 via Browser. [Online]. Available at: <https://www.nyebarilmu.com/monitoring-sensor-dht11-dengan-nodemcu-esp8266-via-browser/> [Accessed, 29 Oktober 2017].
- [11] Kho, Dickson. (2016). Pengertian LDR (Light Dependent Resistor) dan Cara Mengukurnya. [Online]. Available at: <https://teknikelektronika.com/pengertian-ldr-light-dependent-resistor-cara-mengukur-ldr/> [Accessed, 13 Mei 2018].
- [12] Hermawanto, Denny. (2016). Algoritma Genetika Dan Contoh Aplikasinya. [Online]. Available at: <http://www.firman-its.com/2007/05/17/algoritma-genetika-dan-contoh-aplikasinya/> [Accessed, 13 Mei 2018].
- [13] Fauzi, Rahmat. (2015). Optimasi Penjadwalan Mata Kuliah Dengan Menggunakan Algoritma Genetika (Studi Kasus : Program Studi Teknik Informatika) (Universitas Darma Persada). Universitas Darma Persada Jakarta.

Lampiran

Program NodeMCU

```

#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <ArduinoJson.h>

#include "DHT.h" // including the library of DHT11 temperature and humidity sensor

StaticJsonBuffer<256> jsonBuffer;

int pinKipasBesar = 12; //D6
int pinKipasKecil = 14; //D5
int pinAir = 13; //D7
int pinMultiplexer = 5; //D1
int pinDHT11 = 2; //D4

DHT dht(pinDHT11, DHT11);

const char* ssid = "openBel";
const char* password = "antihack22";
float kelembaban_tanah;
float intensitas_cahaya;
int kipas,air,second;

void setup () {
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print("Connecting to ");
    Serial.println(ssid);
  }
  Serial.print("Connected to ");
  Serial.println(ssid);
  dht.begin();
  pinMode(pinMultiplexer, OUTPUT);
  pinMode(pinKipasBesar, OUTPUT);
  pinMode(pinKipasKecil, OUTPUT);
  pinMode(pinAir, OUTPUT);
}

void loop() {
  float kelembaban_udara = dht.readHumidity();
  float suhu = dht.readTemperature();
  int sensorValue = analogRead(A0);

  digitalWrite(D1,LOW);
  delay(500);
  kelembaban_tanah = 0;
  kelembaban_tanah = analogRead(A0);
  kelembaban_tanah = kelembaban_tanah/1024*100;

  digitalWrite(D1,HIGH);
  delay(500);
  intensitas_cahaya = 0;
  intensitas_cahaya = analogRead(A0);
  if(isnan(suhu))

```

```

Serial.println("Suhu Isnan");
if(kelembaban_tanah > 10 && intesitas_cahaya > 10)
Serial.println("kelembaban_tanah && intesitas_cahaya < 10");

if(!isnan(suhu)&& !isnan(kelembaban_udara) && !isnan(intesitas_cahaya) && !isnan(kelembaban_tanah)){
if(kelembaban_tanah > 10 && intesitas_cahaya > 10){
if (WiFi.status() == WL_CONNECTED) {
HTTPClient http;
Serial.print("Kelembaban Tanah : ");
Serial.println(kelembaban_tanah);
Serial.print("Intesitas Cahaya : ");
Serial.println(intesitas_cahaya);
Serial.print("suhu : ");
Serial.println(suhu);
Serial.print("kelembaban_udara : ");
Serial.println(kelembaban_udara);

String url =
String("http://192.168.123.1/oghi/index.php/proses/uploudData?suhu="+String(suhu)+"&kelembaban_udara="+
String(kelembaban_udara)+"&intesitas_cahaya="+String(intesitas_cahaya)+"&kelembaban_tanah="+String(kelembaban_tanah));
http.begin(url);
int httpCode = http.GET();
if (httpCode > 0) {
Serial.println("Isi Web : ");
String payload = http.getString();
Serial.println(payload);
JsonObject& dataDariWeb = jsonBuffer.parseObject(payload);
kipas = dataDariWeb["kipas"];
air = dataDariWeb["air"];
SpeedKipas(kipas);
AirOnOFF(air);
}
http.end();
}
}
jsonBuffer.clear();
}
}

void SpeedKipas(int speed){
Serial.print("Speed Kipas ");
Serial.println(speed);
if(speed == 1){
digitalWrite(pinKipasBesar,HIGH);
digitalWrite(pinKipasKecil,HIGH);
}else if(speed == 2){
digitalWrite(pinKipasBesar,HIGH);
digitalWrite(pinKipasKecil,LOW);
}else if(speed == 3){
digitalWrite(pinKipasBesar,LOW);
digitalWrite(pinKipasKecil,HIGH);
}
}

void AirOnOFF(int air){
Serial.print("Air : ");
Serial.println(air);
if(air == 0){
digitalWrite(pinAir,HIGH);
}else if(air == 1){
digitalWrite(pinAir,LOW);
}
}

```

```

    delay(3000);
    digitalWrite(pinAir,HIGH);
  }
}

```

Data Cuaca

Bulan	Tahun	kelembapan	angin	temperatur
JANUARI	2005	83	6	23,3
FEB		85	6	23,1
MARET		84	4	23,6
APRIL		83	4	23,7
MEI		82	3	23,8
JUNI		85	4	23,4
JULI		80	4	22,8
AGUSTUS		77	4	23,4
SEPT		79	3	23,6
OKT		81	4	23,5
NOV		81	4	23,4
DES		84	4	23,2
JANUARI	2006	85	6	23,1
FEB		83	6	23,5
MARET		82	5	23,9
APRIL		83	5	23,5
MEI		80	4	23,3
JUNI		77	5	22,7
JULI		78	5	23
AGUSTUS		76	5	22,6
SEPT		73	6	23,6
OKT		72	6	24,4
NOV		78	4	24,8
DES		88	2	23,2
JANUARI	2007	77	4	24,1
FEB		87	2	22,8
MARET		83	4	23,4
APRIL		88	3	22,9
MEI		82	3	23,6
JUNI		83	3	23,1
JULI		81	3	23,3
AGUSTUS		73	3	23,6
SEPT		72	3	24,4
OKT		73	3	24,5
NOV		87	3	23,6
DES		86	3	22,9
JANUARI	2008	77	4	23,7
FEB		83	4	22,5
MARET		82	3	22,8
APRIL		84	3	22,9
MEI		77	2	23
JUNI		78	1	22,7
JULI		73	2	22,7
AGUSTUS		77	2	23,1
SEPT		72	2	24,2
OKT		77	2	24
NOV		82	1	23,1
DES		82	2	23,4
JANUARI	2009	80	3	23,6
FEB		81	4	23,1
MARET		83	4	22,9
APRIL		83	3	23,4

MEI		80	3	23,6
JUNI		80	3	23,4
JULI		73	3	23
AGUSTUS		71	3	23,6
SEPT		70	3	24,4
OKT		76	3	23,4
NOV		81	3	23,3
DES		82	3	23,5
JANUARI	2010	84	2	22,9
FEB		87	3	23,2
MARET		86	3	23,1
APRIL		78	3	24,6
MEI		83	3	24
JUNI		84	3	23,3
JULI		85	3	22,9
AGUSTUS		81	3	23,3
SEPT		85	3	22,9
OKT		82	3	23,2
NOV		85	3	23,3
DES		82	4	23
JANUARI	2011	79	4	23
FEB		78	5	23,5
MARET		77	3	23,5
APRIL		79	3	23,4
MEI		80	3	23,6
JUNI		75	3	23,3
JULI		74	3	22,8
AGUSTUS		69	3	23
SEPT		69	3	23,4
OKT		73	3	24,1
NOV		83	3	22,9
DES		79	3	24
JANUARI	2012	78	4	23,4
FEB		79	3	23,5
MARET		75	4	23,7
APRIL		81	3	23,5
MEI		77	3	23,2
JUNI		74	3	23,3
JULI		71	3	22,6
AGUSTUS		67	3	23,2
SEPT		67	3	24
OKT		74	3	24,2
NOV		83	3	23,3
DES		83	3	23,3
JANUARI	2013	79	4	23,4
FEB		80	4	23,4
MARET		79	4	23,8
APRIL		82	3	23,7
MEI		82	3	23,5
JUNI		78	3	23,6
JULI		77	3	22,5
AGUSTUS		71	3	23,2
SEPT		70	3	23,7
OKT		73	3	23,7
NOV		74	4	23,8
DES		80	3	23,1
JANUARI	2014	82	4	22,5
FEB		80	4	22,9
MARET		82	3	23,3
APRIL		81	3	23,7

MEI		81	3	23,5
JUNI		80	3	23,5
JULI		77	3	23
AGUSTUS		73	3	23,1
SEPT		64	3	23,7
OKT		67	4	24,2
NOV		79	3	23,6
DES		80	4	23,7
JANUARI	2015	79	4	23,2
FEB		80	5	23,1
MARET		74	5	23,4
APRIL		74	4	23,6
MEI		77	4	23,4
JUNI		74	4	23,1
JULI		70	4	23,1
AGUSTUS		66	4	23,6
SEPT		64	4	23,8
OKT		61	4	24,1
NOV		79	4	23,9
DES		82	4	23,7
JANUARI	2016	81	4	23,8
FEB		80	5	24
MARET		83	4	24,1
APRIL		84	4	24
MEI		83	4	24
JUNI		79	4	23,7
JULI		79	4	23,3
AGUSTUS		66	4	23,6
SEPT		80	4	23,6
OKT		83	4	23,1
NOV		84	4	23,4
DES		73	6	24,4

Program Matlab

```

clear;
clc;
% 1. Pembentukan kromosom

% aT + bV = k
ind = 2; % jumlah variabel
A = 2.34;
B = 0.58;
V = 50;
% 2. Inisialisasi
indC = 50; %jumlah indeks Chromosome
chromosome(:,1) = randi([0 20],indC,1);
chromosome(:,2) = randi([0 80],indC,1);
display("chromosome awal");
display(chromosome);

% 3. Evaluasi Chromosome (fungsi_obj)
fungsi_obj = zeros();
for i=1:indC
    fungsi_obj(i,1) = abs(A*chromosome(i,1)+B*chromosome(i,2)-V);
end
display("fungsi objektif");
display(fungsi_obj);

```

```

Rata2_fungsi_obj = mean(fungsi_obj);
display("rata-rata fungsi objektif");
display(Rata2_fungsi_obj);

jumlah_iterasi = 100;

for z=1:jumlah_iterasi
    % 4. Seleksi Chromosome (fitness)
    for i=1:indC
        fitness(i,1) = 1/(fungsi_obj(i)+1);
    end
    Total_fitness = sum(fitness);
    display(["fitness" z]);
    display(fitness);
    display(["Total fitness" z]);
    display(Total_fitness);

    % cari probabilitas (probabilitas)
    for i=1:indC
        probabilitas(i,1) = fitness(i)/Total_fitness;
    end
    display(["probabilitas" z]);
    display(probabilitas);

    % cari kumulatif probabilitas (C)
    C = zeros();
    for i=1:indC
        C(i,1) = 0;
        for j = 1:i
            C(i,1) = C(i,1) + probabilitas(j);
        end
    end
    display(["Kumulatif probabilitas" z]);
    display(C);
    % seleksi menggunakan roulette wheel (R)
    R = rand(indC,1);
    for i=1:indC
        if i == 1 && R(i)<C(i)
            R(i) = C(i);
        end
    end
    display(["roulette wheel" z]);
    display(R);
    %populasi chromosome
    for i=1:indC
        for j=1:indC
            if R(i)<C(j)
                break;
            end
        end
        populasi_chromosome(i,1:ind)=chromosome(j,:);
    end
    chromosome=populasi_chromosome;
    display(["chromosome roulette wheel" z]);
    display(chromosome);
    % 5. crossover

    % bangkitkan nilai acak R (R2)
    % crossover probability = 25%

```

```

cp = 0.25;
display(["crossover probability" z]);
display(cp);

% jika lebih dari 1 nilai di R2 kurang dari nilai di cp, maka CrossOver
R2 = rand(indC,1);
display(["peluang acak" z]);
display(R2);

indeks_crossOver = find(R2<cp);
sum_crossOver = sum(R2<cp); % jumlah R2 yang kurang dari cp
display(["indeks crossover" z]);
display(indeks_crossOver);
display(["Jumlah yang di lakukan cross over" z]);
display(sum_crossOver);

if sum_crossOver > 1
    % bangkitkan cut off (co)
    co = randi([0 ind-1],sum_crossOver,1);
    display(["cut off" z]);
    display(co);
    for i=1:sum_crossOver
        chromosome_orangtua_1 = chromosome(indeks_crossOver(i,:));
        if i ~= sum_crossOver
            chromosome_orangtua_2 = chromosome(indeks_crossOver(i+1,:));
        else
            chromosome_orangtua_2 = chromosome(indeks_crossOver(1,:));
        end
        if co(i) == 0
            newChromosome = chromosome_orangtua_2;
        elseif co(i) == ind
            newChromosome = chromosome_orangtua_1;
        else
            newChromosome = [chromosome_orangtua_1(1:co) chromosome_orangtua_2(co+1:end)];
        end
        chromosome(indeks_crossOver(i,:)) = newChromosome;
    end
end
display(["chromosome setelah crossover" z]);
display(chromosome);
% 6. Mutasi
% mutasi rate = 10% (mr)
mr = 0.1;
display(["mutasi Rate" z]);
display(mr);

jumlah_mutasi = round(mr * ind * indC);
display(["Jumlah yang dilakukan mutasi" z]);
display(jumlah_mutasi);

% bangkitkan lokasi yang di mutasi
ind_mutasi_colom = randi([1 indC],jumlah_mutasi,1);
ind_mutasi_baris = randi([1 ind],jumlah_mutasi,1);

display(["indeks yang dilakukan mutasi" z]);
display([ind_mutasi_colom ind_mutasi_baris]);

chromosome(ind_mutasi_colom,ind_mutasi_baris) = randi([0 100]);
display(["chromosome setelah mutasi" z]);

```



```
display(chromosome);  
display([' saham 1 ' ' saham 2 ' ' persentase saham 1+2 ' ' Laba ']);  
display([chromosome chromosome(:,1)+chromosome(:,2) A*chromosome(:,1)+B*chromosome(:,2)-50]);  
end
```