

LAPORAN PEMROGRAMAN KONTROLER

Dosen : Ahmad Radhy, S.Si., M.Si.

***” AgroBot : Autonomous Hybrid Vehicle Robot Sprayer Otomatis untuk Pengendalian
Agriculture Berbasis Watchdog System Guna Memitigasi Kegagalan Panen”***



Disusun Oleh :

- | | |
|--|---------------------|
| 1. Andik Putra Nazwana | (2042231010) |
| 2. Akhmad Maulvin Nazir Zakaria | (2042231028) |
| 3. Rany Surya Oktavia | (2042231060) |

**PRODI D4 TEKNOLOGI REKAYASA INSTRUMENTASI
DEPARTEMEN TEKNIK INSTRUMENTASI
FAKULTAS VOKASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
2025**

DAFTAR ISI

BAB 1. PENDAHULUAN	4
1.1 Latar Belakang	4
1.2 Rumusan Masalah	5
1.3 Tujuan	5
1.4 Manfaat	6
BAB 2. TINJAUAN PUSTAKA.....	7
2.1 STANDAR MISRA C:2012	7
2.2 <i>State Of The Art</i>	28
BAB 3. METODOLOGI.....	39
3.1 Desain V-Model.....	39
3.2 Arsitektur Model Desain	40
3.3 <i>Flowchart</i>	41
BAB 4. HASIL DAN PEMBAHASAN.....	42
4.1 Struktur Program	42
4.2 <i>Functional Safety</i> Dan <i>Unit Testing</i>	47
BAB 5. PENUTUP	60
5.1 Kesimpulan	60
5.2 Saran	60
DAFTAR PUSTAKA.....	61
LAMPIRAN.....	65

DAFTAR GAMBAR

Gambar 1. Model Software System AgroBot.....	39
Gambar 2. Architecture Design.....	40
Gambar 3. Flowchart AgroBot.....	41
Gambar 4. Structure Program AgroBot	42
Gambar 5. Program Pump	43
Gambar 6. Program Servo	43
Gambar 7. Program Soil Moisture.....	44

Gambar 8. Program DHT 11	44
Gambar 9. Program Gripper	45
Gambar 10. Program Integration	45
Gambar 11. Program Mecanishm	46
Gambar 12. Program Motor.....	46

DAFTAR TABEL

Tabel 1. <i>State of The Art</i>	28
Tabel 2. <i>Functional Safety</i>	47
Tabel 3. Hasil Pengujian dan <i>Debugging</i>	57

BAB 1. PENDAHULUAN

1.1 Latar Belakang

Ketahanan pangan dan ekonomi Nasional merupakan dua aspek yang saling terkait dan sangat krusial bagi stabilitas dan kesejahteraan suatu negara. Pada sektor pertanian, sebagai tulang punggung dalam penyediaan bahan pangan, memainkan peran yang vital dalam memastikan ketahanan pangan serta mendorong pertumbuhan ekonomi. Namun, dalam beberapa dekade terakhir sektor pertanian menghadapi berbagai tantangan yang mengancam keberlanjutan dan efektivitasnya bagi suatu negara. Berdasarkan laporan *Food and Agriculture Organization of the United Nations* 2022, permasalahan utama saat ini yang dihadapi sektor pertanian adalah perubahan iklim, degradasi lahan, urbanisasi yang pesat, serta kebijakan yang tidak konsisten sehingga hal ini dapat memperburuk kondisi pangan saat ini dan mengakibatkan hasil produksi pangan menurun.

Di Banyak negara, termasuk di Indonesia sektor pertanian memegang peran yang penting dalam perekonomian. Menurut data terbaru dari Badan Pusat Statistik 2023, sektor pertanian masih menjadi penyumbang utama yaitu sekitar 13,5% dari Produk Domestik Bruto (PDB) dan menyerap lebih dari 30% tenaga kerja nasional dalam jumlah banyak. Namun, saat ini kontribusi sektor pertanian terhadap ekonomi nasional cenderung mengalami penurunan karena berbagai faktor salah satunya perubahan iklim yang tak menentu, sehingga berdampak pada tingkat produktivitas pertanian. Seperti, terjadi fluktuasi cuaca yang ekstrim, kekeringan dan banjir, menyebabkan ketidakpastian hasil panen (*Food and Agriculture Organization of the United Nations*, 2022). Disisi lain, teknologi pertanian yang modern dan inovatif masih belum merata dalam penggunaannya, yang mengakibatkan terjadinya kesenjangan produktivitas antara petani yang memiliki akses terhadap teknologi dengan yang tidak memiliki akses.

Hal tersebut sangat kontradiktif dengan era revolusi industri, yang dimana seharusnya sektor pertanian juga harus ikut adaptif dalam era revolusi industri. Era Revolusi Industri 4.0 merupakan revolusi keempat yang melanda dunia. Segala perubahan yang terjadi, khususnya pada masa revolusi industri 4.0, akan memberikan manfaat bagi kehidupan dunia karena kemudahan akses terhadap penemuan-penemuan yang terus diperbarui. Era revolusi industri 4.0 dapat mengubah pekerjaan masyarakat menjadi lebih baik karena masyarakat dapat memperluas gerakannya. Salah satu aspek dari revolusi industri 4.0 adalah berkembangnya *Internet of Things* (IoT). Kemajuan ini dicapai melalui penemuan teknologi baru di bidang sains, robotika, dan teknik. Teknologi yang dimaksud dalam konteks ini adalah nanoteknologi, 3D dan kecerdasan buatan (Febrianti et al., 2021).

Perkembangan teknologi robotika dalam bidang pertanian telah membuka peluang besar untuk otomatisasi proses seperti penyemprotan pestisida dan pemupukan secara presisi menggunakan robot *autonomous*. Salah satu komponen penting dalam sistem ini adalah *watchdog system*, yaitu sistem pemantauan berbasis mikrokontroler yang berfungsi untuk mendeteksi dan mengatasi kegagalan sistem secara *real-time*. *Watchdog* akan secara otomatis mereset atau memperbaiki sistem apabila terjadi kerusakan fungsi seperti *crash* atau *hang*, sehingga robot dapat

terus bekerja tanpa intervensi manual yang signifikan. Integrasi teknologi ini dalam robot *sprayer* pertanian tidak hanya meningkatkan keandalan operasional, tetapi juga mengurangi risiko kegagalan fungsi di lapangan yang berpotensi menyebabkan kerusakan tanaman atau pemborosan sumber daya. Dengan dukungan kecerdasan buatan dan sistem navigasi *modern*, serta penggunaan kendaraan *hybrid* yang efisien dan ramah lingkungan, solusi ini sangat potensial untuk diterapkan dalam sistem pertanian presisi masa kini.

Sehingga dalam menghadapi tantangan pertanian *modern*, dibutuhkan sebuah inovasi yang tidak hanya mampu meningkatkan produktivitas, tetapi juga dapat beroperasi secara mandiri dan adaptif terhadap kondisi lapangan yaitu **AgroBot : Autonomous Hybrid Vehicle Robot Sprayer Otomatis untuk Pengendalian Agriculture Berbasis Watchdog System Guna Memitigasi Kegagalan Panen**. AgroBot hadir sebagai solusi inovatif melalui penggabungan teknologi kendaraan *hybrid* yang hemat energi, sistem penyemprotan otomatis berbasis sensor, serta integrasi sistem *watchdog* untuk memastikan operasi yang stabil dan bebas dari kegagalan fungsi. Dengan kemampuan *autonomous*, AgroBot dirancang untuk mendeteksi area yang memerlukan penyemprotan pestisida atau nutrisi secara presisi. Sistem *watchdog* yang tertanam di dalamnya akan terus memantau performa robot secara *real-time*, memungkinkan deteksi dini terhadap kerusakan atau *error* yang bisa mengganggu jalannya operasi. Dengan pendekatan teknologi ini, AgroBot diharapkan mampu memitigasi risiko kegagalan panen akibat serangan hama dan penyakit, sekaligus meningkatkan efisiensi kerja di sektor pertanian secara signifikan. Inovasi ini menjawab kebutuhan petani akan solusi cerdas dan andal di era digitalisasi pertanian, sekaligus memperkuat upaya mencapai ketahanan pangan nasional di masa depan.

1.2 Rumusan Masalah

Dari latar belakang yang ada, rumusan masalah yang dapat diberikan adalah :

1. Bagaimana perancangan dan mekanisme kerja AgroBot sebagai robot *sprayer* otomatis berbasis sistem *watchdog*?
2. Bagaimana sistem *watchdog* mampu mendeteksi dan mengatasi kegagalan fungsi pada robot secara *real-time*?
3. Bagaimana integrasi teknologi sensor dan kecerdasan buatan dalam AgroBot dapat meminimalkan risiko kegagalan panen?

1.3 Tujuan

Tujuan yang akan dicapai dari Tugas Proyek ini adalah sebagai berikut :

1. Merancang dan mengembangkan AgroBot sebagai robot *sprayer* otomatis yang dilengkapi dengan sistem *watchdog*.
2. Menguji kemampuan sistem *watchdog* dalam memantau performa robot secara *real-time* dan melakukan tindakan korektif saat terjadi gangguan.
3. Menganalisis kemampuan AgroBot dalam mendeteksi area target penyemprotan secara presisi guna meningkatkan hasil pertanian dan memitigasi risiko kegagalan panen.

1.4 Manfaat

Manfaat yang didapatkan dari inovasi ini adalah sebagai berikut :

1. Bagi Mahasiswa

Menjadi sarana penerapan ilmu di bidang teknologi pertanian, dan kecerdasan buatan dalam proyek nyata. Serta memberikan pengalaman langsung dalam pengembangan perangkat berbasis sistem pengendalian otomatis.

2. Bagi Petani

Dengan adanya inovasi teknologi dalam sektor pertanian bisa mempermudah proses penyemprotan pestisida dan pupuk secara otomatis dan lebih presisi, mengurangi beban kerja manual, terutama bagi petani kecil dan menengah, mengurangi risiko gagal panen akibat keterlambatan atau kesalahan dalam proses penyemprotan, serta meningkatkan efisiensi penggunaan sumber daya seperti air, pupuk, dan pestisida, sehingga menekan biaya operasional.

3. Bagi Pemerintah

Mendukung program ketahanan pangan nasional melalui pemanfaatan teknologi tepat guna. Serta menjadi acuan dalam pengembangan kebijakan berbasis inovasi teknologi untuk sektor pertanian berkelanjutan.

4. Bagi Industri

Mendorong kolaborasi antara dunia industri dengan institusi pendidikan tinggi dalam pengembangan alat dan sistem pertanian presisi. Serta memberi insentif untuk investasi di sektor agritech (*agriculture technology*) sebagai bagian dari diversifikasi industri nasional.

BAB 2. TINJAUAN PUSTAKA

2.1 STANDAR MISRA C:2012

Aturan 11.3 : Pemindahan tidak boleh dilakukan antara penunjuk ke tipe objek dan penunjuk ke tipe objek yang berbeda.

Kategori : Diperlukan

Analisis : Dapat Diputuskan, Unit Terjemahan Tunggal

Berlaku untuk : C90, C99

Amplifikasi : Aturan ini berlaku untuk tipe yang tidak memenuhi syarat yang ditunjuk oleh pointer.

Alasan : Mentransmisikan pointer (variabel yang menyimpan alamat memori atau variabel lain) ke objek lain dapat mengakibatkan pointer yang tidak selaras dengan benar, sehingga menghasilkan perilaku yang tidak terdefinisi. Bahkan jika konversi diketahui menghasilkan pointer yang selaras dengan benar, perilaku tersebut mungkin tidak terdefinisi jika pointer tersebut digunakan untuk mengakses objek. Misalnya, jika objek yang bertipe *int* diakses sebagai *short*, perilaku tersebut tidak terdefinisi meskipun *int* dan *short* memiliki representasi dan persyaratan penyelarasan yang sama. Lihat C90 Bagian 6.3, C99 Bagian 6.5, paragraf 7 untuk detailnya.

Pengecualian : Diperbolehkan untuk mengubah pointer ke tipe objek menjadi pointer ke salah satu tipe objek *char*, *signed char* atau *unsigned char*. Standar menjamin bahwa pointer ke tipe ini dapat digunakan untuk mengakses byte individual dari suatu objek.

Contoh :

```

uint8_t *p1;
uint32_t *p2;

/* Non-compliant - possible incompatible alignment */
p2 = ( uint32_t * ) p1;

extern uint32_t read_value ( void );
extern void print ( uint32_t n );

void f ( void )
{
    uint32_t u    = read_value ( );
    uint16_t *hi_p = ( uint16_t * ) &u;    /* Non-compliant even though
                                           * probably correctly aligned */

    *hi_p = 0;    /* Attempt to clear high 16-bits on big-endian machine */
    print ( u );  /* Line above may appear not to have been performed */
}

const short *p;
const volatile short *q;

q = ( const volatile short * ) p;    /* Compliant */

The following example is non-compliant because the unqualified pointer types are different, namely
"pointer to const-qualified int" and "pointer to int".

int * const * pcpci;
const int * const * pcpci;

pcpci = ( const int * const * ) pcpci;

```

- Aturan 11.4** : Konversi tidak boleh dilakukan antara pointer ke objek dan tipe *integer*
- Kategori** : Saran
- Analisis** : Dapat Diputuskan, Unit Terjemahan Tunggal
- Berlaku untuk** : C90, C99
- Amplifikasi** : Pointer tidak boleh diubah menjadi integer. Integer tidak boleh diubah menjadi pointer.
- Alasan** : Konversi *integer* menjadi pointer ke objek dapat menghasilkan pointer yang tidak selaras dengan benar, yang mengakibatkan perilaku tidak terdefinisi. Konversi pointer ke objek menjadi integer dapat menghasilkan nilai yang tidak dapat direpresentasikan dalam tipe integer yang dipilih sehingga mengakibatkan perilaku tidak terdefinisi.
- Catatan** : tipe C99 *intptr_t* dan *uintptr_t*, yang dideklarasikan dalam *<stdint.h>*, masing-masing adalah tipe *integer* bertanda dan tidak bertanda yang mampu merepresentasikan nilai pointer. Meskipun demikian, konversi antara pointer ke objek dan tipe ini tidak diizinkan oleh

aturan ini karena penggunaannya tidak menghindari perilaku tidak terdefinisi yang terkait dengan pointer yang tidak selaras. Pengecoran antara pointer dan tipe *integer* harus dihindari jika memungkinkan, tetapi mungkin diperlukan saat menangani register yang dipetakan memori atau fitur khusus perangkat keras lainnya. Jika pengecoran antara integer dan pointer digunakan, perhatian harus diberikan untuk memastikan bahwa setiap pointer yang dihasilkan tidak menimbulkan perilaku tidak terdefinisi yang dibahas dalam Aturan 11.3.

Pengecualian : Konstanta penunjuk null yang bertipe integer dapat diubah menjadi penunjuk ke objek.

Contoh :

```
uint8_t *PORTA = ( uint8_t * ) 0x0002;    /* Non-compliant */

uint16_t *p;

int32_t  addr = ( int32_t ) &p;            /* Non-compliant */
uint8_t  *q    = ( uint8_t * ) addr;       /* Non-compliant */
bool_t   b     = ( bool_t ) p;             /* Non-compliant */

enum etag { A, B } e = ( enum etag ) p;    /* Non-compliant */
```

Aturan 11.5 : Konversi tidak boleh dilakukan dari pointer ke void menjadi pointer ke objek

Kategori : Saran

Analisis : Dapat diputuskan, Unit Terjemahan Tunggal

Berlaku untuk : C90, C99

Alasan : Konversi pointer ke void menjadi pointer ke objek dapat mengakibatkan pointer yang tidak selaras dengan benar, sehingga menghasilkan perilaku yang tidak terdefinisi. Hal ini harus dihindari jika memungkinkan tetapi mungkin diperlukan, misalnya saat menangani fungsi alokasi memori. Jika konversi dari pointer ke objek menjadi pointer ke void yang digunakan, harus diperhatikan untuk memastikan bahwa pointer yang dihasilkan tidak menimbulkan perilaku tidak terdefinisi yang dibahas dalam Aturan 11.3.

Pengecualian : Konstanta penunjuk null yang bertipe pointer ke void dapat diubah menjadi pointer ke objek.

Contoh :

```
uint32_t *p32;
void      *p;
uint16_t *p16;

p  = p32;                /* Compliant - pointer to uint32_t into
                          * pointer to void */
p16 = p;                 /* Non-compliant */
p  = ( void * ) p16;      /* Compliant */
p32 = ( uint32_t * ) p;   /* Non-compliant */
```

Aturan 11.6 : Sebuah cast tidak boleh dilakukan antara pointer ke void dan tipe aritmatika

Kategori : Diperlukan

Analisis : Dapat Diputuskan, Unit Terjemahan Tunggal

Berlaku untuk : C90, C99

Alasan : Konversi *integer* ke pointer ke void dapat menghasilkan pointer yang tidak selaras dengan benar, yang mengakibatkan perilaku tidak terdefinisi. Konversi pointer ke void ke *integer* dapat menghasilkan nilai yang tidak dapat direpresentasikan dalam tipe *integer* yang dipilih sehingga mengakibatkan perilaku tidak terdefinisi. Konversi antara tipe aritmatika non-integer dan pointer ke void tidak terdefinisi.

Pengecualian : Ekspresi konstan *integer* dengan nilai 0 dapat diubah menjadi pointer ke void.

Contoh :

```
void    *p;
uint32_t u;

/* Non-compliant - implementation-defined */
p = ( void * ) 0x1234u;

/* Non-compliant - undefined */
p = ( void * ) 1024.0f;

/* Non-compliant - implementation-defined */
u = ( uint32_t ) p;
```

Aturan 11.7 : Pemrosesan tidak boleh dilakukan antara pointer ke objek dan tipe aritmatika non-integer

Kategori : Diperlukan

Analisis : Dapat Diputuskan, Unit Terjemahan Tunggal

Berlaku untuk : C90, C99

Amplifikasi : Untuk tujuan aturan ini, tipe aritmatika non-integer berarti salah satu dari

1. Pada dasarnya Boolean
2. Pada dasarnya karakter
3. Pada dasarnya enum

4. Pada dasarnya mengambang

Alasan : Konversi tipe Boolean, karakter, atau enum menjadi penunjuk ke objek dapat menghasilkan penunjuk yang tidak selaras dengan benar, sehingga menghasilkan perilaku yang tidak terdefinisi. Konversi penunjuk ke objek menjadi tipe Boolean, karakter, atau enum dapat menghasilkan nilai yang tidak dapat direpresentasikan dalam tipe integer yang dipilih sehingga menghasilkan perilaku yang tidak terdefinisi. Konversi penunjuk ke objek menjadi atau dari tipe mengambang menghasilkan perilaku yang tidak terdefinisi.

Contoh :

```
int16_t *p;
float32_t f;

f = ( float32_t ) p;    /* Non-compliant */
p = ( int16_t * ) f;    /* Non-compliant */
```

Lihat juga Aturan 11.4

Aturan 11.8 : Sebuah cast tidak boleh menghilangkan kualifikasi const atau volatile dari tipe yang ditunjuk oleh pointer

Kategori : Diperlukan

Analisis : Dapat Diputuskan, Unit Terjemahan Tunggal

Berlaku untuk : C90, C99

Alasan : Setiap upaya untuk menghapus kualifikasi yang terkait dengan tipe yang dituju dengan menggunakan casting merupakan pelanggaran terhadap prinsip kualifikasi tipe.

Catatan: kualifikasi yang dirujuk di sini tidak sama dengan kualifikasi apa pun yang dapat diterapkan pada penunjuk itu sendiri. Beberapa masalah yang mungkin timbul jika kualifikasi dihapus dari objek yang dituju adalah:

1. Menghapus kualifikasi const dapat menghindari status baca-saja suatu objek dan mengakibatkan dimodifikasi
2. Menghapus kualifikasi const dapat mengakibatkan pengecualian saat objek diakses
3. Menghapus kualifikasi volatil dapat mengakibatkan akses ke objek dioptimalkan.

Catatan : penghapusan kualifikasi tipe pembatasan C99 tidak berbahaya

Contoh :

```

uint16_t      x;
uint16_t * const cpi = &x;    /* const pointer          */
uint16_t * const *pcpi;       /* pointer to const pointer */
uint16_t *      *ppi;
const uint16_t  *pci;         /* pointer to const          */
volatile uint16_t *pvi;       /* pointer to volatile       */
uint16_t        *pi;

pi = cpi;                    /* Compliant - no conversion
                               no cast required */
pi = (uint16_t *)pci;        /* Non-compliant            */
pi = (uint16_t *)pvi;        /* Non-compliant            */
ppi = (uint16_t * *)pcpi;    /* Non-compliant            */

```

Lihat juga aturan 11.3

Aturan 11.9 : Makro NULL adalah satu-satunya bentuk yang diizinkan dari konstanta pointer null integer

Kategori : Diperlukan

Analisis : Dapat Diputuskan, Unit Terjemahan Tunggal

Berlaku untuk : C90, C99

Amplifikasi : Ekspresi konstanta integer dengan nilai 0 akan diturunkan dari perluasan makro NULL jika muncul dalam salah satu konteks berikut:

1. Sebagai nilai yang ditetapkan ke pointer
2. Sebagai operan dari operator == atau != yang operan lainnya adalah pointer
3. Sebagai operan kedua dari operator ?: yang operan ketiganya adalah pointer
4. Sebagai operan ketiga dari operator ?: yang operan keduanya adalah pointer

Dengan mengabaikan spasi dan tanda kurung di sekitarnya, ekspresi konstanta integer tersebut akan mewakili seluruh perluasan NULL. Catatan: konstanta penunjuk null dalam bentuk (void *)0 diizinkan, baik diperluas dari NULL atau tidak.

Alasan : Penggunaan NULL daripada 0 memperjelas bahwa konstanta penunjuk null memang dimaksudkan.

Contoh : Dalam contoh berikut, inisialisasi p2 patuh karena ekspresi konstanta integer 0 tidak muncul dalam salah satu konteks yang dilarang oleh aturan ini.

```
int32_t *p1 = 0;                /* Non-compliant */

int32_t *p2 = ( void * ) 0;    /* Compliant    */
```

Dalam contoh berikut, perbandingan antara p2 dan (void *)0 patuh karena ekspresi konstanta integer 0 muncul sebagai operan dari suatu cast dan tidak dalam salah satu konteks yang dilarang oleh aturan ini.

```
#define MY_NULL_1 0
#define MY_NULL_2 ( void * ) 0

if ( p1 == MY_NULL_1 )    /* Non-compliant */
{
}
if ( p2 == MY_NULL_2 )    /* Compliant    */
{
}
```

Contoh berikut ini patuh karena penggunaan makro NULL yang disediakan oleh implementasi selalu diizinkan, bahkan jika diperluas ke ekspresi konstanta integer dengan nilai 0.

```
/* Could also be stdio.h, stdlib.h and others */
#include <stddef.h>

extern void f ( uint8_t *p );

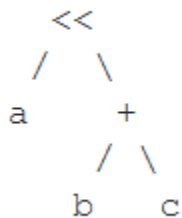
/* Compliant for any conforming definition of NULL, such as:
 * 0
 * ( void * ) 0
 * ( ( ( 0 ) ) )
 * ( ( ( 1 - 1 ) ) )
 */
f ( NULL );
```

Lihat juga aturan 11.4

Aturan 12.1	: Prioritas operator dalam ekspresi harus dibuat eksplisit
Kategori	: Saran
Analisis	: Dapat Diputuskan, Unit Terjemahan Tunggal
Berlaku untuk	: C90, C99
Amplifikasi	: Tabel berikut digunakan dalam definisi aturan ini.

Description	Operator or Operand	Precedence
Primary	identifier, constant, string literal, (expression)	16 (high)
Postfix	[] () (function call) . -> ++ (post-increment) -- (post-decrement) () { } (C99: compound literal)	15
Unary	++ (pre-increment) -- (pre-decrement) & * + - ~ ! <i>sizeof defined</i> (preprocessor)	14
Cast	()	13
Multiplicative	* / %	12
Additive	+ -	11
Bitwise shift	<< >>	10
Relational	< > <= >=	9
Equality	== !=	8
Bitwise AND	&	7
Bitwise XOR	^	6
Bitwise OR		5
Logical AND	&&	4
Logical OR		3
Conditional	? :	2
Assignment	= *= /= %= += -= <<= >>= &= ^= =	1
Comma	,	0 (low)

Preceden yang digunakan dalam tabel ini dipilih untuk memungkinkan deskripsi aturan yang ringkas. Preceden tersebut tidak harus sama dengan preceden yang mungkin ditemukan dalam deskripsi lain tentang preceden operator. Untuk tujuan aturan ini, preceden suatu ekspresi adalah preceden elemen (operand atau operator) pada akar pohon parse untuk ekspresi tersebut. Misalnya: pohon parse untuk ekspresi `a << b + c` dapat direpresentasikan sebagai:



Elemen pada akar pohon parse ini adalah '<<-' sehingga ekspresi memiliki prioritas 10.

Saran berikut diberikan:

1. Operand dari operator `sizeof` harus diapit tanda kurung

2. Ekspresi yang prioritasnya berada dalam rentang 2 hingga 12 harus memiliki tanda kurung di sekitar operan yang memiliki keduanya:
 - a. Prioritas kurang dari 13, dan
 - b. Prioritas lebih besar dari prioritas ekspresi.

Alasan : Bahasa C memiliki jumlah operator yang relatif besar dan prioritas relatifnya tidak intuitif. Hal ini dapat menyebabkan programmer yang kurang berpengalaman melakukan kesalahan. Menggunakan tanda kurung untuk membuat prioritas operator eksplisit menghilangkan kemungkinan bahwa ekspektasi programmer tidak benar. Hal ini juga membuat maksud programmer asli jelas bagi pengulas atau pengelola kode. Diakui bahwa penggunaan tanda kurung yang berlebihan dapat mengacaukan kode dan mengurangi keterbacaannya. Aturan ini bertujuan untuk mencapai kompromi antara kode yang sulit dipahami karena mengandung terlalu banyak atau terlalu sedikit tanda kurung.

Contoh : Contoh berikut menunjukkan ekspresi dengan operator unary atau postfix yang operannya berupa ekspresi primer atau ekspresi yang operator tingkat atasnya memiliki prioritas 15.

```
a[ i ]->n;      /* Compliant - no need to write ( a[ i ] )->n      */
*p++;          /* Compliant - no need to write *( p++ )          */
sizeof x + y;   /* Non-compliant - write either sizeof ( x ) + y
                * or sizeof ( x + y )                             */
```

Contoh berikut menunjukkan ekspresi yang berisi operator pada tingkat prioritas yang sama. Semua ini patuh tetapi, tergantung pada jenis a, b, dan c, ekspresi apa pun dengan lebih dari satu operator dapat melanggar aturan lain.

```
a + b;
a + b + c;
( a + b ) + c;
a + ( b + c );
a + b - c + d;
( a + b ) - ( c + d );
```

Contoh berikut menunjukkan berbagai ekspresi operator campuran:


```

/* Compliant - no need to write f ( ( a + b ), c ) */
x = f ( a + b, c );

/* Non-compliant
* Operands of conditional operator (precedence 2) are:
*   == precedence 8 needs parentheses
*   a precedence 16 does not need parentheses
*   - precedence 11 needs parentheses
*/
x = a == b ? a : a - b;

/* Compliant */
x = ( a == b ) ? a : ( a - b );

/* Compliant
* Operands of << operator (precedence 10) are:
*   a precedence 16 does not need parentheses
*   ( E ) precedence 16 already parenthesized
*/
x = a << ( b + c );

/* Compliant
* Operands of && operator (precedence 4) are:
*   a precedence 16 does not need parentheses
*   && precedence 4 does not need parentheses
*/
if ( a && b && c )
{
}

/* Compliant
* Operands of && operator (precedence 4) are:
*   defined(X) precedence 14 does not need parentheses
*   (E) precedence 16 already parenthesized
*/
#if defined ( X ) && ( ( X + Y ) > Z )

/* Compliant
* Operands of && operator (precedence 4) are:
*   !defined ( X ) precedence 14 does not need parentheses
*   defined ( Y ) precedence 14 does not need parentheses
*   Operand of ! operator (precedence 14) is:
*   defined ( X ) precedence 14 does not need parentheses
*/
#if !defined ( X ) && defined ( Y )

```

Catatan : aturan ini tidak mengharuskan operan operator , diberi tanda kurung.

Penggunaan operator , dilarang oleh Aturan 12.3.

x = a, b; /* Compliant - parsed as (x = a), b */

Aturan 12.2 : Operan tangan kanan dari operator pergeseran harus berada dalam rentang nol hingga satu kurang dari lebar bit tipe esensial dari operan tangan kiri.

Kategori : Diperlukan

Analisis : Tidak dapat diputuskan, Sistem

Berlaku untuk : C90, C99

Alasan : Jika operan sisi kanan negatif, atau lebih besar dari atau sama dengan lebar operan sisi kiri, maka perilakunya tidak terdefinisi. Jika, misalnya, operan sisi kiri dari pergeseran kiri atau pergeseran kanan adalah bilangan bulat 16-bit, maka penting untuk memastikan bahwa ini hanya digeser oleh angka dalam rentang 0 hingga 15. Lihat Bagian 8.10 untuk deskripsi tipe esensial dan batasan pada tipe esensial untuk operan operator pergeseran. Ada berbagai cara untuk memastikan aturan ini diikuti. Yang paling sederhana adalah operan sisi kanan menjadi konstanta (yang nilainya kemudian dapat diperiksa secara statis). Penggunaan tipe bilangan bulat tak bertanda akan memastikan bahwa operannya tidak negatif, jadi hanya batas atas yang perlu diperiksa (secara dinamis pada waktu proses atau dengan peninjauan). Jika tidak, kedua batas tersebut perlu diperiksa.

Contoh :

```
u8a = u8a << 7;          /* Compliant */
u8a = u8a << 8;          /* Non-compliant */
u16a = ( uint16_t ) u8a << 9; /* Compliant */
```

Untuk membantu dalam memahami contoh-contoh berikut, perlu dicatat bahwa tipe esensial 1u pada dasarnya adalah unsigned char, sedangkan tipe esensial 1UL pada dasarnya adalah unsigned long.

```
1u << 10u;                /* Non-compliant */
( uint16_t ) 1u << 10u;    /* Compliant */
1UL << 10u;               /* Compliant */
```

Aturan 12.3 : Operator koma tidak boleh digunakan

Kategori : Saran

Analisis : Dapat Diputuskan, Unit Terjemahan Tunggal

Berlaku untuk : C90, C99

Alasan : Penggunaan operator koma umumnya merugikan keterbacaan kode, dan efek yang sama biasanya dapat dicapai dengan cara lain.

Contoh :

```
f ( ( 1, 2 ), 3 ); /* Non-compliant - how many parameters? */
```

Contoh berikut tidak mematuhi aturan ini dan aturan lainnya:

```
for ( i = 0, p = &a[ 0 ]; i < N; ++i, ++p )  
{  
}
```

Aturan 12.4 : Evaluasi ekspresi konstan tidak boleh mengarah ke unsigned integer wrap-around

Kategori : Saran

Analisis : Dapat Diputuskan, Unit Terjemahan Tunggal

Berlaku untuk : C90, C99

Amplifikasi : Aturan ini berlaku untuk ekspresi yang memenuhi batasan untuk ekspresi konstan, baik muncul dalam konteks yang memerlukan ekspresi konstan atau tidak.

Jika suatu ekspresi tidak dievaluasi, misalnya karena muncul di operan kanan operator logika AND yang operan kirinya selalu salah, maka aturan ini tidak berlaku.

Alasan : Ekspresi integer tak bertanda tidak sepenuhnya melimpah, tetapi sebaliknya berputar-putar. Meskipun mungkin ada alasan yang baik untuk menggunakan aritmatika modulo pada waktu proses, kecil kemungkinan penggunaannya disengaja pada waktu kompilasi.

Contoh : Ekspresi yang dikaitkan dengan label case harus berupa ekspresi konstan. Jika unsigned wraparound terjadi selama evaluasi ekspresi case, kemungkinan besar hal itu tidak disengaja. Pada mesin dengan tipe int 16-bit, nilai BASE apa pun yang lebih besar dari atau sama dengan 65 024 akan mengakibatkan wraparound dalam contoh berikut:

```

#define BASE 65024u

switch ( x )
{
    case BASE + 0u:
        f ( );
        break;
    case BASE + 1u:
        g ( );
        break;
    case BASE + 512u: /* Non-compliant - wraps to 0 */
        h ( );
        break;
}

```

Ekspresi pengontrol dari arahan praprosesor `#if` atau `#elif` harus berupa ekspresi konstan.

```

#if 1u + ( 0u - 10u ) /* Non-compliant as ( 0u - 10u ) wraps */

```

Dalam contoh ini, ekspresi `DELAY + WIDTH` memiliki nilai 70.000 tetapi ini akan menghasilkan 4.464 pada mesin dengan tipe int 16-bit.

```

#define DELAY 10000u
#define WIDTH 60000u

void fixed_pulse ( void )
{
    uint16_t off_time16 = DELAY + WIDTH; /* Non-compliant */
}

```

Aturan ini tidak berlaku untuk ekspresi `c + 1` dalam contoh yang sesuai berikut karena ekspresi tersebut mengakses suatu objek dan karenanya tidak memenuhi batasan untuk ekspresi konstan:

```

const uint16_t c = 0xffffu;

void f ( void )
{
    uint16_t y = c + 1u; /* Compliant */
}

```

Dalam contoh berikut, sub-ekspresi `(0u - 1u)` mengarah ke unsigned integer wrap-around. Dalam inisialisasi `x`, sub-ekspresi tidak dievaluasi dan ekspresi tersebut karenanya patuh. Namun, dalam inisialisasi `y`, ekspresi tersebut dapat dievaluasi dan ekspresi tersebut karenanya tidak patuh.

```

bool_t b;

void g ( void )
{
    uint16_t x = ( 0u == 0u ) ? 0u : ( 0u - 1u );    /* Compliant */
    uint16_t y = b ? 0u : ( 0u - 1u );                /* Non-compliant */
}

```

Aturan 13.1 : Daftar inisialisasi tidak boleh berisi efek samping yang terus-menerus

Kategori : Diperlukan

Analisis : Tidak dapat diputuskan, Sistem

Berlaku untuk : C99

Alasan : C90 membatasi inisialisasi untuk objek otomatis dengan tipe agregat agar hanya berisi ekspresi konstan. Namun, C99 mengizinkan inisialisasi agregat otomatis untuk berisi ekspresi yang dievaluasi saat run-time. C99 juga mengizinkan literal gabungan yang berperilaku sebagai objek inisialisasi anonim. Urutan terjadinya efek samping selama evaluasi ekspresi dalam daftar inisialisasi tidak ditentukan dan perilaku inisialisasi tidak dapat diprediksi jika efek samping tersebut terus menerus.

Contoh :

```

volatile uint16_t v1;

void f ( void )
{
    /* Non-compliant - volatile access is persistent side effect */
    uint16_t a[ 2 ] = { v1, 0 };
}

void g ( uint16_t x, uint16_t y )
{
    /* Compliant - no side effects */
    uint16_t a[ 2 ] = { x + y, x - y };
}

uint16_t x = 0u;

extern void p ( uint16_t a[ 2 ] );

void h ( void )
{
    /* Non-compliant - two side effects */
    p ( ( uint16_t[ 2 ] ) { x++, x++ } );
}

```

Lihat juga aturan 13.2

Aturan 13.2 : Nilai suatu ekspresi dan efek sampingnya yang terus-menerus harus sama di bawah semua perintah evaluasi yang diizinkan

Kategori : Diperlukan

Analisis : Tidak dapat diputuskan, Sistem

Berlaku untuk : C90, C99

Amplifikasi : Antara dua titik sekuens yang berdekatan atau dalam ekspresi penuh apa pun:

1. Tidak ada objek yang boleh dimodifikasi lebih dari satu kali;
2. Tidak ada objek yang boleh dimodifikasi dan dibaca kecuali pembacaan nilai objek tersebut
3. berkontribusi terhadap penghitungan nilai yang akan disimpan ke dalam objek;
4. Tidak boleh ada lebih dari satu akses modifikasi dengan tipe berkualifikasi volatil;
5. Tidak boleh ada lebih dari satu akses baca dengan tipe berkualifikasi volatil.

Catatan : Suatu objek dapat diakses secara tidak langsung, melalui penunjuk atau fungsi yang dipanggil, serta diakses secara langsung oleh ekspresi.

Catatan : Amplifikasi ini sengaja dibuat lebih ketat daripada judul aturan. Akibatnya, ekspresi seperti: $x = x = 0$;

Tidak diizinkan oleh aturan ini meskipun nilai dan efek samping yang persisten, asalkan x tidak mudah berubah, tidak bergantung pada urutan evaluasi atau efek samping. Poin urutan dirangkum dalam Lampiran C dari standar C90 dan C99. Poin urutan dalam C90 merupakan bagian dari yang ada di C99. Ekspresi lengkap didefinisikan dalam Bagian 6.6 dari standar C90 dan Bagian 6.8 dari standar C99.

Alasan : Standar memberikan fleksibilitas yang cukup besar kepada penyusun saat mengevaluasi ekspresi. Sebagian besar operator dapat mengevaluasi operannya dalam urutan apa pun. Pengecualian utamanya adalah:

1. AND && Logika di mana operan kedua dievaluasi hanya jika operan pertama dievaluasi menjadi bukan nol;
2. OR || Logika di mana operan kedua dievaluasi hanya jika operan pertama dievaluasi menjadi nol; Operator kondisional ?: di mana operan pertama selalu dievaluasi dan kemudian operan kedua atau ketiga dievaluasi;

3. Operator , di mana operan pertama dievaluasi dan kemudian operan kedua dievaluasi.

Catatan : Kehadiran tanda kurung dapat mengubah urutan penerapan operator. Namun, hal ini tidak memengaruhi urutan evaluasi operan tingkat terendah, yang dapat dievaluasi dalam urutan apa pun. Banyak kejadian umum perilaku tak terduga yang terkait dengan evaluasi ekspresi dapat dihindari dengan mengikuti saran yang diberikan oleh Aturan 13.3 dan Aturan 13.4.

Ketika makro *COPY_ELEMENT* dipanggil dalam contoh yang tidak sesuai ini, *i* dibaca dua kali dan dimodifikasi dua kali. Tidak ditentukan apakah urutan operasi pada *i* adalah:

1. Membaca, memodifikasi, membaca, memodifikasi, atau
2. Membaca, membaca, memodifikasi, memodifikasi

```
#define COPY_ELEMENT ( index ) ( a[( index )] = b[( index )] )

COPY_ELEMENT ( i++ );
```

Dalam contoh yang tidak sesuai ini, urutan pembacaan *v1* dan *v2* tidak ditentukan.

```
extern volatile uint16_t v1, v2;
uint16_t t;

t = v1 + v2;
```

Dalam contoh yang patuh ini *PORT* dibaca dan dimodifikasi.

```
extern volatile uint8_t PORT;

PORT = PORT & 0x80u;
```

Urutan evaluasi argumen fungsi tidak ditentukan, begitu pula urutan terjadinya efek samping, seperti yang ditunjukkan dalam contoh yang tidak sesuai ini.

```
uint16_t i = 0;

/*
 * Unspecified whether this call is equivalent to:
 *     f ( 0, 0 )
 * or   f ( 0, 1 )
 */
f ( i++, i );
```

Urutan relatif evaluasi penunjuk fungsi dan argumen fungsi tidak ditentukan. Dalam contoh yang tidak sesuai ini, jika panggilan ke *g* mengubah *p* maka tidak ditentukan apakah penunjuk fungsi *p->f* menggunakan nilai *p* sebelum panggilan *g* atau setelahnya.

```
p->f ( g ( &p ) );
```

Lihat juga aturan 13.1; 13.3 ; dan 13.4

Aturan 13.3 : Ekspresi lengkap yang berisi operator increment (++) atau decrement (--) tidak boleh memiliki efek samping potensial lain selain yang disebabkan oleh operator increment atau decrement

Kategori : Advisory

Analisis : Unit Terjemahan Tunggal yang Dapat Diputuskan

Berlaku untuk : C90, C99

Amplifikasi : Pemanggilan fungsi dianggap sebagai efek samping untuk tujuan aturan ini. Semua sub-ekspresi dari ekspresi lengkap diperlakukan seolah-olah dievaluasi untuk tujuan aturan ini, bahkan jika ditetapkan sebagai tidak dievaluasi oleh The Standard.

Dasar pemikiran : Penggunaan operator increment dan decrement dalam kombinasi dengan operator lain tidak direkomendasikan karena:

1. Dapat secara signifikan mengganggu keterbacaan kode
2. Menyebabkan efek samping tambahan ke dalam pernyataan dengan potensi perilaku yang tidak terdefinisi (dicakup oleh Aturan 13.2).

Lebih jelas untuk menggunakan operasi ini secara terpisah dari operator lainnya.

Contoh :

The expression:

```
u8a = u8b++
```

is non-compliant. The non-compliant expression statement:

```
u8a = ++u8b + u8c--;
```

is clearer when written as the following sequence:

```
++u8b;  
u8a = u8b + u8c;  
u8c--;
```

The following are all compliant because the only *side effect* in each expression is caused by the increment or decrement operator.

```
x++;  
a[ i ]++;  
b.x++;  
c->x++;  
++( *p );  
*p++;  
( *p )++;
```


Berikut ini semuanya tidak sesuai karena mengandung pemanggilan fungsi serta operator penambahan atau pengurangan :

```
if ( ( f ( ) + --u8a ) == 0u )
{
}

g ( u8b++ );
```

Berikut ini semuanya tidak patuh meskipun sub-ekspresi yang berisi operator kenaikan atau penurunan atau beberapa efek samping lainnya tidak dievaluasi:

```
u8a = ( 1u == 1u ) ? 0u : u8b++;

if ( u8a++ == ( ( 1u == 1u ) ? 0u : f ( ) ) )
{
}
```

Lihat juga aturan 13.2

Aturan 13.4 : Hasil dari operator penugasan tidak boleh digunakan

Kategori : Saran

Analisis : Dapat Diputuskan, Unit Terjemahan Tunggal

Berlaku untuk : C90, C99

Amplifikasi : Aturan ini berlaku bahkan jika ekspresi yang berisi operator penugasan tidak dievaluasi.

Alasan : Penggunaan operator penugasan, sederhana atau gabungan, dalam kombinasi dengan operator aritmatika lainnya tidak disarankan karena:

1. Dapat secara signifikan mengganggu keterbacaan kode;
2. Memberikan efek samping tambahan ke dalam pernyataan yang membuatnya lebih sulit untuk menghindari perilaku tidak terdefinisi yang tercakup dalam Aturan 13.2.

Contoh :

```
x = y;                                /* Compliant                                */
a[ x ] = a[ x = y ];                 /* Non-compliant - the value of x = y
                                     * is used                                */

/*
 * Non-compliant - value of bool_var = false is used but
 * bool_var == false was probably intended
 */
if ( bool_var = false )
{

/* Non-compliant even though bool_var = true isn't evaluated */
if ( ( 0u == 0u ) || ( bool_var = true ) )
{

/* Non-compliant - value of x = f() is used */
if ( ( x = f ( ) ) != 0 )
{

/* Non-compliant - value of b += c is used */
a[ b += c ] = a[ b ];

/* Non-compliant - values of c = 0 and b = c = 0 are used */
a = b = c = 0;
```

Lihat juga aturan 13.2

Aturan 13.5 : Operan sisi kanan dari operator logika && atau || tidak boleh mengandung efek samping yang terus-menerus

Kategori : Diperlukan

Analisis : Tidak dapat diputuskan, Sistem

Berlaku untuk : C90, C99

Dasar pemikiran : Evaluasi operan sisi kanan dari operator && dan || bergantung pada nilai operan sisi kiri. Jika operan sisi kanan mengandung efek samping, maka efek samping tersebut mungkin terjadi atau tidak, yang mungkin bertentangan dengan harapan programmer. Jika evaluasi operan sisi kanan akan menghasilkan efek samping yang tidak terus-menerus pada titik dalam program tempat ekspresi terjadi, maka tidak masalah apakah operan sisi kanan dievaluasi atau tidak. Istilah efek samping yang terus-menerus didefinisikan dalam Lampiran J.

Contoh :

```
uint16_t f ( uint16_t y )
{
    /* These side effects are not persistent as seen by the caller */
    uint16_t temp = y;

    temp = y + 0x8080U;

    return temp;
}

uint16_t h ( uint16_t y )
{
    static uint16_t temp = 0;

    /* This side effect is persistent */
    temp = y + temp;

    return temp;
}

void g ( void )
{
    /* Compliant - f ( ) has no persistent side effects */
    if ( ishigh && ( a == f ( x ) ) )
    {
    }

    /* Non-compliant - h ( ) has a persistent side effect */
    if ( ishigh && ( a == h ( x ) ) )
    {
    }
}

volatile uint16_t v;
uint16_t x;

/* Non-compliant - access to volatile v is persistent */
if ( ( x == 0u ) || ( v == 1u ) )
{
}

/* Non-compliant if fp points to a function with persistent side effects */
( fp != NULL ) && ( *fp ) ( 0 );
```

Aturan 13.6 : Operan operator sizeof tidak boleh berisi ekspresi apa pun yang berpotensi menimbulkan efek samping

Kategori : Wajib

Analisis : Dapat Diputuskan, Unit Terjemahan Tunggal

Berlaku untuk : C90, C99

Amplifikasi : Ekspresi apa pun yang muncul dalam operan operator sizeof biasanya tidak dievaluasi. Aturan ini mengharuskan bahwa evaluasi ekspresi tersebut tidak boleh berisi efek samping, baik dievaluasi atau tidak. Pemanggilan fungsi dianggap sebagai efek samping untuk tujuan aturan ini.

Alasan : Operan operator sizeof dapat berupa ekspresi atau dapat menentukan tipe. Jika operan berisi ekspresi, kesalahan pemrograman yang mungkin terjadi adalah mengharapkan ekspresi tersebut dievaluasi padahal sebenarnya tidak dievaluasi dalam sebagian besar keadaan. Standar C90 menyatakan bahwa ekspresi yang muncul dalam operan tidak dievaluasi saat dijalankan. Dalam C99, ekspresi yang muncul dalam operan biasanya tidak dievaluasi saat dijalankan. Namun, jika operan berisi tipe array dengan panjang variabel maka ekspresi ukuran array akan dievaluasi.

2.2 State Of The Art

Tabel 1. *State of The Art*

Penulis dan Tahun	Judul	Hasil Review
(Visentin et al., 2023)	A Mixed-Autonomous Robotic Platform For Intra-Row And Inter-Row Weed Removal For Precision Agriculture. <i>Computers and Electronics in Agriculture</i> , 214.	Jurnal ini membahas pengembangan sistem robotik campuran untuk penghilangan gulma di lahan pertanian, yang terdiri dari rover kendali jarak jauh dan sistem robotik otonom berbasis <i>gantry</i> , <i>gripper</i> , serta kamera RGB-D. Dengan memanfaatkan visi komputer dan kecerdasan buatan, sistem ini mampu mengenali dan menghilangkan gulma secara presisi. Meski masih dalam tahap prototipe, sistem ini menunjukkan potensi besar untuk ditingkatkan, baik dari sisi perangkat keras seperti penambahan GPS RTK, LIDAR, sonar, dan sensor lainnya guna mendukung navigasi otonom—maupun perangkat lunak, terutama dalam sistem pengenalan tanaman melalui pelatihan jaringan saraf tiruan agar dapat mengenali berbagai warna tanaman. Proyek ini menunjukkan kelayakan teknologi robotik dalam mendukung pertanian berkelanjutan yang efisien, aman, dan ramah lingkungan, serta membuka peluang besar untuk inovasi lebih lanjut dalam sektor agrikultur.
(Shamshiri et al., 2024)	A modular sensing system with CANBUS communication for assisted navigation of an	Jurnal ini melaporkan desain, pengembangan, dan evaluasi sistem sensor modular dengan arsitektur kontrol

	agricultural mobile robot. <i>Computers and Electronics in Agriculture</i> , 223.	terdistribusi dan algoritma berbasis pengetahuan fuzzy untuk membantu navigasi otonom robot pertanian beroda empat (4WS) di antara barisan tanaman, dengan kemampuan menghindari rintangan. Sistem ini divalidasi melalui simulasi dan uji lapangan, membuktikan keandalannya dalam menyesuaikan arah kemudi dan kecepatan robot terhadap berbagai jenis hambatan. Perangkat keras dirancang modular agar mudah diganti saat rusak, sementara penggunaan sensor inframerah dan laser terbukti efektif memberikan umpan balik real-time di kondisi cahaya luar ruangan, sebagai alternatif ekonomis dari LiDAR mahal. Sistem komunikasi menggunakan protokol CANBUS terbukti efisien, dan penggunaan ECU modular meningkatkan skalabilitas serta mengurangi beban komputasi. Metodologi ini dapat diterapkan pada robot lain di kebun dengan konfigurasi baris yang berbeda, dan pengembangan selanjutnya mencakup integrasi dengan pembelajaran mesin serta kendali jarak jauh melalui VR dengan jaringan LoRaWAN dan 5G.
(Yang et al., 2023)	A review of core agricultural robot technologies for crop productions. <i>Computers and Electronics in Agriculture</i> , 206.	Artikel ini mereview teknologi inti robot pertanian, meliputi navigasi, manipulasi, deteksi target, dan perencanaan gerak. Mobilitas dibagi menjadi roda, track, dan kaki; manipulasi mencakup manipulator serial, paralel, koordinat, dan lunak. End-effector dikembangkan dalam tipe pemotong dan genggam-pemotong. Deteksi target dilakukan dengan metode tradisional dan cerdas. Perencanaan gerak terbagi menjadi global dan lokal, didukung sistem persepsi untuk mendeteksi kondisi alat dan kualitas kerja. Tren ke depan mencakup end-effector bionik fleksibel dengan kemampuan persepsi, algoritma visual komprehensif, edge computing, serta pengembangan robot multiagen dan kolaboratif.
(Jo et al., 2024)	A suction cup-based soft robotic gripper for cucumber harvesting: Design and	Penelitian ini mengusulkan gripper robotik lunak berbasis suction cup untuk panen mentimun. Gripper ini dirancang untuk

	validation. <i>Biosystems Engineering</i> , 238, 143–156.	menyesuaikan bentuk dan permukaan agar sesuai dengan karakteristik mentimun, meningkatkan daya cengkeram dan efektivitas adsorpsi. Hasil uji menunjukkan peningkatan tingkat keberhasilan genggamannya hingga 100% dan peningkatan radius efektif sebesar 35,6%. Gripper kemudian diintegrasikan dengan modul pemotong dan diuji di lapangan sebanyak 174 kali, menghasilkan tingkat keberhasilan panen 86,2% dan tingkat kerusakan 4,7%. Variasi keberhasilan antar lokasi disebabkan oleh perbedaan karakteristik batang buah. Modul genggam bekerja independen dari modul pemotong, sehingga tetap efektif meski modul pemotong diganti. Penelitian ini menjadi dasar pengembangan robot panen mentimun otonom di masa depan.
(Pekaslan et al., 2020)	ADONiS-Adaptive Online Nonsingleton Fuzzy Logic Systems. <i>IEEE Transactions on Fuzzy Systems</i> , 28(10), 2302–2312.	Artikel ini mengusulkan robot paralel hibrida redundan 6+3 DOF yang backdrivable, dengan desain yang menghindari singularitas dan mengurangi inersia melalui penempatan aktuator dekat basis. Gripper dapat dioperasikan tanpa aktuator tambahan di end-effector. Robot diuji dengan kontrol impedansi sederhana tanpa sensor, menghasilkan interaksi manusia-robot yang halus. Ke depannya, robot akan digunakan pada sistem gantry untuk tugas industri presisi.
(Gollakota & Srinivas, n.d.)	AGRIBOT-A MULTIPURPOSE AGRICULTURAL ROBOT.	Penelitian ini mengembangkan dan mendemonstrasikan robot AGRIBOT yang dirancang untuk membajak lahan pertanian. Robot ini diharapkan dapat membantu petani meningkatkan efisiensi kerja di ladang. Proyek ini merupakan hasil penelitian tingkat sarjana.
(TIAR 2015 : 2015 IEEE International Conference on Technological Innovations in ICT for	TIAR 2015 : 2015 IEEE International Conference on Technological Innovations in ICT for Agriculture and Rural Development : proceedings : 10th - 12th July 2015, Chennai, Tamil Nadu, India.	Artikel ini membahas adopsi robotika dalam pertanian sebagai langkah revolusioner menuju masa depan yang lebih produktif, berkelanjutan, dan berbasis teknologi. Penggunaan agribot untuk pertanian presisi, panen otomatis, pengendalian gulma, dan pemantauan lingkungan dapat meningkatkan efisiensi, pemanfaatan sumber daya, serta ketahanan ekosistem pertanian. Kemajuan

Agriculture and Rural)		AI, machine learning, dan IoT mendorong kemampuan adaptif dan komunikasi agribot. Meski menghadapi tantangan seperti biaya awal, keamanan data, dan regulasi, kolaborasi diperlukan untuk mengatasinya. Artikel juga menyoroti pentingnya pertimbangan etis dan lingkungan serta potensi penerapan agribot di negara berkembang seperti India.
(Raikov & Abrosimov, 2022)	Artificial Intelligence and Robots in Agriculture. <i>Proceedings of 2022 15th International Conference Management of Large-Scale System Development, MLSD 2022.</i>	Artikel ini membahas tantangan robot dalam menggantikan peran manusia di pertanian, terutama karena kompleksitas lingkungan dan variasi objek. Manusia unggul dalam persepsi aktif, pembelajaran, koordinasi mata-tangan, dan kemampuan tangan yang fleksibel. Untuk menghadapi kondisi agrikultur yang dinamis, robot perlu dilengkapi dengan teknologi canggih dalam persepsi, pemodelan, perencanaan, kontrol, dan sistem genggam adaptif. Tantangan ini masih menjadi fokus utama dalam riset robotika pertanian.
(Van Henten et al., 2022)	Agricultural Robotics and Automation [TC Spotlight]. <i>IEEE Robotics and Automation Magazine</i> , 29(4), 145–147.	Artikel ini membahas integrasi robotika cerdas dengan platform digital dalam pertanian untuk meningkatkan kolaborasi, efisiensi, dan inovasi. Platform digital seperti “Digital Agriculture” di Rusia mencakup berbagai layanan seperti pengelolaan lahan, pelacakan produk, dan prakiraan cuaca. Dengan dukungan teknologi seperti IoT dan penginderaan jauh, integrasi ini bertujuan meningkatkan produktivitas dan daya saing pertanian.
(Li et al., 2024)	Assessing fruit hardness in robot hands using electric gripper actuators with tactile sensors. <i>Sensors and Actuators A: Physical</i> , 365.	Artikel ini membahas sistem penilaian kekerasan buah menggunakan robot yang menggabungkan penglihatan visual, sensor taktil, dan jaringan saraf TSACN. Robot mendeteksi buah, menggenggamnya, lalu menilai kekerasannya melalui data taktil untuk mencegah kerusakan saat menggenggam. Hasil eksperimen menunjukkan sistem ini efektif dalam menilai kekerasan buah dan mendukung kontrol genggam yang presisi. Penelitian ini menunjukkan potensi besar dalam pengembangan sistem genggam robotik

		adaptif, terutama jika dikombinasikan dengan sensor taktil generasi baru dan strategi kontrol yang lebih fleksibel.
(Rodríguez-Nieto et al., 2025)	Automated Seeding in Hydroponic Greenhouse with a Dual-Arm Robotic System. <i>IEEE Access</i> .	Jurnal ini mengusulkan robot mobile lengan ganda untuk otomatisasi penyemaian hidroponik. Sistem mampu menyemai dengan presisi tinggi dalam 75 detik per kubus rock-wool dan mencapai tingkat keberhasilan 100% pada 45 uji coba. Dilengkapi end effector khusus dan deteksi visual berbasis YOLO, sistem ini efektif dan efisien. Riset selanjutnya fokus pada seleksi benih, evaluasi pascatanam, dan peningkatan portabilitas.
(Proceedings of IEEE CONECCT, 2020)	<i>International Conference on Electronics, Computing and Communication Technologies : July 2-4, 2020. (2020). IEEE.</i>	Jurnal ini membahas Agrobot, robot pertanian murah dan efisien yang mampu melakukan pengeboran, penanaman benih, penyiraman, deteksi rintangan, dan otomatisasi. Dalam uji coba di area 6,3 m × 3,3 m, Agrobot menyelesaikan tugas dalam 14 menit. Dirancang hemat daya dan biaya, robot ini mudah dimodifikasi sesuai kebutuhan petani dan berpotensi merevolusi sistem pertanian dengan biaya rendah.
(Al-Mashhadani & Park, 2023)	<i>Autonomous Agricultural Monitoring Robot for Efficient Smart Farming.</i>	Jurnal ini membahas sistem robotik pertanian yang mengintegrasikan navigasi otonom, deteksi kematangan tomat, identifikasi penyakit, dan pemantauan kelembaban tanah. Sistem menggunakan teknologi SLAM, algoritma A*, YOLOv5, dan sensor berbasis ESP8266. Hasilnya menunjukkan akurasi tinggi di kondisi nyata. Dataset tersedia di Kaggle. Sistem ini berpotensi meningkatkan produktivitas dan pengambilan keputusan di pertanian cerdas. Riset selanjutnya diarahkan pada kolaborasi multi-robot, algoritma perencanaan jalur dinamis, dan penyesuaian untuk berbagai jenis tanaman.
(ICVEE, 2020)	<i>strengthening the framework of Society 5.0 through Innovations in Education, Electrical, Engineering and Informatics Engineering : proceeding</i>	Artikel ini membahas penggunaan robot otonom dalam pertanian untuk meningkatkan efisiensi dan produktivitas. Teknologi seperti GPS, kamera CCD, dan laser digunakan untuk navigasi dan panen otomatis. Hasil uji coba menunjukkan akurasi tinggi dalam pengenalan tanaman

		dan pemetikan buah. Penelitian selanjutnya diarahkan pada penerapan komersial dan pendekatan kolaboratif dengan petani.
(Ghafar et al., 2021)	Design and development of a robot for spraying fertilizers and pesticides for agriculture. <i>Materials Today: Proceedings</i> , 81(2), 242–248.	Artikel ini membahas pengembangan prototipe robot pertanian berbiaya rendah untuk penyemprotan pupuk dan pestisida serta pemantauan tanaman. Robot dua roda ini dilengkapi kamera dan sistem kontrol nirkabel, serta mampu menyemprot 20 tanaman per menit secara otonom. Meskipun sedikit lebih lambat dari pekerja manusia, robot ini lebih hemat biaya tenaga kerja dan lebih efisien dalam penggunaan bahan kimia karena hanya menyemprot pada tanaman. Daya tahan baterainya cukup untuk dua shift kerja. Ke depannya, robot ini disarankan untuk ditingkatkan menjadi sepenuhnya otonom agar efisiensinya makin optimal.
(Huang & Chang, 2019)	Design and Implementation of Bio-inspired Snake Bone-Armed Robot for Agricultural Irrigation Application. <i>IFAC-PapersOnLine</i> , 52(30), 98–101.	Penelitian ini menguji robot lengan tulang ular dengan sudut tekuk maksimum $133,88^\circ$, tanpa masalah tenaga. Akurasi penyemprotan menurun seiring bertambahnya jarak deviasi pusat, terutama di sudut-sudut. Akurasi tertinggi (100%) dicapai dalam jarak 10 cm. Perbaikan ke depan mencakup peningkatan akurasi deviasi, penambahan fleksibilitas multi-arrah, dan integrasi dengan aplikasi mobile untuk pemantauan dan kontrol jarak jauh.
(Cohen & Zarrouk, 2023)	Design, Analysis and Experiments of a High-Speed Water Hovering Amphibious Robot: AmphiSTAR.	Artikel ini memperkenalkan robot amfibi AmphiSTAR yang sangat mobile dan ringan (246 gram), mampu merayap di darat, berenang, dan melayang cepat di atas air. Robot ini terinspirasi dari kecoa dan kadal basilisk, dan terbukti menjadi salah satu robot amfibi tercepat dengan kecepatan hingga 13,6 panjang tubuh/detik di darat dan 5,6 di air. Hasil eksperimen menunjukkan bahwa gaya angkat meningkat dengan kecepatan baling-baling, namun dipengaruhi oleh perpindahan air dan turbulensi. Robot menunjukkan stabilitas alami dan mampu pulih dari gangguan kemiringan dengan cepat. Riset lanjutan akan fokus pada optimalisasi desain baling-baling dan

		distribusi beban ideal antara daya apung dan dorongan.
(Fadhaeel et al., 2022)	Design and development an Agriculture robot for Seed sowing, Water spray and Fertigation. <i>Proceedings of International Conference on Computational Intelligence and Sustainable Engineering Solution, CISES 2022</i> , 148–153.	Penelitian ini merancang robot pertanian multifungsi untuk penanaman benih, penyemprotan air, dan pemupukan. Robot ini dilengkapi sensor pH dan kelembaban tanah untuk menyesuaikan penyemprotan pupuk dan air secara efisien. Penanaman dilakukan otomatis melalui drum benih tanpa pemborosan. Data lingkungan seperti suhu dan kelembaban dibaca melalui platform ThingSpeak. Sistem ini menunjukkan potensi dalam meningkatkan efisiensi pertanian berbasis sensor.
(Wei et al., 2019)	Equipment and Intelligent Control System in Aquaponics: Institute of Electrical and Electronics Engineers Inc.	Makalah ini membahas sistem akuaponik yang mengintegrasikan budidaya ikan dan tanaman dengan efisiensi daur ulang air. Tiga metode hidroponik utama NFT, DFT, dan FCH diulas, serta teknologi filtrasi dan pemantauan lingkungan secara real-time. Tantangan utamanya adalah biaya tinggi dan keterbatasan sensor dalam lingkungan kompleks. Fokus riset ke depan meliputi pengembangan sensor andal, algoritma kontrol cerdas, dan model pertumbuhan untuk mendukung produksi berkelanjutan dan efisien.
(Pak et al., 2022)	Field Evaluation of Path-Planning Algorithms for Autonomous Mobile Robot in Smart Farms. <i>IEEE Access</i> , 10, 60253–60266.	Studi ini membandingkan algoritma perencanaan jalur untuk pertanian cerdas, yaitu Dijkstra, A*, RRT, dan RRT*. A* menunjukkan akurasi tinggi dalam semua skenario (jarak pendek, jauh, dan hambatan dinamis), serta paling cocok digunakan di lingkungan smart farm yang memerlukan presisi tinggi agar tanaman tidak rusak. Hasil eksperimen lapangan juga menegaskan keunggulan A* dalam efisiensi waktu dan lokasi.
(Udekwe & Seyyedhasani, 2025)	Human robot interaction for agricultural Tele-Operation, using virtual Reality: A feasibility study. <i>Computers and Electronics in Agriculture</i> , 228.	Penelitian ini mengevaluasi sistem teleoperasi robot berbasis VR untuk praktik pertanian, seperti panen buah dan inspeksi daun. Dengan menggunakan HTC Vive, sistem menunjukkan akurasi dan efisiensi tinggi, mencapai 90% tingkat keberhasilan interaksi pada panen buah dan waktu penyelesaian tugas rata-rata 18 detik.

		Meskipun terdapat tantangan persepsi jarak, sistem tetap efektif meski dalam kondisi jaringan terbatas. Studi ini menegaskan potensi VR dalam meningkatkan efisiensi pertanian dan mengurangi risiko kerja lapangan. Penelitian lanjutan disarankan di lingkungan pertanian terbuka.
(Dutta et al., 2021)	Multi-Robot Information Gathering for Precision Agriculture: Current State, Scope, and Challenges.	Artikel ini membahas tantangan dan kondisi terkini pengumpulan informasi multi-agen dalam pertanian presisi. Penggunaan banyak robot atau drone dianggap solusi utama untuk memenuhi kebutuhan data yang kompleks di lahan pertanian luas. Tiga tantangan utama yang diidentifikasi adalah perencanaan jalur, keamanan dan integritas informasi, serta efisiensi energi. Studi ini menekankan pentingnya menemukan keseimbangan antara jalur optimal, efisiensi energi, dan kebutuhan pertukaran informasi. Ditekankan juga bahwa meskipun beberapa masalah serupa dengan bidang lain, pertanian presisi memiliki kebutuhan unik yang membutuhkan pendekatan khusus.
(Luo et al., 2024)	Preliminary Design and Control of an Ankle Rehabilitation Robot with Variable Stiffness Actuator.	Artikel ini mengusulkan robot rehabilitasi pergelangan kaki dengan kekakuan variabel untuk meningkatkan kepatuhan robotik secara intrinsik. Aktuator kekakuan variabel dirancang dengan tiga set pegas daun dan sistem pengaturan sinkron berbasis prinsip self-centering. Model kekakuan disederhanakan menggunakan algoritma optimisasi partikel. Hasil simulasi menunjukkan efektivitas model dan strategi kontrolnya. Ke depannya, penelitian akan mempertimbangkan gesekan, membuat prototipe fisik, dan mengeksplorasi kontrol adaptif yang lebih robust untuk meningkatkan keamanan interaksi manusia-robot.
(2016 IEEE 4th, 2016)	<i>International Symposium on Robotics and Intelligent Sensors : “Empowering Robots with Smart Sensors”</i>	Makalah ini membahas lima tantangan utama dalam penerapan robot pertanian, yaitu: integrasi perangkat keras, konektivitas nirkabel, pengembangan perangkat lunak khusus pertanian, interaksi manusia-robot (HRI), serta reusabilitas dan keandalan perangkat lunak. Penulis menekankan bahwa

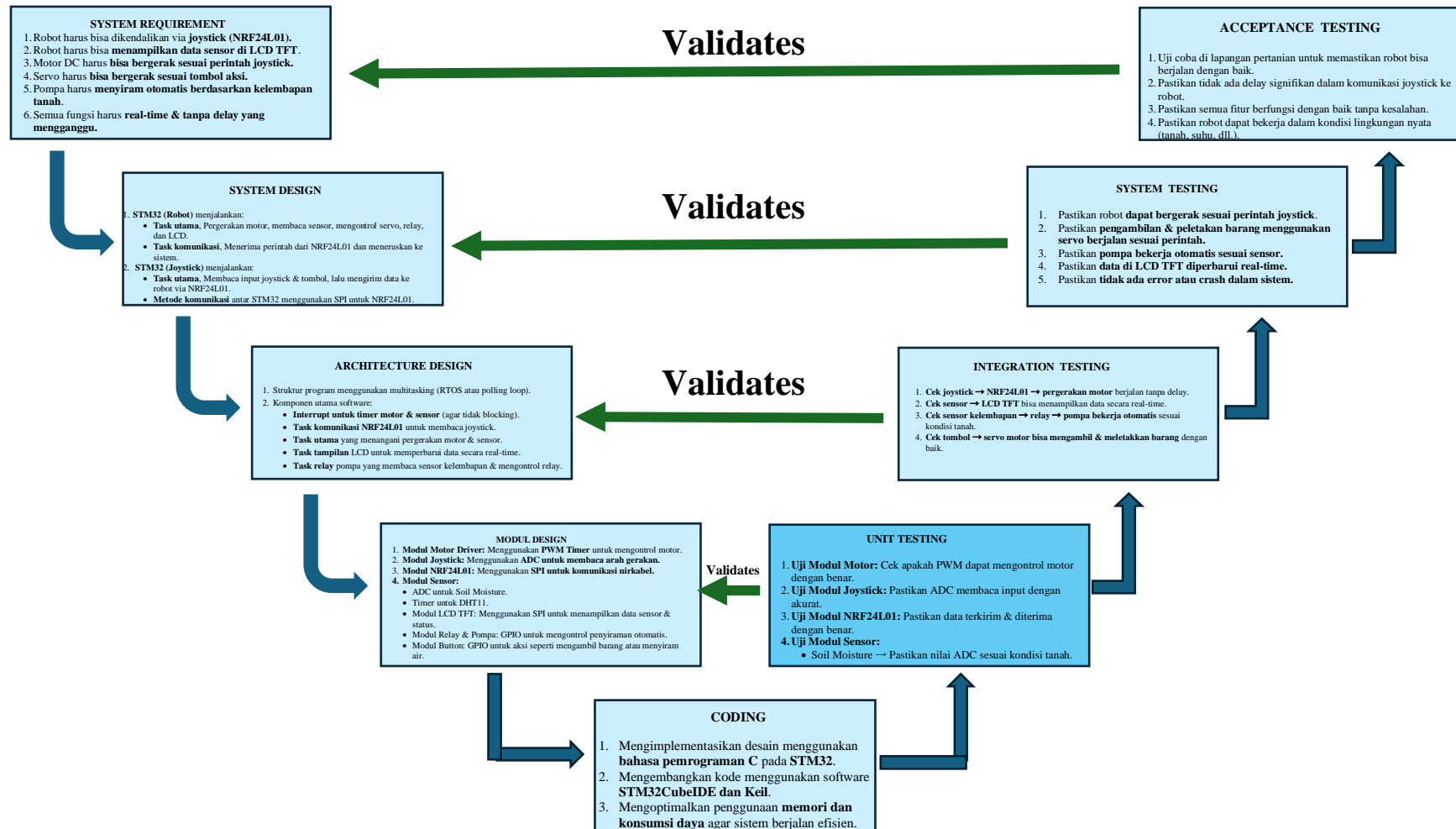
		tanpa infrastruktur pendukung yang memadai di aspek-aspek ini, robot pertanian sulit diterapkan secara praktis di lapangan.
(Noda et al., 2025)	Robot Simulation on Agri-field Point Cloud with Centimeter Resolution.	Penelitian ini mengembangkan simulator dinamika untuk robot pertanian yang mampu menangani model medan 3D berbasis point cloud dengan densitas tinggi, berkat kemajuan teknologi fotografi udara. Untuk mempercepat deteksi tabrakan, digunakan algoritma grid, dan gaya kontak dihitung dengan mengintegrasikan gaya antar titik tabrakan dan mesh robot. Evaluasi performa menunjukkan bahwa simulasi berjalan 8,8 kali lebih cepat dari waktu nyata, menunjukkan efisiensi tinggi dalam perhitungan dinamika untuk aplikasi di berbagai kondisi lapangan pertanian.
(Rehman et al., 2024a)	Smart agriculture technology: An integrated framework of renewable energy resources, IoT-based energy management, and precision robotics.	Artikel ini menyimpulkan bahwa prototipe yang dikembangkan mulai dari REI untuk efisiensi energi, integrasi PV dan BESS, hingga sistem pemantauan lingkungan berbasis IoT dan robotika aplikasi Android efektif dan aplikatif dalam konteks pertanian cerdas. Integrasi teknologi ini melalui platform Blynk IoT dan Bluetooth RC menunjukkan pendekatan yang saling terhubung dan efisien. Hasil studi menekankan manfaat ekonomi dan lingkungan dari penggunaan energi terbarukan dan otomatisasi dalam praktik pertanian. Ke depan, riset diarahkan pada adaptasi teknologi ini untuk berbagai jenis tanaman dengan kebutuhan energi rendah dan sistem yang lebih terjadwal, fleksibel, dan skalabel.
(Rehman et al., 2024)	<i>Smart agriculture technology: An integrated framework of renewable energy resources, IoT-based energy management, and precision robotics. Cleaner Energy Systems.</i>	Penelitian ini menyoroti efektivitas berbagai prototipe teknologi dalam konteks pertanian cerdas, termasuk REI untuk manajemen energi, sistem monitoring berbasis IoT untuk irigasi presisi, dan robot aplikasi kimia berbasis aplikasi Android. Integrasi prototipe melalui platform Blynk dan Bluetooth menunjukkan potensi besar untuk membentuk ekosistem pertanian yang berkelanjutan dan efisien. Hasil studi kasus menegaskan manfaat ekonomi dan

		lingkungan dari pemanfaatan energi terbarukan. Arah penelitian selanjutnya akan difokuskan pada penerapan teknologi ini untuk jenis tanaman spesifik, dengan penyesuaian terhadap kebutuhan energi rendah dan fleksibilitas jadwal pertanian.
(Al Rakib et al., 2022)	<i>Smart Agriculture Robot Controlling using Bluetooth. European Journal of Engineering and Technology Research.</i>	Penelitian ini membahas pengembangan sistem irigasi cerdas sebagai solusi atas keterbatasan metode irigasi manual yang masih banyak digunakan di negara berkembang seperti Nepal. Sistem ini dirancang untuk mengurangi pemborosan air dan meningkatkan efisiensi irigasi, yang berdampak langsung pada biaya dan hasil produksi pertanian. Setelah proses perencanaan dan perakitan komponen, sistem berhasil diuji dan menunjukkan hasil yang sesuai harapan. Solusi ini dinilai mampu menggantikan metode konvensional yang kurang efisien dan boros air.
(Rosero-Montalvo et al., 2023)	<i>Smart Farming Robot for Detecting Environmental Conditions in a Greenhouse.</i>	Jurnal ini menyimpulkan bahwa jaringan sensor nirkabel berhasil mengumpulkan data lingkungan yang akurat dari dalam greenhouse, berkat metode pengambilan sampel rata-rata. Kontroler PID yang dirancang memungkinkan perencanaan lintasan robot yang fleksibel sesuai bentuk tanaman. Analisis data berhasil mengidentifikasi zona bermasalah di greenhouse, seperti area dekat pintu dan pompa air. Sebagai pengembangan ke depan, penulis berencana membangun robot otonom dengan visi komputer untuk memberikan informasi visual dan kontrol otomatis terhadap sistem irigasi.
(Auat Cheein & Carelli, 2013)	Agricultural robotics: Unmanned robotic service units in agricultural tasks. <i>IEEE Industrial Electronics Magazine.</i>	Artikel ini mengulas teknologi terbaru pada unit layanan tanpa awak (unmanned service units) di bidang pertanian, dengan menyoroti empat kemampuan utama: deteksi, panduan, pemetaan, dan aksi. Ditekankan bahwa akurasi sistem pelokalan sangat penting, karena kesalahan posisi dapat menyebabkan pemetaan yang tidak akurat dan kegagalan tugas pertanian. SLAM diusulkan sebagai solusi murah untuk masalah pelokalan. Studi kasus pada pengawasan kebun zaitun turut

		<p>mendukung pembahasan. Artikel ini memberikan landasan kuat bagi pengembangan lebih lanjut teknologi robotika pertanian.</p>
--	--	--

BAB 3. METODOLOGI

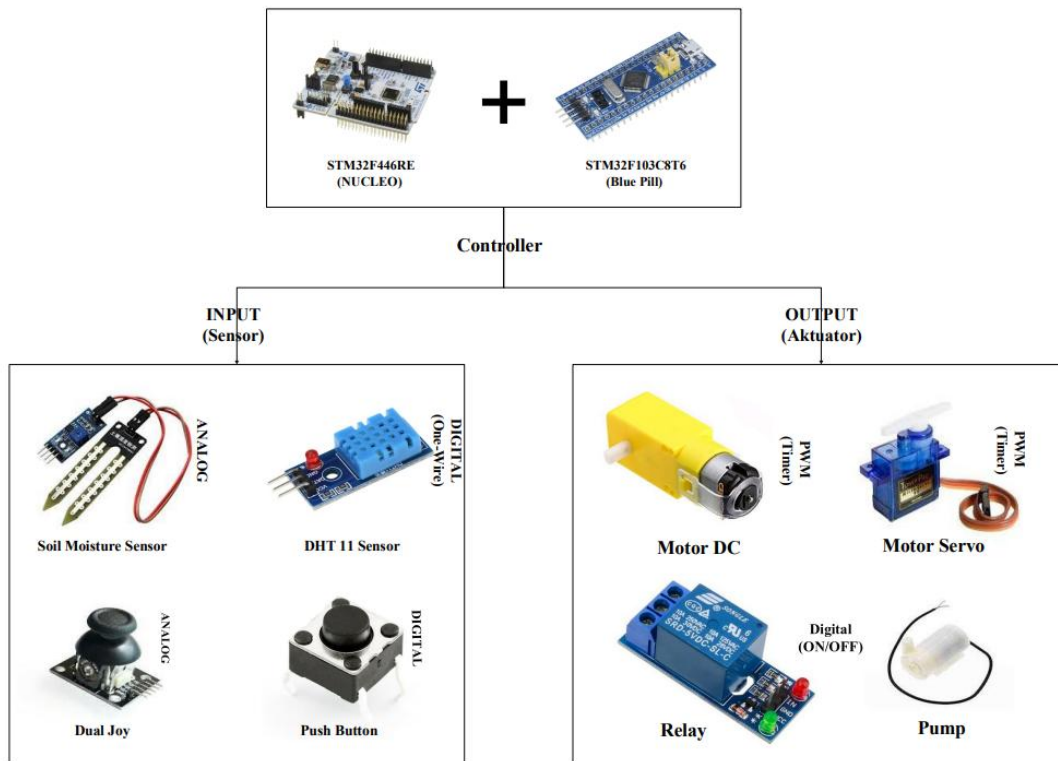
3.1 Desain V-Model



Gambar 1. Model Software System AgroBot

V-Model diatas dibuat sebagai pendekatan rekayasa perangkat lunak yang membagi proses pengembangan menjadi dua sisi utama: *tahap perancangan (development)* di sebelah kiri dan *tahap pengujian (testing)* di sebelah kanan. Setiap tahap perancangan memiliki pasangan uji yang sesuai untuk memastikan kesesuaian fungsi dan kualitas sistem. Model V memastikan pengembangan sistem Agrobot dilakukan secara terstruktur, dengan pengujian menyeluruh untuk menjamin fungsionalitas, keakuratan, dan keandalan dalam kondisi nyata di lahan pertanian.

3.2 Arsitektur Model Desain



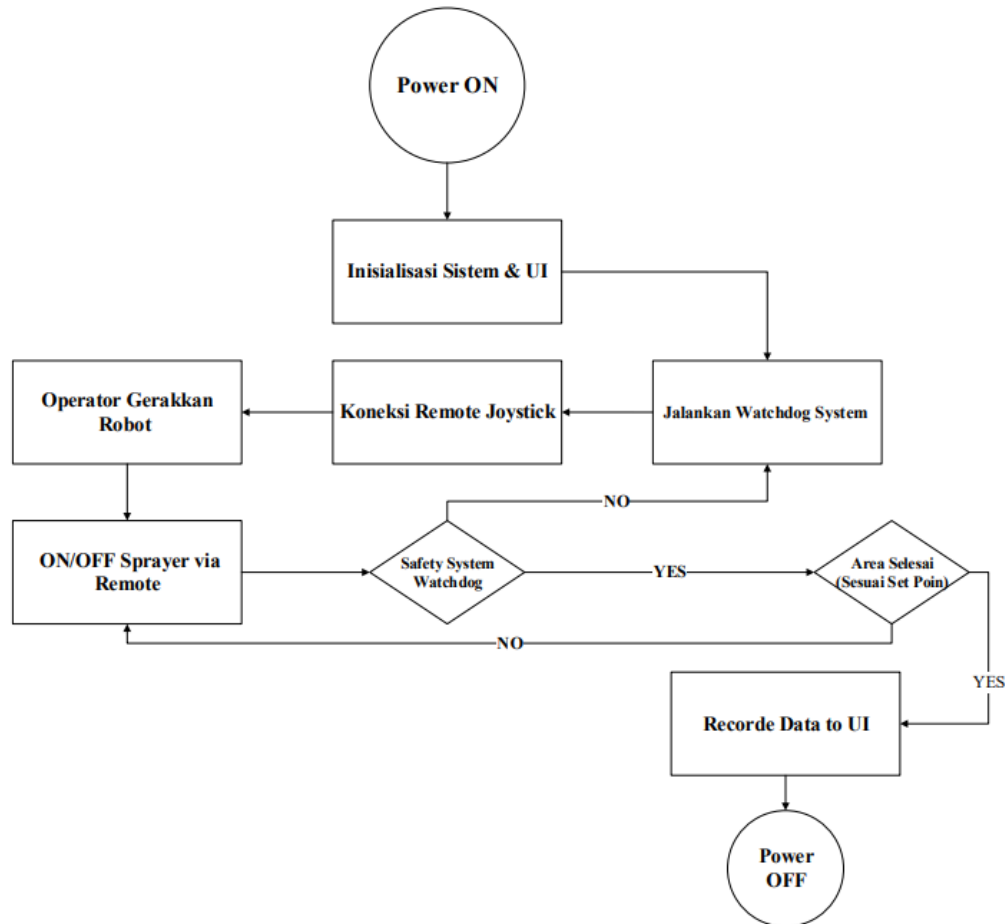
Gambar 2. *Architecture Design*

Arsitektur desain sistem diatas menggambarkan integrasi antar komponen dalam suatu sistem kendali berbasis mikrokontroler. Sistem terdiri dari dua unit mikrokontroler utama, yaitu STM32F103C8T6 (Blue Pill) dan STM32F446RE (NUCLEO). STM32F103C8T6 berfungsi sebagai sensor hub, yang bertugas membaca data dari sensor Soil Moisture Sensor, DHT11, serta Push Button dan Joystick. Data yang diperoleh kemudian dikirimkan melalui Wireless Communication ke mikrokontroler utama STM32F446RE menggunakan NRF.

STM32F446RE berperan sebagai unit pengendali utama yang menerima data dari sensor hub, mengolahnnya berdasarkan logika yang telah diprogram, dan kemudian menentukan aksi yang akan dilakukan. Hasil dari pemrosesan ini kemudian digunakan untuk mengendalikan berbagai aktuator, antara lain motor DC, motor servo, dan relay yang mengaktifkan pompa air. Selain itu, STM32F446RE juga dapat berkomunikasi dengan sistem perangkat lunak eksternal

(seperti aplikasi desktop atau mobile) untuk keperluan monitoring dan kontrol jarak jauh, memberikan visibilitas terhadap kondisi sistem dan memungkinkan interaksi pengguna. Sehingga dengan pembagian fungsi ini, arsitektur sistem menjadi modular dan efisien, di mana sensor hub fokus pada akuisisi data, sedangkan unit kontrol utama fokus pada pemrosesan dan pengambilan keputusan. Pendekatan ini meningkatkan skalabilitas serta memudahkan pemeliharaan dan pengembangan sistem lebih lanjut.

3.3 Flowchart

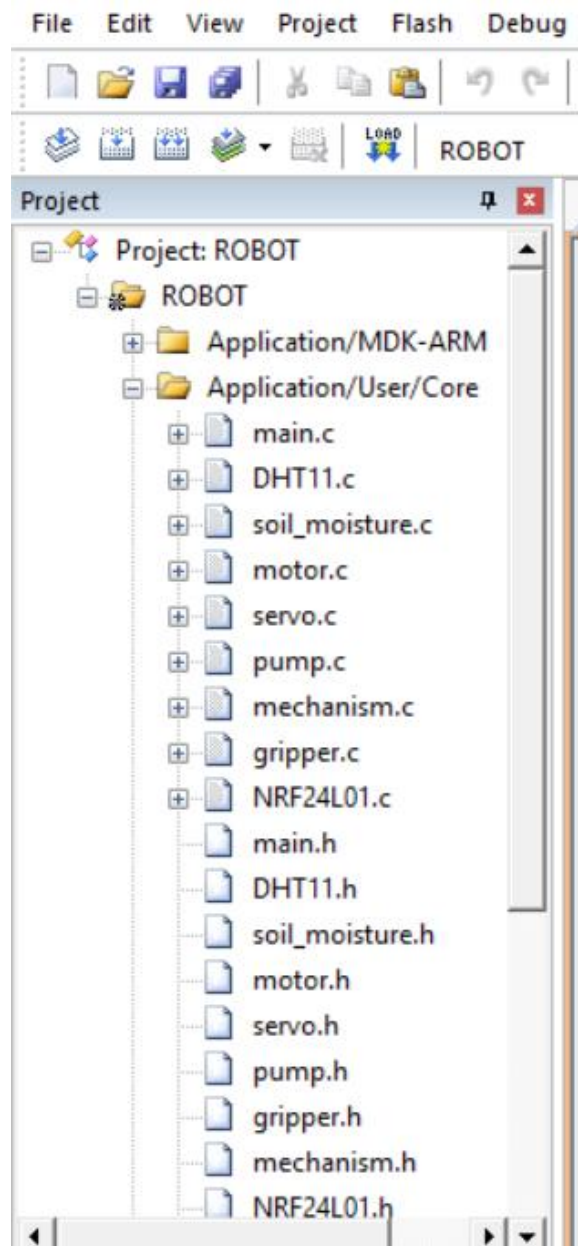


Gambar 3. Flowchart AgroBot

Alur kerja "AgroBot" robot penyemprot yang dikendalikan manual oleh operator. Setelah sistem dinyalakan dan *watchdog* aktif, operator mengendalikan robot menggunakan remote control untuk bergerak dan menyemprot area target. Selama proses, *watchdog* memantau kondisi sistem. Setelah seluruh area disemprot, robot dihentikan dan laporan hasil kerja dicatat. Alur ini menekankan kontrol manual dengan pengawasan otomatis untuk menjaga kestabilan dan keamanan sistem.

BAB 4. HASIL DAN PEMBAHASAN

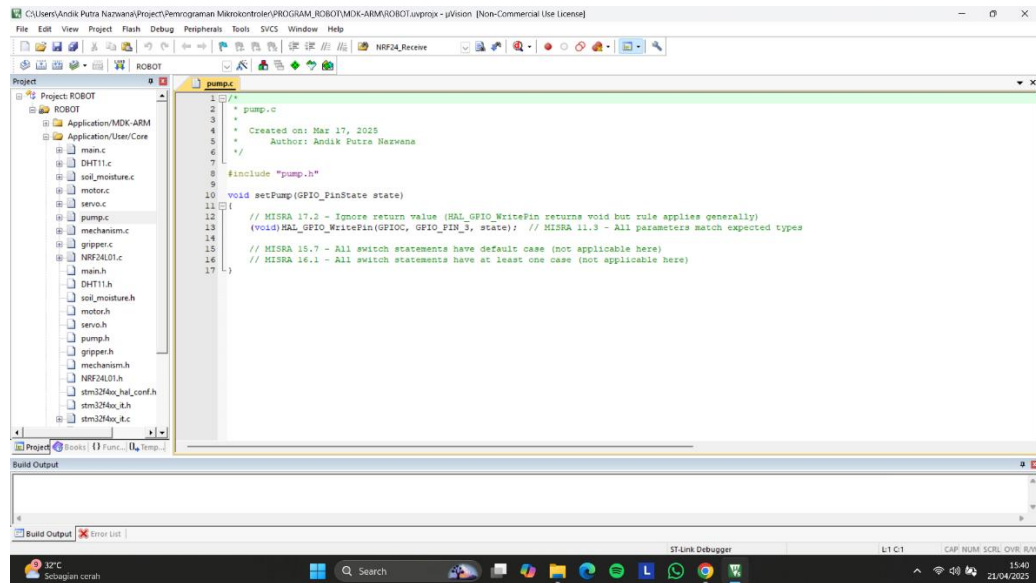
4.1 Struktur Program



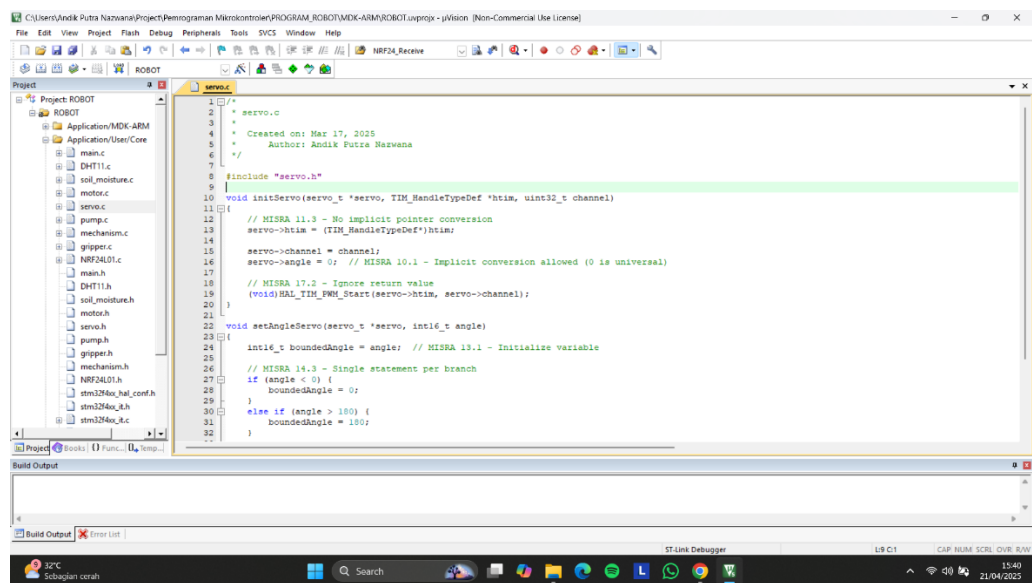
Gambar 4. *Structure Program AgroBot*

Gambar 4. Merupakan struktur program dari AgroBot yang dibuat secara terorganisir setiap komponen (modulnya). Masing-masing komponen *hardware* (sensor, aktuator, komunikasi) terdiri dari *file .c* untuk implementasi dan *.h* untuk deklarasi. Hal tersebut dilakukan karena cara praktik pemrograman yang baik dalam pengembangan *embedded system*.

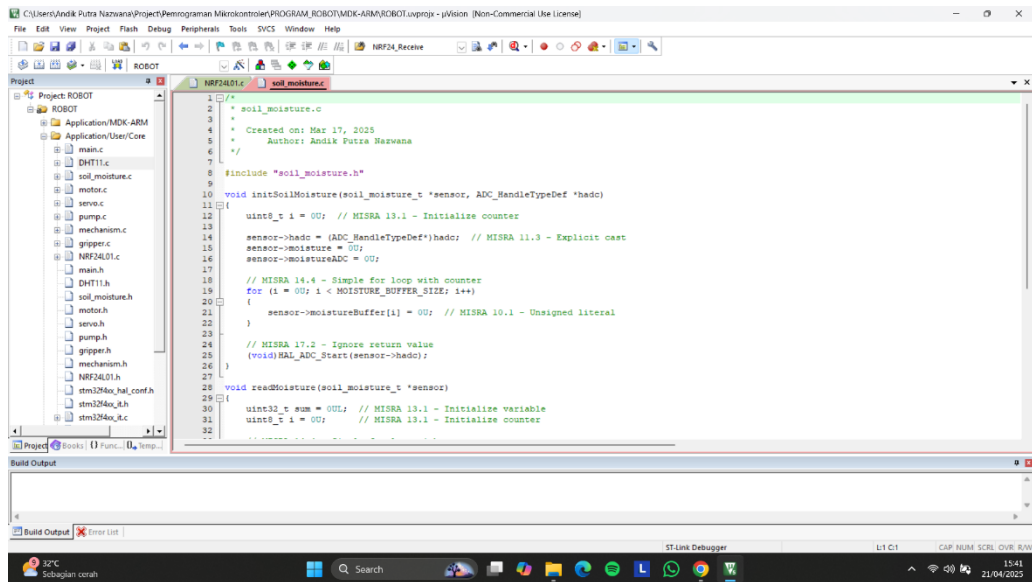
File *main.c* berperan sebagai pusat kendali program, tempat inisialisasi dan pengaturan alur utama berlangsung. Modul-modul lain seperti DHT 11, soil moisture, motor, servo, dan pump masing-masing mengatur sensor atau aktuator tertentu, seperti membaca suhu dan kelembapan, mengontrol motor penggerak, mengatur posisi servo, dan menghidupkan pompa. File *mechanism* berfungsi sebagai pengatur logika kerja robot secara keseluruhan, sedangkan *gripper* menangani sistem penjepit objek. Komunikasi data antar perangkat dilakukan melalui modul NRF24L01, yang memungkinkan robot berinteraksi secara nirkabel. Berikut merupakan dokumentasi setiap modul dari program project tugas kelompok 6 kami.



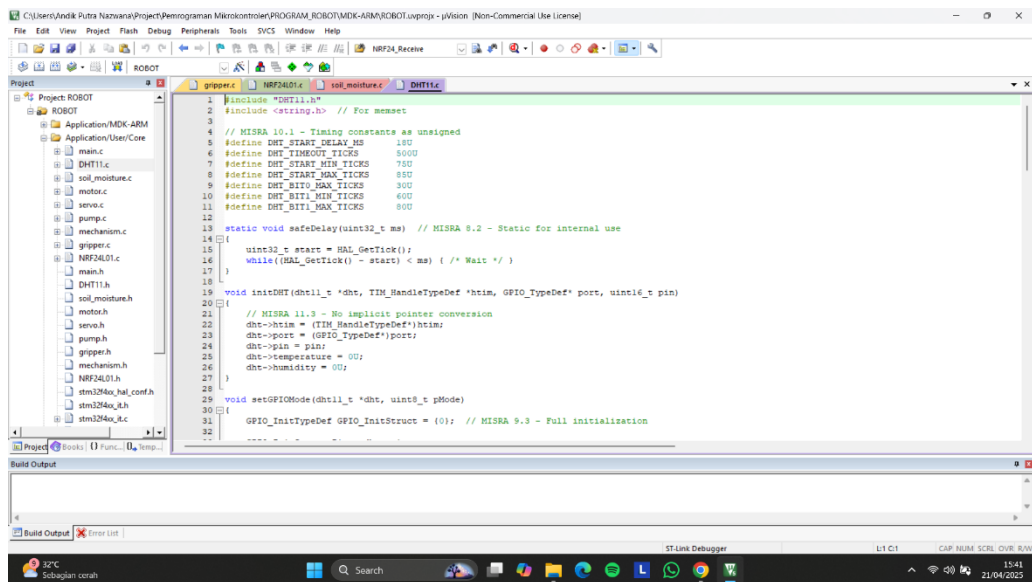
Gambar 5. Program Pump



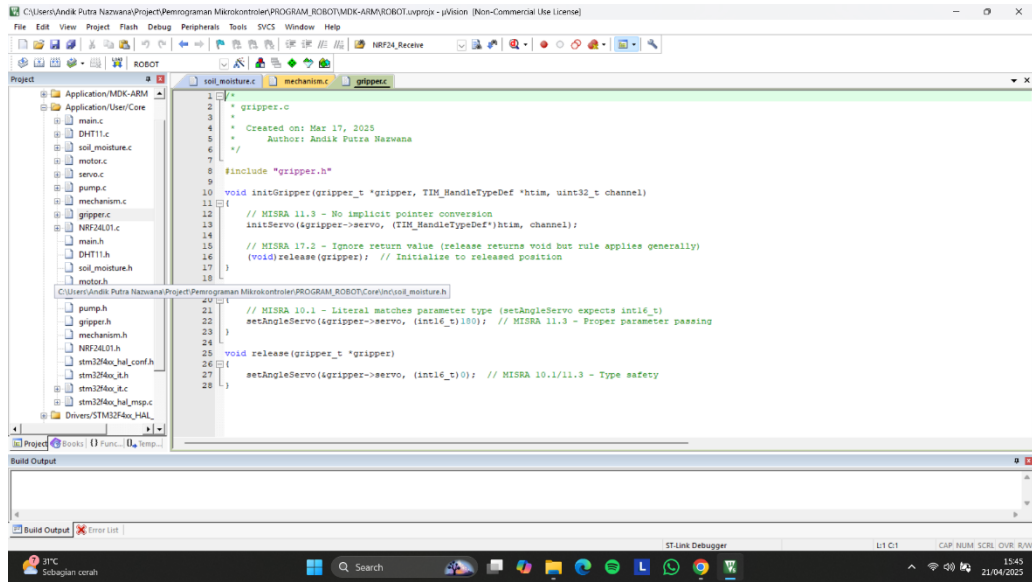
Gambar 6. Program Servo



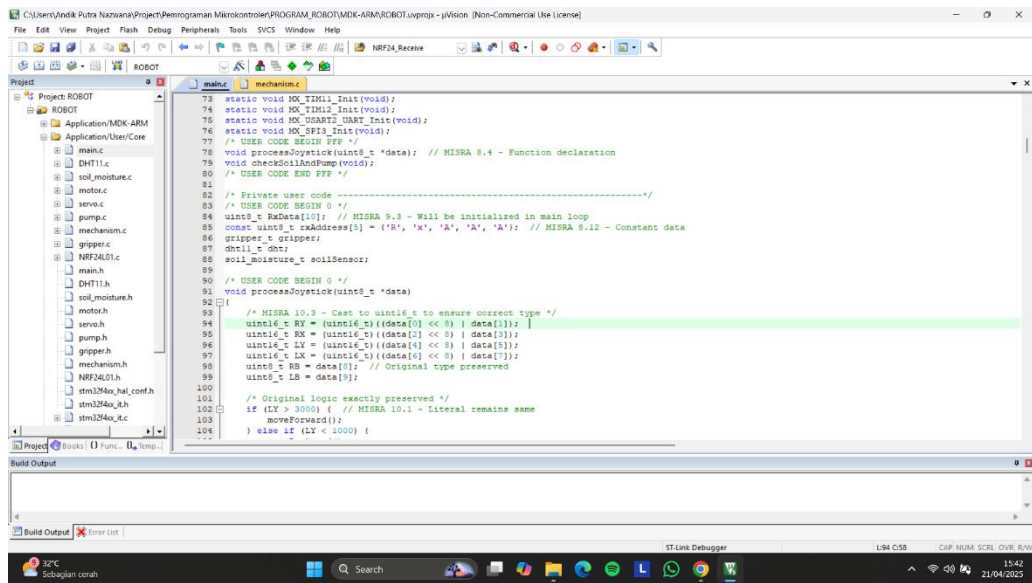
Gambar 7. Program Soil Moisture



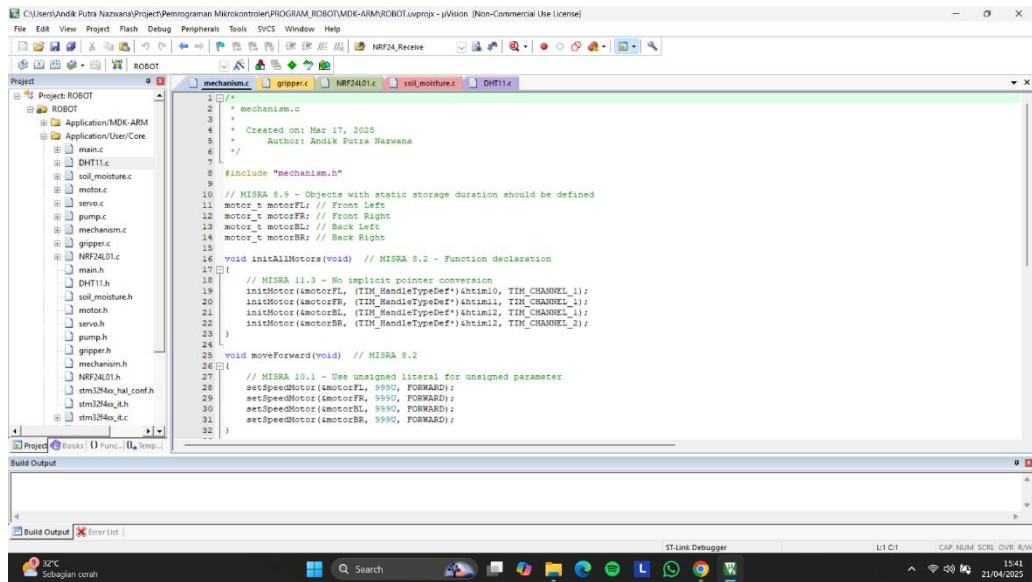
Gambar 8. Program DHT 11



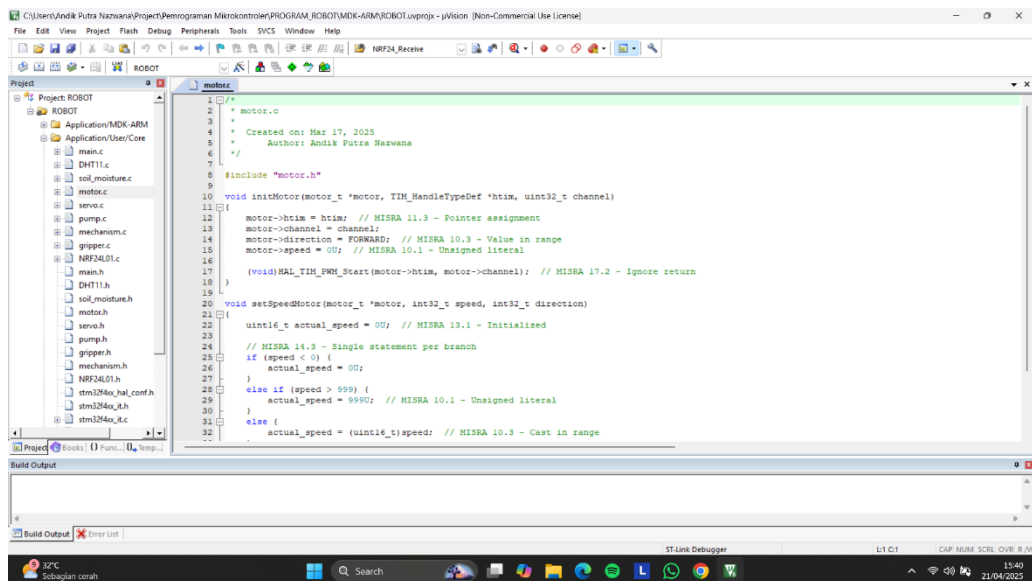
Gambar 9. Program Gripper



Gambar 10. Program Integration



Gambar 11. Program Mekanishm



Gambar 12. Program Motor

4.2 Functional Safety Dan Unit Testing

4.2.1 Tabel Functional Safety



Tabel checklist pengujian perangkat lunak *embedded system* sesuai standar *Functional Safety*.

Tabel 2. *Functional Safety*

Tanggal:	10 April 2025
Nama Mahasiswa:	1. Andik Putra Nazwana 2. Akhmad Maulvin Nazir 3. Rany Surya Oktavia
Tim:	Agrobot
Nama Supervisor:	Ahmad Radhy, S.Si., M.Si
Mata Kuliah:	Pemrograman Kontroller (3 SKS)
Jenis Kontroler:	STM32F446RE dan STM32F103C8T6
Nilai:	

No	Kategori Pengujian	Deskripsi Uji	Standard	Status	Catatan
1	Analisis Risiko (Hazard Analysis)	Mengidentifikasi potensi bahaya dalam sistem dengan menggunakan Watchdog Timer untuk mereset sistem secara otomatis saat terjadi program crash.	MISRA C:2012 Rule-Dir-4.1 "Run-time failures shall be minimized."	[v] Sudah	Dokumentasikan analisis FMEA atau HAZOP.
2	Static Code Analysis	Pemeriksaan kode secara menyeluruh menggunakan analisis statis tanpa	MISRA C:2012 Rule 2.2 — "There shall be no dead code."	[v] Sudah	Sesuaikan dengan standart pada dokumen MISRA.

		eksekusi program. Fokus pada pelanggaran terhadap aturan keselamatan (seperti aturan MISRA C 2012), penggunaan variabel yang tidak diinisialisasi, kondisi tak tercapai, memory leak, dan struktur kontrol yang tidak tepat.	Rule 9.1 — <i>"The value of an object shall be initialized before being used."</i> Rule 21.1 — <i>"Assembly language shall be encapsulated and isolated."</i>		
3	Gerak Forward Motor	Pengujian fungsional motor untuk bergerak maju saat diberi input 1. Validasi arah rotasi motor dan konfirmasi bahwa PWM dikirim ke kanal motor yang benar.	MISRA C:2012 Rule 13.2 — <i>"The value of an expression shall be the same under any order of evaluation."</i>	[v] Sudah	Validasi arah gerak maju. Motor FL, FR, BL, BR memiliki nilai 999 dan arahnya forward.
4	Gerak Backward Motor	Pengujian fungsional motor untuk bergerak mundur saat diberi input 0. Validasi arah rotasi motor dan konfirmasi bahwa PWM dikirim ke kanal motor yang benar.	MISRA C:2012 Rule 13.2 — <i>"The value of an expression shall be the same under any order of evaluation."</i>	[v] Sudah	Validasi arah gerak mundur. Motor FL, FR, BL, BR memiliki nilai 999 dan arahnya backward.
5	Motor Stop	Verifikasi bahwa motor benar-benar berhenti saat menerima perintah stop. Dicek juga apakah sinyal PWM menjadi nol atau logika arah dimatikan.	MISRA C:2012 Rule 14.3 — <i>"Controlling expressions shall not be invariant."</i>	[v] Sudah	Validasi berhenti. Motor FL, FR, BL, BR memiliki nilai 0 dan arahnya forward.
6	Turn Right Motor	Uji manuver belok kanan dengan memperlambat atau menghentikan motor kiri sambil tetap menggerakkan motor	MISRA C:2012 Rule 13.2 — <i>"The value of an expression shall be the same under any</i>	[v] Sudah	Validasi arah belok kanan. Motor FL, FR, BL, BR memiliki nilai 999. Arah FL BL

		kanan. Validasi bahwa robot berputar/berbelok sesuai ekspektasi.	<i>order of evaluation."</i>		Forward dan FR BR Backward.
7	Turn Left Motor	Sama seperti Turn Right, namun arah sebaliknya. Mengetes diferensial kecepatan motor untuk manuver kiri.	MISRA C:2012 Rule 13.2 — <i>"The value of an expression shall be the same under any order of evaluation."</i>	[v] Sudah	Validasi arah belok kanan. Motor FL, FR, BL, BR memiliki nilai 999. Arah FL BL Backward dan FR BR Forward.
8	Speed Motor	Mengukur kecepatan motor berdasarkan variasi input PWM. Uji linearitas antara nilai PWM dan kecepatan aktual	MISRA C:2012 Rule 12.2 — <i>"The value of an expression shall be the same under any order of evaluation."</i> Rule 10.3 — <i>"The value of an expression shall not be assigned to an object with a narrower essential type."</i>	[v] Sudah	Validasi input speed motor dengan range 0 – 999 dengan kondisi limit di 0 dan 999.
9	Soil Moisture	Uji akurasi pembacaan sensor kelembaban tanah. Bandingkan nilai ADC dengan tingkat kelembaban.	MISRA C:2012 Rule 12.1 — <i>"The precedence of operators within expressions shall be made explicit."</i> Rule 10.4 — <i>"A composite expression of a 'usual arithmetic conversion' shall not be</i>	[v] Sudah	Validasi ADC dengan skala RH%.

			<i>implicitly narrowed."</i>		
10	Pump On	Pengujian bahwa pompa aktif saat nilai kelembaban tanah < 40%. Validasi pemicu logika, aliran arus ke pompa, dan fungsionalitas pompa.	MISRA C:2012 Rule 14.4 — <i>"All if...else if constructs shall be terminated with an else clause."</i> Rule 13.2 — <i>"The value of an expression shall be the same under any order of evaluation."</i>	[v] Sudah	Validasi input 1 untuk fungsi Pump On
11	Pump Off	Memastikan pompa nonaktif saat kelembaban $\geq 40\%$. Validasi bahwa tidak terjadi delay, kebocoran logika, atau kondisi race.	MISRA C:2012 Rule 13.2 — <i>"The value of an expression shall be the same under any order of evaluation."</i>	[v] Sudah	Validasi input 0 untuk fungsi Pump On.
12	Motor Servo 0-180 Derajat	Pengujian bahwa motor servo bergerak dari sudut 0 hingga 180 derajat secara halus dan akurat. Uji juga posisi menengah dan respon terhadap perintah cepat.	MISRA C:2012 Rule 10.1 — <i>"Operands shall not be implicitly converted to a narrower essential type."</i> Rule 12.1 — <i>"Operator precedence shall be explicit."</i>	[v] Sudah	Validasi input sudut dengan PWM dan aktual sudut dalam derajat.
13	Grip On	Uji fungsionalitas gripper saat mengaktifkan genggam.	MISRA C:2012 Rule 13.3 — <i>"Expressions with side effects shall not be used within other expressions."</i>	[v] Sudah	Validasi input sudut untuk menggerakkan servo.

14	Grip Off	Uji fungsionalitas mekanisme gripper saat melepaskan objek.	MISRA C:2012 Rule 13.3 — <i>"Expressions with side effects shall not be used within other expressions."</i>	[v] Sudah	Validasi input sudut untuk menggerakkan servo.
15	Integrasi Data Joystick	Pengujian akurasi pembacaan input analog/digital dari joystick, pengolahan data di mikrokontroler.	MISRA C:2012 Rule 10.4 — <i>"Avoid implicit narrowing of arithmetic expressions."</i> Rule 14.4 — <i>"All conditional constructs shall be complete."</i>	[v] Sudah	Uji nilai ADC dari NRF24L01
16	Integrasi Data Soil and Pump	Uji apakah logika sistem membaca nilai kelembaban tanah dan mengaktifkan atau menonaktifkan pompa secara otomatis berdasarkan ambang batas (40%).	MISRA C:2012 Rule 13.2 — <i>"Evaluation order shall be consistent."</i> Rule 14.1 — <i>"All if statements shall have a well-defined block."</i>	[v] Sudah	Cek logika dan delay.
17	TX Mode Transceiver	Uji transmisi data NRF24L01 dari mikrokontroler utama ke penerima. Pastikan data dikirim dengan benar, tanpa korupsi, dan pada channel yang sesuai.	MISRA C:2012 Rule 21.1 — <i>"Assembly code shall be isolated."</i> Rule 21.2 — <i>"Standard library functions shall be proven safe if used in safety-critical code."</i>	[v] Sudah	Validasi TXMode, data, dan channel yang dikirim.
18	RX Mode Transceiver	Uji penerimaan data dari NRF24L01. Validasi integritas data, konsistensi alamat RX, dan	MISRA C:2012 Rule 21.1 — <i>"Assembly code shall be isolated."</i> Rule 21.2 —	[v] Sudah	Validasi RXMode, data, dan channel yang diterima.

		reaksi sistem terhadap data masuk.	<i>"Standard library functions shall be proven safe if used in safety-critical code."</i>		
--	--	------------------------------------	---	--	--

4.2.2 Tabel Test Case/Unit Testing



Tabel checklist Test Case/Unit Testing

Tanggal:	23 April 2024
Nama Mahasiswa:	1. Andik Putra Nazwana 2. Akhmad Maulvin Nazir 3. Rany Surya Oktavia
Tim:	AgroBot
Nama Supervisor:	Ahmad Radhy, S.Si., M.Si
Matakuliah:	Pemrograman Kontroller (3 SKS)
Jenis Kontroler:	STM32F446RE dan STM32F103C8T6
Nilai:	

Detail Langkah untuk Pengujian

1. Valid Input Test:
 - Pastikan input sesuai spesifikasi fungsi.
 - Gunakan alat pengukur seperti osiloskop, debugger, atau analyzer.
2. Boundary Test:
 - Tes dengan input di batas minimum dan maksimum yang diizinkan.
 - Gunakan simulator untuk kondisi ekstrem.
3. Power Failure Test:
 - Simulasikan gangguan daya pada perangkat keras.

- Verifikasi apakah sistem kembali ke kondisi aman setelah restart.

No.	Nama Modul	Function yang Diuji	Test Case	Langkah Pengujian	Hasil yang Diharapkan	Kriteria Keberhasilan	Status
1.	gripper.c	<i>void</i> grip	Validasi tes case 1 pada program gripper.c	1. Jalankan program unit testing pada VSCode 2. Input case 1. 3. Verifikasi Logika	Logika grip sesuai dengan yang diharapkan yakni set grip saat input 1.	Logika grip berjalan dengan benar	[] Belum [V] Sudah
2.	gripper.c	<i>void</i> realease	Validasi tes case 2 pada program gripper.c	1. Jalankan program unit testing pada VSCode 2. Input case 2. 3. Verifikasi Logika	Logika realease sesuai dengan yang diharapkan yakni set realease saat input 2	Logika realease berjalan dengan benar	[] Belum [V] Sudah
3.	motor.c	<i>void</i> setSpeedMotor	Valid Input Test: Menulis Tes case input set speed 0-999 dan direction forward atau backward	1. Jalankan program unit testing pada VSCode 2. Input set speed dan direction. 3. Verifikasi Logika	Logika speed sesuai dengan yang diharapkan yakni motor speed akan tetap dalam range 0-999 meskipun input lebih dari luar nilai itu, serta logika HIGH/LOW forward atau backward sudah benar.	Logika set speed berjalan dengan benar	[] Belum [V] Sudah
4.	pump.c	<i>void</i> setPump	Valid Input Test: Menulis Tes case input 1/0 untuk pump on atau pump off.	1. Jalankan program unit testing pada VSCode	Logika pump sesuai dengan yang diharapkan yakni saat input 1 akan pump on dan	Logika pump berjalan dengan benar	[] Belum [V] Sudah

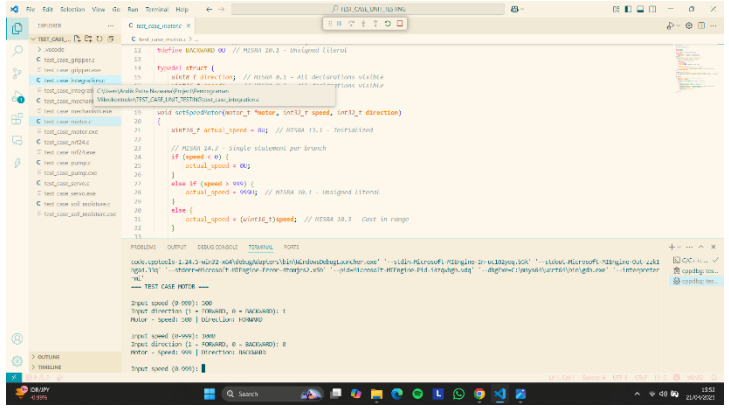
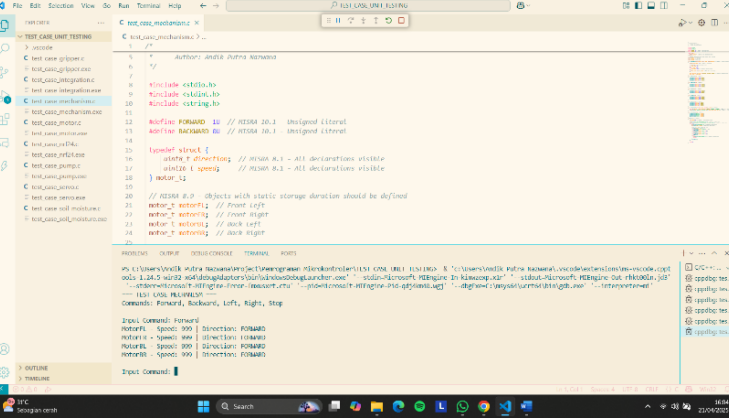
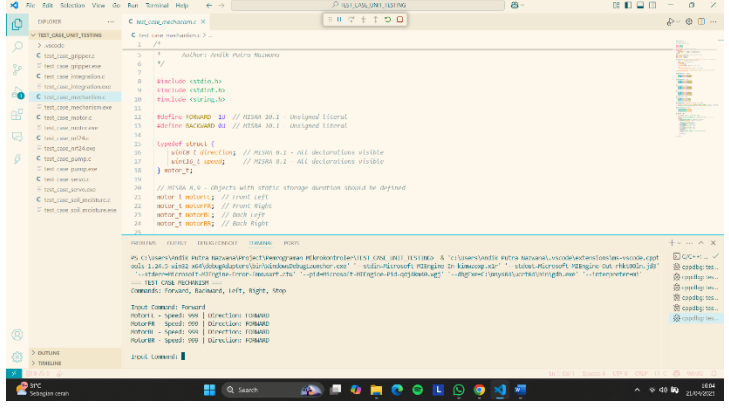
				2. Input set HIGH/LOW. 3. Verifikasi Logika	input 0 akan pump off		
5.	servo.c	<i>void setAngleServo</i>	Valid Input Test: Menulis Tes case set angle servo.	1. Jalankan program unit testing pada VSCode 6Input set angle. 2. Verifikasi Logika	Logika servo sesuai dengan yang diharapkan yakni saat saat angle sudut tertentu akan menghasilkan output PWM Value dan sudut yang sama dengan input.	Logika set angle servo berjalan dengan benar	[] Belum [V] Sudah
6.	Soil_moisture.c	<i>void readMoisture</i>	Valid Input Test: Menulis Tes case input ADC untuk mengetahui nilai <i>relative humidity</i> .	1. Jalankan program unit testing pada VSCode 2. Input set ADC. 3. Verifikasi Hasil RH%.	Logika membaca <i>moisture</i> sesuai dengan yang diharapkan yakni dengan input 4122 akan menghasilkan n RH 100%.	Logika membaca <i>moisture</i> berjalan dengan benar	[] Belum [V] Sudah
7.	mechanism.c	<i>void moveForward</i>	Valid Input Test: Menulis Tes case input <i>forward</i> untuk mengetahui motor speed dan motor direction	1. Jalankan program unit testing pada VSCode 2. Input command <i>forward</i> . 3. Verifikasi Logika	Logika <i>forward</i> sesuai dengan yang diharapkan yakni saat input <i>forward</i> , motor speed FR, FL, BR, BR 999 dan directionnya <i>forward</i> .	Logika <i>move forward</i> berjalan dengan benar	[] Belum [V] Sudah
8.	mechanism.c	<i>void moveBackward</i>	Valid Input Test: Menulis Tes case input	1. Jalankan program unit testing	Logika <i>backward</i> sesuai	Logika <i>move backward</i>	[] Belum

			<i>backward</i> untuk mengetahui motor speed dan motor direction	2. Input command <i>backward</i> . Verifikasi Logika	dengan yang diharapkan yakni saat input <i>backward</i> , motor speed FR, FL, BR, BR 999 dan directionnya <i>backward</i> .	berjalan dengan benar	[V] Sudah
9.	mechanis m.c	<i>void turnRight</i>	Valid Input Test: Menulis Tes case input <i>right</i> untuk mengetahui motor speed dan motor direction	1. Jalankan program unit testing pada VSCode 2. Input command <i>right</i> . 3. Verifikasi Logika	Logika <i>right</i> sesuai dengan yang diharapkan yakni saat input <i>right</i> , motor speed FL, BR, 999 dengan directionnya forward dan motor FR BL 999 dengan direction <i>backward</i> .	Logika <i>turn right</i> berjalan dengan benar	[] Belum [V] Sudah
10.	mechanis m.c	<i>void turnLeft</i>	Valid Input Test: Menulis Tes case input <i>left</i> untuk mengetahui motor speed dan motor direction	1. Jalankan program unit testing pada VSCode 2. Input command <i>left</i> . 3. Verifikasi Logika	Logika <i>left</i> sesuai dengan yang diharapkan yakni saat input <i>left</i> , motor speed FL, BR, 999 dengan directionnya <i>backward</i> dan motor FR BL 999 dengan direction forward.	Logika <i>turn left</i> berjalan dengan benar	[] Belum [V] Sudah
11.	mechanis m.c	<i>void stopsMotor</i>	Valid Input Test: Menulis Tes case input <i>stop</i> untuk mengetahui	1. Jalankan program unit testing pada VSCode	Logika <i>stop</i> sesuai dengan yang diharapkan yakni saat	Logika <i>stop motors</i> berjalan dengan benar	[] Belum [V] Sudah

			motor speed dan motor direction	2. Input command <i>stop</i> . 3. Verifikasi Logika	input <i>backward</i> , motor speed FR, FL, BR, BR 0 dan directionnya forward.		
12.	NRF24L01.c	<i>void TXMode RXMode</i>	Valid Test: Tes case status dan adress dan register untuk TXMode dan RXMode	1. Jalankan program unit testing pada VSCode 2. Verifikasi Logika	Logika <i>TXMode</i> dan <i>RXMode</i> sesuai dengan yang diharapkan yakni dapat menampilkan address dan register	Logika <i>TXMode</i> dan <i>RXMode</i> berjalan dengan benar	[] Belum [V] Sudah
13.	main.c	<i>void checkSoilMoisture</i>	Valid Input Test: Menulis Tes case input <i>RH%</i> untuk mengetahui Mengaktifkan atau mematikan pompa.	1. Jalankan program unit testing pada VSCode 2. Input value <i>RH%</i> . 3. Verifikasi Logika	Logika <i>Check Moisture</i> sesuai dengan yang diharapkan yakni saat kelembaban tanah dibawah 40% maka pompa akan menyala serta jika diatas 40% pompa akan mati.	Logika <i>check moisture</i> berjalan dengan benar	[] Belum [V] Sudah
14.	main.c	<i>void printInstruction</i>	Valid Input Test: Menulis Tes case input <i>ADC Joystick</i> untuk mengetahui Arah Gerak Motor.	1. Jalankan program unit testing pada VSCode 2. Input value <i>ADC Joystick</i> . 3. Verifikasi Logika	Logika <i>Print Instruction</i> sesuai dengan yang diharapkan yakni LY 4000 dengan RX 100 maka directionnya forward.	Logika <i>instruction</i> berjalan dengan benar	[] Belum [V] Sudah

4.3 Hasil Pengujian Dan Debugging

Tabel 3. Hasil Pengujian dan *Debugging*

Pengujian dan <i>Debugging</i>	Keterangan
	<p>Unit Testing Gerak Motor.</p> <p>Dalam pengujian ini digunakan untuk validasi arah gerak maju. Motor FL, FR, BL, BR memiliki nilai 999 dan arahnya forward, validasi arah gerak mundur. Motor FL, FR, BL, BR memiliki nilai 999 dan arahnya backward. Dan validasi berhenti. Motor FL, FR, BL, BR memiliki nilai 0 dan arahnya forward.</p>
	<p>Unit Testing Turn Right and Left Motor.</p> <p>Bertujuan untuk memvalidasi arah belok kanan. Motor FL, FR, BL, BR memiliki nilai 999. Arah FL BL Forward dan FR BR Backward. Dan validasi arah belok kanan. Motor FL, FR, BL, BR memiliki nilai 999. Arah FL BL Backward dan FR BR Forward.</p>
	<p>Unit Testing Speed Motor.</p> <p>Untuk memvalidasi input <i>speed</i> motor dengan <i>range</i> 0 – 999 dengan kondisi limit di 0 dan 999.</p>

Unit Testing Soil Moisture

Validasi ADC dan output skalanya adalah RH%.

Unit Testing Pump

Hasil pengujian memvalidasi
input 1 untuk fungsi Pump On.
Validasi input 0 untuk fungsi
Pump On.

Unit Testing Motor Servo

Hasil *Debugging* validasi input sudut dengan PWM dan aktual sudut dalam derajat.

BAB 5. PENUTUP

5.1 Kesimpulan

Berdasarkan hasil perancangan dan implementasi, sistem yang dikembangkan menunjukkan kinerja yang andal dan efisien dalam mendukung proses penyemprotan tanaman secara otomatis dan adaptif. Dengan fitur-fitur utama seperti navigasi berbasis *wireless controller*, penyemprotan otomatis yang menyesuaikan kondisi lingkungan, sistem *watchdog* untuk deteksi dan pemulihan crash, pemantauan status sistem secara *real-time*, serta efisiensi dalam konsumsi daya dan air, sistem ini mampu memberikan solusi yang cerdas dan praktis untuk otomasi pertanian skala kecil hingga menengah. Didukung oleh teknologi mikrokontroler STM32F446RE dan STM32F103C8T6, serta penggunaan sensor DHT11 dan *soil moisture sensor*, motor DC dan motor servo yang dikendalikan oleh driver L298N, sistem ini berhasil diintegrasikan menggunakan *framework* STM32CubeMX dan Keil uVision dengan bahasa pemrograman C. Kombinasi antara perangkat keras dan perangkat lunak yang tepat menjadikan sistem ini sebagai prototipe yang potensial untuk diterapkan lebih luas di sektor pertanian berbasis teknologi.

5.2 Saran

Berdasarkan Tugas Proyek yang sudah dilakukan, Tim Agrobot memberikan beberapa saran untuk Tugas Proyek seperti ini jika diterapkan di masa mendatang sebagai berikut :

1. Disarankan untuk mengintegrasikan sistem dengan *dashboard* IoT yang dapat diakses secara daring guna memantau kondisi dan aktivitas robot secara *real-time* dan dari jarak jauh. Hal ini akan meningkatkan efisiensi pemantauan dan memungkinkan intervensi lebih cepat saat terjadi anomali atau gangguan pada sistem.
2. Pengembangan sistem lebih lanjut dapat mencakup penambahan modul kecerdasan buatan (AI) untuk melakukan klasifikasi dan deteksi penyakit tanaman secara otomatis berdasarkan citra atau data sensor. Implementasi ini akan membantu petani dalam melakukan diagnosis dini terhadap kondisi tanaman dan mengambil tindakan yang tepat.
3. Untuk meningkatkan kemandirian dan efisiensi energi, disarankan agar sistem dilengkapi dengan manajemen energi berbasis panel surya. Pemanfaatan energi terbarukan ini akan mendukung pengoperasian sistem di lahan terbuka tanpa ketergantungan pada sumber listrik eksternal.

DAFTAR PUSTAKA

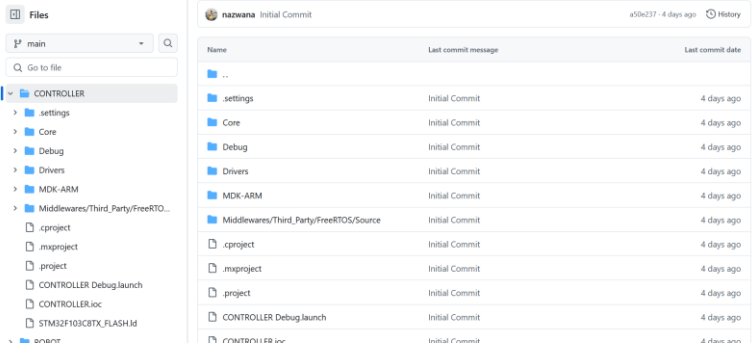

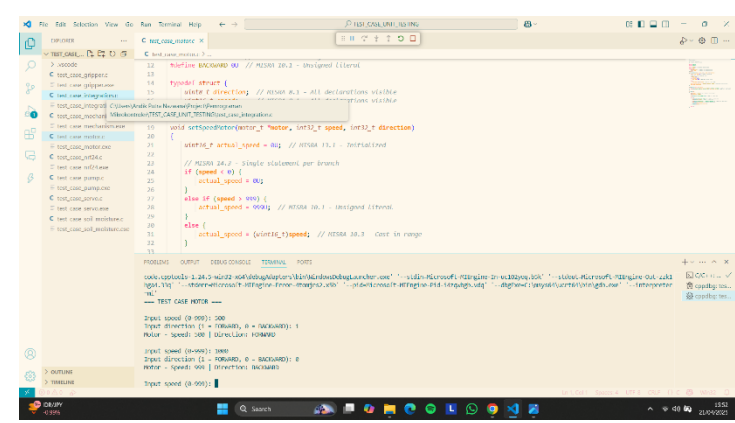
- 2016 IEEE 4th International Symposium on Robotics and Intelligent Sensors : “Empowering Robots with Smart Sensors” : 17-20 December 2016, Tokyo, Japan.* (2016). IEEE.
- al Rakib, M. A., Rahman, Md. M., Uddin, S., Khan, Md. A. H., Rahman, Md. A., Hossain, Md. M., Samad, M., & Abbas, F. I. (2022). Smart Agriculture Robot Controlling using Bluetooth. *European Journal of Engineering and Technology Research*, 7(6), 77–81. <https://doi.org/10.24018/ejeng.2022.7.6.2867>
- Al-Mashhadani, Z., & Park, J.-H. (2023). *Autonomous Agricultural Monitoring Robot for Efficient Smart Farming*.
- Auat Cheein, F. A., & Carelli, R. (2013). Agricultural robotics: Unmanned robotic service units in agricultural tasks. *IEEE Industrial Electronics Magazine*, 7(3), 48–58. <https://doi.org/10.1109/MIE.2013.2252957>
- Badan Pusat Statistik (BPS). (2023). Data Sensus Pertanian. Badan Pusat Statistik (BPS). <https://sensus.bps.go.id/st2023/>
- Cohen, A., & Zarrouk, D. (2023). Design, Analysis and Experiments of a High-Speed Water Hovering Amphibious Robot: AmphiSTAR. *IEEE Access*, 11, 80874–80885. <https://doi.org/10.1109/ACCESS.2023.3299498>
- Dutta, A., Roy, S., Patrick Kreidl, O., & Boloni, L. (2021). Multi-Robot Information Gathering for Precision Agriculture: Current State, Scope, and Challenges. *IEEE Access*, 9, 161416–161430. <https://doi.org/10.1109/ACCESS.2021.3130900>
- Fadhaeel, T., Patel, H. C., al Ahdal, A., Rakhra, M., & Singh, D. (2022). Design and development an Agriculture robot for Seed sowing, Water spray and Fertigation. *Proceedings of International Conference on Computational Intelligence and Sustainable Engineering Solution, CISES 2022*, 148–153. <https://doi.org/10.1109/CISES54857.2022.9844341>
- Febrianti, V. P., Permata, T. A., Humairoh, M., Putri, O. M., Amelia, L., Fatimah, S., & Khastini, R. O. (2021). Analisis Pengaruh Perkembangan Teknologi Pertanian Di Era Revolusi Industri 4.0 Terhadap Hasil Produksi Padi. *Jurnal Pengolahan Pangan*, 6(2), 54–60. <https://doi.org/10.31970/pangan.v6i2.50>
- Food and Agriculture Organization of the United Nations. (2022). State of Food Security and Nutrition in the World 2022. <https://openknowledge.fao.org/server/api/core/bitstreams/1c38676f-f5f7-47cf-81b3-f4c9794eba8a/content>
- Food and Agriculture Organization of the United Nations. (2023). Revealing the True Cost of Food to Transform Agrifood Systems.

- Ghafar, A. S. A., Hajjaj, S. S. H., Gsangaya, K. R., Sultan, M. T. H., Mail, M. F., & Hua, L. S. (2021). Design and development of a robot for spraying fertilizers and pesticides for agriculture. *Materials Today: Proceedings*, 81(2), 242–248. <https://doi.org/10.1016/j.matpr.2021.03.174>
- Gollakota, A., & Srinivas, M. B. (n.d.). *AGRIBOT-A MULTIPURPOSE AGRICULTURAL ROBOT*.
- Huang, C. C., & Chang, C. L. (2019). Design and Implementation of Bio-inspired Snake Bone-Armed Robot for Agricultural Irrigation Application. *IFAC-PapersOnLine*, 52(30), 98–101. <https://doi.org/10.1016/j.ifacol.2019.12.504>
- ICVEE : 2020 the third International Conference on Vocational Education and Electrical Engineering (ICVEE) : strengthening the framework of Society 5.0 through Innovations in Education, Electrical, Engineering and Informatics Engineering : proceeding : 3-4 October 2020, Universitas Negeri Surabaya, Vi[r]tual Event, Surabaya-Indonesia.* (2020). IEEE.
- Jo, Y., Park, Y., & Son, H. il. (2024). A suction cup-based soft robotic gripper for cucumber harvesting: Design and validation. *Biosystems Engineering*, 238, 143–156. <https://doi.org/10.1016/j.biosystemseng.2024.01.008>
- Li, S., Sun, W., Liang, Q. K., Liu, C. P., & Liu, J. (2024). Assessing fruit hardness in robot hands using electric gripper actuators with tactile sensors. *Sensors and Actuators A: Physical*, 365. <https://doi.org/10.1016/j.sna.2023.114843>
- Luo, L., Wei, Z., Sun, Y., Zhang, W., & Zhou, J. (2024). Preliminary Design and Control of an Ankle Rehabilitation Robot with Variable Stiffness Actuator. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2024.3517746>
- Noda, S., Kogoshi, M., & Iijima, W. (2025). Robot Simulation on Agri-field Point Cloud with Centimeter Resolution. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2025.3530967>
- Pak, J., Kim, J., Park, Y., & Son, H. il. (2022). Field Evaluation of Path-Planning Algorithms for Autonomous Mobile Robot in Smart Farms. *IEEE Access*, 10, 60253–60266. <https://doi.org/10.1109/ACCESS.2022.3181131>
- Pekaslan, D., Wagner, C., & Garibaldi, J. M. (2020). ADONiS-Adaptive Online Nonsingleton Fuzzy Logic Systems. *IEEE Transactions on Fuzzy Systems*, 28(10), 2302–2312. <https://doi.org/10.1109/TFUZZ.2019.2933787>
- Proceedings of IEEE CONECCT 2020 : 6th International Conference on Electronics, Computing and Communication Technologies : July 2-4, 2020.* (2020). IEEE.

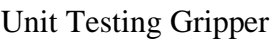
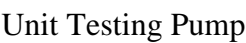
- Raikov, A., & Abrosimov, V. (2022). Artificial Intelligence and Robots in Agriculture. *Proceedings of 2022 15th International Conference Management of Large-Scale System Development, MLSD 2022*. <https://doi.org/10.1109/MLSD55143.2022.9934170>
- Rehman, A. U., Alamoudi, Y., Khalid, H. M., Morchid, A., Muyeen, S. M., & Abdelaziz, A. Y. (2024a). Smart agriculture technology: An integrated framework of renewable energy resources, IoT-based energy management, and precision robotics. *Cleaner Energy Systems*, 9. <https://doi.org/10.1016/j.cles.2024.100132>
- Rehman, A. U., Alamoudi, Y., Khalid, H. M., Morchid, A., Muyeen, S. M., & Abdelaziz, A. Y. (2024b). Smart agriculture technology: An integrated framework of renewable energy resources, IoT-based energy management, and precision robotics. *Cleaner Energy Systems*, 9. <https://doi.org/10.1016/j.cles.2024.100132>
- Rodríguez-Nieto, D., Navas, E., & Fernández, R. (2025). Automated Seeding in Hydroponic Greenhouse with a Dual-Arm Robotic System. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2025.3541954>
- Rosero-Montalvo, P. D., Gordillo-Gordillo, C. A., & Hernandez, W. (2023). Smart Farming Robot for Detecting Environmental Conditions in a Greenhouse. *IEEE Access*, 11, 57843–57853. <https://doi.org/10.1109/ACCESS.2023.3283986>
- Shamshiri, R. R., Navas, E., Dworak, V., Auat Cheein, F. A., & Weltzien, C. (2024). A modular sensing system with CANBUS communication for assisted navigation of an agricultural mobile robot. *Computers and Electronics in Agriculture*, 223. <https://doi.org/10.1016/j.compag.2024.109112>
- TIAR 2015 : 2015 IEEE International Conference on Technological Innovations in ICT for Agriculture and Rural Development : proceedings : 10th - 12th July 2015, Chennai, Tamil Nadu, India*. (2015). IEEE.
- Udekwe, D., & Seyyedhasani, H. (2025). Human robot interaction for agricultural Tele-Operation, using virtual Reality: A feasibility study. *Computers and Electronics in Agriculture*, 228. <https://doi.org/10.1016/j.compag.2024.109702>
- van Henten, E. J., Tabb, A., Billingsley, J., Popovic, M., Deng, M., & Reid, J. (2022). Agricultural Robotics and Automation [TC Spotlight]. *IEEE Robotics and Automation Magazine*, 29(4), 145–147. <https://doi.org/10.1109/MRA.2022.3213136>
- Visentin, F., Cremasco, S., Sozzi, M., Signorini, L., Signorini, M., Marinello, F., & Muradore, R. (2023). A mixed-autonomous robotic platform for intra-row and inter-row weed removal for precision agriculture. *Computers and Electronics in Agriculture*, 214. <https://doi.org/10.1016/j.compag.2023.108270>

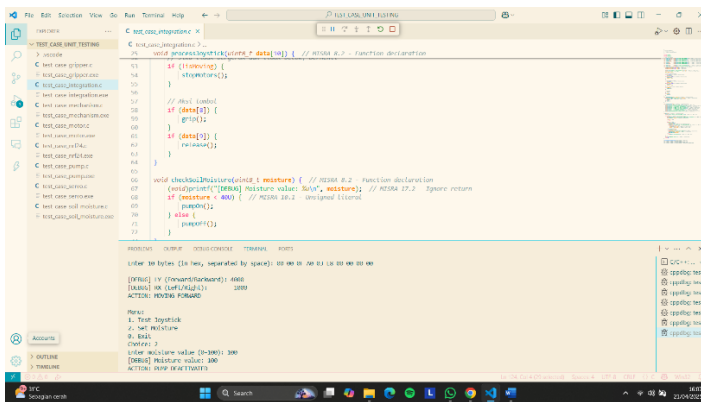
- Wei, Y., Li, W., An, D., Li, D., Jiao, Y., & Wei, Q. (2019). Equipment and Intelligent Control System in Aquaponics: A Review. In *IEEE Access* (Vol. 7, pp. 169306–169326). Institute of Electrical and Electronics Engineers Inc.
<https://doi.org/10.1109/ACCESS.2019.2953491>
- Yang, Q., Du, X., Wang, Z., Meng, Z., Ma, Z., & Zhang, Q. (2023). A review of core agricultural robot technologies for crop productions. *Computers and Electronics in Agriculture*, 206.
<https://doi.org/10.1016/j.compag.2023.107701>

LAMPIRAN

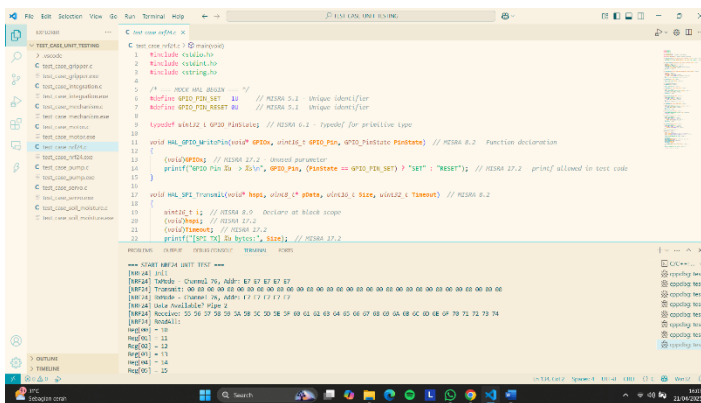
Dokumentasi	Keterangan
	<p>GitHub</p> <p>https://github.com/nazwana/AgroBot</p>
	<p>Materi Presentasi Tugas Proyek Pemrograman Kontroler AgroBot</p> <p>https://its.id/m/Kelompok6_PPTProjectPemKon</p>
	<p>Unit Testing Gerak Motor</p>







Unit Testing Integration



Unit Testing TX dan RX

<https://its.id/m/Kelompok6VideoPemKon>

Dokumentasi Video Pemrograman Kontroler