



TEKNIK MEMBINA ATUR CARA DENGAN BAHASA C

DAYANG NORHAYATI ABANG JAWAWI
ROSBI MAMAT



Bab 9: Membina Perisian Robot

© Copyright Universiti Teknologi Malaysia

innovative • entrepreneurial • global

9.1 Pengenalan

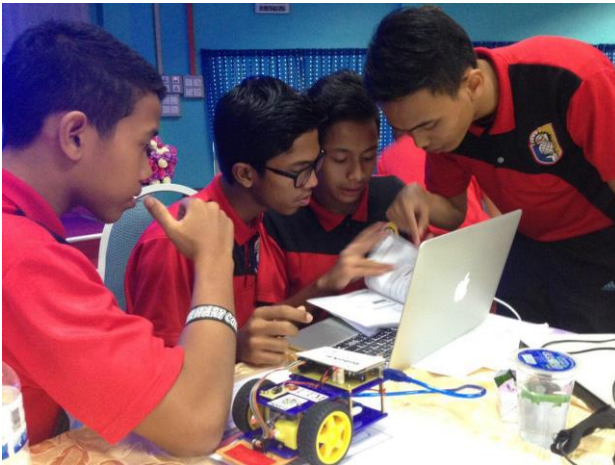
- fokus bab ialah kepada pembangunan perisian terbenam untuk sebuah robot berautonomi yang dipanggil **RoboKar**.
- Tujuan utama bab ini ialah untuk menunjukkan contoh pembangunan perisian terbenam yang ada perbezaannya dengan perisian sistem maklumat.
- **Robot** adalah sistem elektro-mekanikal yang boleh diatur cara untuk melakukan sesuatu tugas secara automatik.
- **RoboKar** ialah sebuah robot kecil yang boleh dipandu secara automatik untuk mengikuti jejak yang disediakan dengan bantuan penderia.

© Copyright Universiti Teknologi Malaysia

9.1 Pengenalan

- RoboKar telah dibangunkan di Universiti Teknologi Malaysia (UTM) untuk tujuan pengajaran topik pengaturcaraan perisian komputer melalui penyelesaian masalah RoboKar.
- Contoh penggunaan RoboKar untuk sesi pembelajaran (a) bengkel penyelesaian masalah pelajar sekolah dan (b) kerja kursus Kejuruteraan Perisian Masa Nyata pelajar UTM :

a)



b)



innovative • entrepreneurial • global

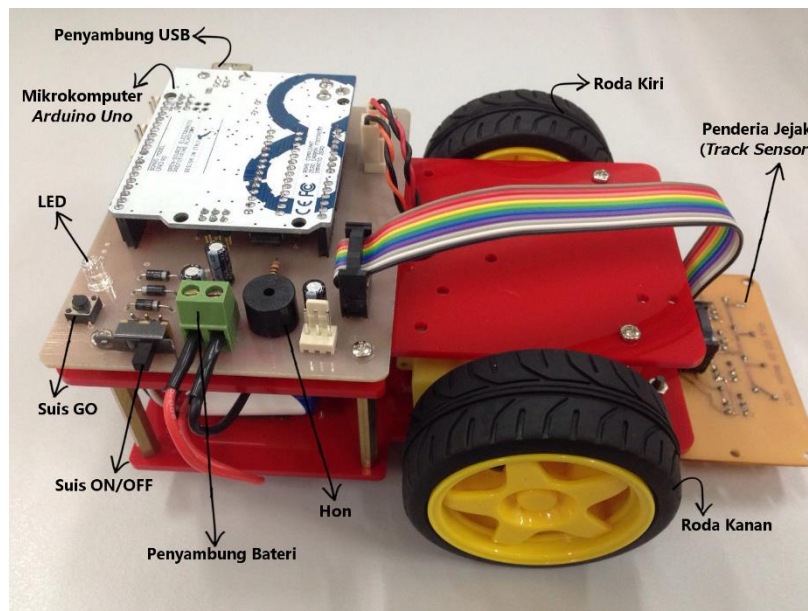
9.2 Binaan Asas RoboKar

Penyambung USB untuk pengaturcaraan – digunakan untuk menghantar program yang ditulis di PC ke mikrokomputer.

Mikrokomputer – tempat atur cara akan dilaksanakan.

LED - peranti semi konduktor yang memancarkan cahaya apabila arus elektrik melaluinya.

Roda dan pemacu motor - untuk mengawal kelajuan RoboKar dengan menukar kelajuan motor kiri dan motor kanan.

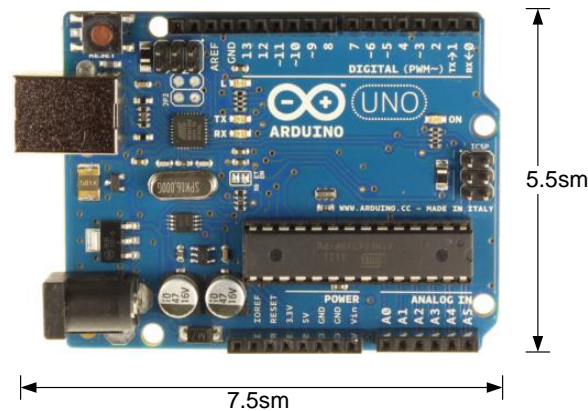


Penderia jejak - untuk mengesan garisan hitam di litar.

(3)Bateri - memperuntukkan kuasa kepada RoboKar.

9.2.1 Mikrokomputer dan Perantaramuka Elektronik

- Mikrokomputer yang menjadi otak kepada RoboKar adalah mikrokomputer kecil Arduino Uno :



- Pengantara muka elektronik di antara Arduino dengan penderia dan penggerak dibina pada lapisan atas badan RoboKar

9.2.2 Penderia dan Penggerak

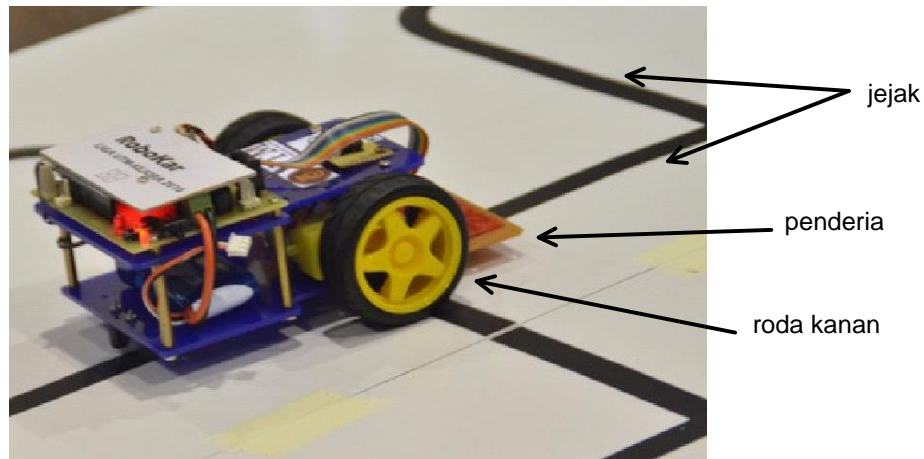
- RoboKar dilengkapi dengan penderia optik. Rajah menunjukkan empat penderia optik, RoboKar juga boleh dikawal dengan hanya dua penderia optik.



- Penderia optik digunakan untuk mengesan jejak hitam di atas lantai dan memberi maklumat kedudukan haluan robot relatif kepada jejak hitam tersebut.
- Penderia yang dipasang di hadapan RoboKar ini digunakan untuk mengemudi robot di atas jejak.

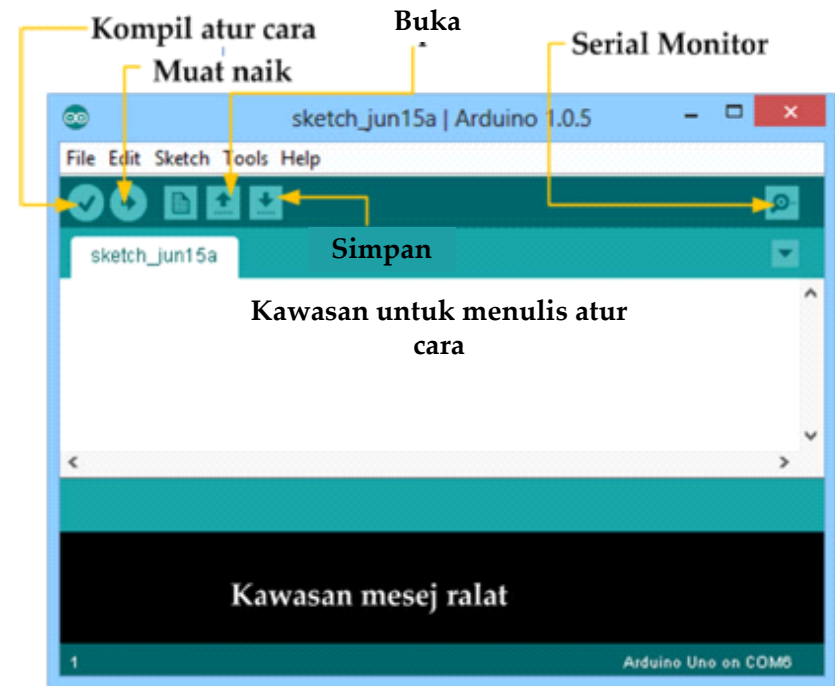
9.2.2 Penderia dan Penggerak

- RoboKar boleh digerakkan oleh dua motor kecil sebagai penggerak yang disambung kepada roda kanan dan kiri.
- Contoh :



9.3 Alat Pembangunan Perisian

- Persekitaran pengaturcaraan bersepadu (IDE) dan bahasa pengaturcaraan C digunakan sebagai alat pembangunan RoboKar.
- Mikrokomputer Arduino di atas RoboKar boleh diatur cara menggunakan **IDE Arduino**
- Atur cara ditulis menggunakan IDE dan dimuat naik ke dalam ingatan kekal Arduino melalui kabel USB. Contoh antara muka IDE Arduino :



9.3.1 Perpustakaan RoboKar

- Bahasa pengaturcaraan C yang disokong oleh IDE Arduino digunakan untuk pengaturcaraan RoboKar.

- C dipilih kerana:
 1. algoritma kawalan robot lebih sesuai untuk diungkapkan dengan bahasa pengaturcaraan
 2. kemahiran pelajar menggunakan bahasa pengaturcaraan adalah salah satu objektif utama;
 3. bahasa pengaturcaraan C tidak terhad kepada robot tertentu sahaja, kemahiran dalam bahasa pengaturcaraan C berguna bagi tujuan lain untuk pelajar semasa mereka berada di kolej matrikulasi dan universiti nanti.

9.3.1 Perpustakaan RoboKar

- Senarai arahan yang telah disediakan sebagai modul pada perpustakaan untuk mengawal RoboKar adalah :

Fungsi	Tujuan
<code>roboSteerRight(percent, rqspeed);</code>	Pusingkan stereng sebanyak darjah (0-90) dengan laju yang diberikan oleh nilai speed (0-100). Guna untuk belok ke kanan. Syor untuk nilai stereng: 50 – 90.
<code>roboSteerLeft(percent, rqspeed);</code>	Pusingkan stereng sebanyak darjah (0-90) dgn laju yang diberikan oleh nilai speed (0-100). Guna utk belok ke kiri. Syor untuk nilai stereng: 50 – 90.
<code>roboSteerStraight(speed);</code>	Memandu RoboKar bergerak lurus dengan laju yang diberikan oleh nilai speed (0-100). Disyorkan untuk guna nilai speed berikut: STOP, SLOW, MEDIUM, FAST. Contoh: <code>roboSteerStraight(STOP);</code> //berhentikan RoboKar <code>roboSteerStraight(FAST);</code> //RoboKar gerak dgn laju
<code>roboReverse(speed);</code>	Undur RoboKar.
<code>roboStop();</code>	Berhentikan RoboKar
<code>roboHonk();</code>	Membunyikan hon RoboKar sekali.
<code>roboCheckBattery();</code>	Untuk menguji kandungan cas bateri, jika lemah nyalakan LED 'battery low' atas RoboKar. Jika kuat padamkan LED tersebut.

9.3.1 Perpustakaan RoboKar

- Senarai arahan yang telah disediakan sebagai modul pada perpustakaan untuk mengawal RoboKar adalah (Sambungan):

Fungsi	Tujuan
<code>keyPressed();</code>	Menguji samada suis GO ditekan atau tidak. Contoh penggunaan fungsi: <code>char key = keyPressed();</code> Jika ditekan, nilai '1' akan disimpan dalam pemboleh ubah key dan jika tidak nilai '0' pula akan disimpan.
<code>wait4keyPress();</code>	Tunggu sehingga suis GO ditekan. Disyorkan untuk diguna pada permulaan atur cara dalam fungsi <code>loop()</code> sebelum arahan lain dilaksanakan.
<code>readLineSensor();</code>	Untuk membaca penderia jejak dan memulangkan satu nombor (0-15). Contoh penggunaan fungsi: <code>int sensor = readLineSensor();</code> Nombor yang dipulangkan dari fungsi diumpuk ke dalam pemboleh ubah sensor yang memberikan kedudukan RoboKar di atas jejak. Nilai dalam sensor boleh digunakan untuk mengemudi RoboKar.
<code>waitSeconds (secs);</code>	Lengahkan selama masa saat.

9.3.1 Perpustakaan RoboKar

- Contoh keratan atur cara untuk kawalan robot yang menggunakan set fungsi RoboKar :

```
sensor = readLineSensor(); // baca penderia jejak
if (sensor == 6) // RoboKar di tengah
    roboSteerStraight(MEDIUM); // gerak lurus
else if (sensor == 1) // Hampir keluar ke kiri
    robotSteerRight(90, MEDIUM); // belok kanan
else if (sensor == 8) // Hampir keluar ke kanan
    robotSteerLeft(90, MEDIUM); // belok kiri
```

9.4 Pembangunan Perisian bagi RoboKar

- Satu perisian akan dibangunkan di seksyen ini bagi mengemudi RoboKar bergerak ke hadapan sambil mengikuti jejak hitam di atas permukaan putih.
- Perbezaan warna di antara jejak dan permukaan ini penting kerana penderia optik pada RoboKar hanya boleh membezakan warna hitam atau putih.

9.4.1 Spesifikasi Keperluan Perisian RoboKar

- Pertimbangkan satu RoboKar dengan dua penderia optik sahaja.
- Perisian pengemudi RoboKar yang hendak dibangunkan mestilah memenuhi keperluan yang diberikan di dalam Jadual :

Nombor	Tujuan
1	Mengasalkan semua perkakasan (penderia dan penggerak) RoboKar.
2	Menunggu suis 'GO' di tekan sebelum memulakan perjalanan/pengemudian RoboKar. Bunyikan hon sekali untuk mengisytiharkan kepada pengguna yang RoboKar sedang menunggu suis ditekan.
3	Menggunakan dua penderia optik untuk mengesan kedudukan RoboKar semasa mengemudi mengikut jejak hitam.
4	Jika RoboKar terkeluar dari jejak, berhenti dan kelipkan LED 3 kali dan seterusnya bergerak/belok ke kiri dengan sudut stering 90 darjah dan kelajuan sederhana.

9.4.2 Analisis Keperluan dan Reka bentuk Perisian RoboKar

- Robot yang boleh mengesan, merancang dan bekerja tanpa bantuan dari manusia di panggil robot berautonomi.

- Salah satu kaedah yang digunakan dalam atur cara robot sebegini ialah kaedah **kesan-rancang-laku**.
 1. Di dalam proses **kesan**, robot menggunakan penderia untuk mengesan persekitarannya.
 2. Maklumat dari penderia ini digunakan untuk merancang pergerakan yang diperlukan di dalam proses **rancang**.
 3. di dalam proses **laku**, isyarat kawalan dihantar kepada penggerak untuk melakukan pergerakan yang telah dirancang.

9.4.2 Analisis Keperluan dan Reka bentuk Perisian RoboKar

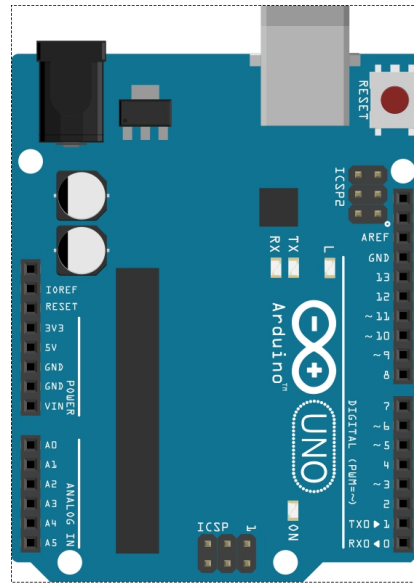
- Input dan Output yang ada di atas RoboKar :

INPUT / PENDERIA

Suis 'Go'



Penderia Optik

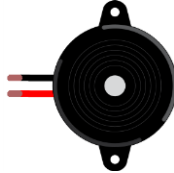



OUTPUT / PENGGERAK

LED



Hon



Motor/Roda Kiri-Kanan



9.4.2 Analisis Keperluan dan Reka bentuk Perisian RoboKar

- *Analisis dan reka bentuk bagi keperluan 1*

Semua peranti perkakasan RoboKar boleh diasalkan dengan memanggil fungsi `roboSetup()` di dalam fungsi `setup()`. Fungsi ini adalah daripada fungsi perpustakaan RoboKar, oleh itu kenyataan `#include "robokarlibs.h"` perlu dimasukkan pada fail punca atur cara RoboKar seperti keratan atur cara berikut:

```
#include "robokarlibs.h"

void setup() {
    roboSetup();    // asalkan perkakakasan
RoboKar
}
```

9.4.2 Analisis Keperluan dan Reka bentuk Perisian RoboKar

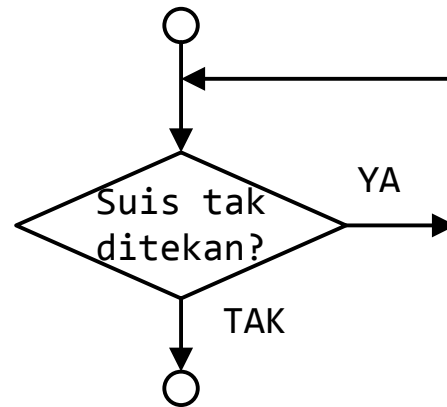
- *Analisis dan reka bentuk bagi keperluan 2*

Utk penuhi keperluan ini, peranti input suis 'Go' perlu dibaca dan diuji smada ia ditekan atau tidak. Fungsi `keyPressed()` daripada fungsi perpustakaan RoboKar boleh digunakan untuk baca keadaan suis 'Go' ini. Fungsi `keyPressed()` akan memulangkan nilai 1 jika suis ditekan atau nilai 0 jika tidak ditekan. Penggunaan fungsi `keyPressed()` digunakan untuk menghalusi kenyataan Suis tak ditekan di dalam simbol keputusan.

9.4.2 Analisis Keperluan dan Reka bentuk Perisian RoboKar

- *Analisis dan reka bentuk bagi keperluan 2*

Carta alir menguji suis :



Carta alir ini boleh diterjemahkan kepada gelung untuk menguji suis 'Go' seperti keratan atur cara berikut:

```
while (keyPressed() == 0) // selagi suis tak ditekan  
; // ulang dan uji
```

9.4.2 Analisis Keperluan dan Reka bentuk Perisian RoboKar

- *Analisis dan reka bentuk bagi keperluan 3*

Kedudukan RoboKar relatif kepada jejak yang diikutinya diketahui menggunakan penderia optik yang boleh mengesan jejak yang berwarna hitam dan bukan jejak yang berwarna putih. Penderia optik ini dibaca menggunakan fungsi `readLineSensor()`

Nilai yang dipulangkan oleh fungsi ini bergantung kepada kedudukan RoboKar relatif kepada jejak seperti yang diberikan di dalam Jadual 9.3. Kawasan berlorek menunjukkan jejak.

9.4.2 Analisis Keperluan dan Reka bentuk Perisian RoboKar

- *Analisis dan reka bentuk bagi keperluan 3*



Jadual 9.3 :

KESAN		RANCANG	
Kedudukan Penderia dan Jejak	Nilai dari readLineSensor()	Kedudukan RoboKar	Pengemudian RoboKar
	3	Di tengah jejak. Boleh kekalkan haluan sekarang dengan kelajuan tinggi.	roboSteerStraight(FAST)
	2	Di kiri jejak. Belok ke kanan dengan kelajuan sederhana.	roboSteerRight(60, MEDIUM)

9.4.2 Analisis Keperluan dan Reka bentuk Perisian RoboKar

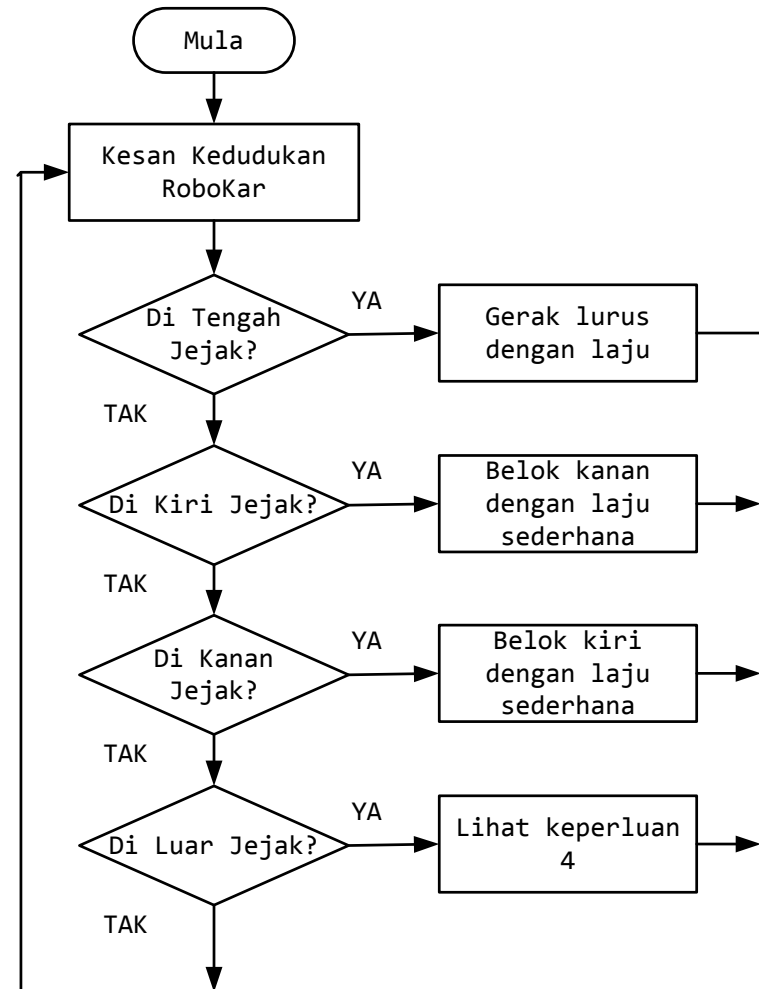
- *Analisis dan reka bentuk bagi keperluan 3*

Jadual 9.3
(Sambungan) :

KESAN		RANCANG		LAKU	
Kedudukan Penderia dan Jejak	Nilai dari readLineSensor()	Kedudukan RoboKar	Pengemudian RoboKar		
	1	Di kanan jejak. Belok ke kiri dengan kelajuan sederhana.	roboSteerLeft(60, MEDIUM)		
	0	Terkeluar dari jejak. Perlu kemudikan RoboKar untuk masuk semula ke dalam jejak.	Lihat analisis keperluan 4.		

9.4.2 Analisis Keperluan dan Reka bentuk Perisian RoboKar

- *Analisis dan reka bentuk bagi keperluan 3*
- carta alir berasaskan Jadual 9.3 :
- Gelung pelaksanaan ini boleh dicapai dengan meletakkan semua arahan pengemudian ini di dalam fungsi loop() Arduino.



9.4.2 Analisis Keperluan dan Reka bentuk Perisian RoboKar

- *Analisis dan reka bentuk bagi keperluan 4*
- Dari keperluan 4, tindakan berikut perlu dilakukan apabila RoboKar terkeluar dari jejak:
 1. Berhentikan RoboKar
 2. Kelipkan LED tiga kali
 3. Belok ke kiri dengan sudut 90 darjah.
- Untuk berhentikan RoboKar, fungsi roboStop() daripada fungsi perpustakaan RoboKar boleh dipanggil.

9.4.2 Analisis Keperluan dan Reka bentuk Perisian RoboKar

- *Analisis dan reka bentuk bagi keperluan 4*
- untuk kelipkan LED sebanyak tiga kali, keratan atur cara berikut boleh dilakukan:

```
for (int i = 0; i < 3; i++) {           // 3 kelipan
    digitalWrite(8, HIGH); delay(500); // nyala & tunggu
    digitalWrite(8, LOW);  delay(500); // padam & tunggu
}
```

9.4.3 Pelaksanaan Atur Cara Pengemudian RoboKar

- Contoh atur cara pengemudian RoboKar Dengan Tiga Pendeia Optik :

```
1 #include "robokarlibs.h"
2 void setup() {
3   roboSetup(); // asalkan perkakakasan RoboKar
4   roboHonk(); // bunyikan hon
5   while (keyPressed() == 0) // selagi suis tak ditekan
6     ; // ulang dan uji
7 }
8 void loop()
9 {
10  int sensor;
11  int i;
12
13  sensor = readLineSensor();
14  if (sensor == 3) { // di tengah jejak?
15    roboSteerStraight(FAST); // gerak lurus dgn laju
16  }
17  else if (sensor == 2) { // di kiri jejak?
18    roboSteerRight(60, MEDIUM); // belok ke kanan
19  }
```

9.4.3 Pelaksanaan Atur Cara Pengemudian RoboKar

- Contoh atur cara pengemudian RoboKar Dengan Tiga Pendeia Optik (Sambungan):

```
20 else if (sensor == 1) {           // di kanan jejak?
21     roboSteerLeft(60, MEDIUM); // belok ke kiri
22 }
23 else if (sensor == 0) {           // keluar dari jejak?
24     roboStop();                   // berhentikan RoboKar
25     for (i = 0; i < 3; i++) {     // 3 kelipan
26         digitalWrite(8, HIGH); delay(500); // nyala & tunggu
27         digitalWrite(8, LOW); delay(500); // padam & tunggu
28         roboSteerLeft(90, SLOW);    // belok ke kiri
29     }
30 }
31
```

9.4.4 Penentusahan dan Pengesahihan RoboKar

- Penilaian dokumen dan artifek perisian yang dihasilkan pada Seksyen 9.4.1 hingga 9.4.3 dinamakan fasa penentusahan.
- Semakan adalah salah satu teknik yang biasa digunakan di fasa penentusahan.

9.4.4 Penentusahan dan Pengesahihan RoboKar

- contoh senarai semakan berikut boleh digunakan untuk proses semakan dokumen keperluan :
 1. Adakah keperluan menyatakan input dan output yang sesuai?
 2. Adakah persekitaran perkakasan dan perisian ditetapkan?
 3. Adakah keperluan antara muka robot telah dinyatakan dengan betul?
 4. Adakah keperluan bebas daripada pertindihan dan tidak bercanggah dengan keperluan lain?
 5. Adakah setiap keperluan ditulis dengan konsisten, jelas, bahasa yang tepat?
 6. Adakah setiap keperluan hanya mempunyai satu makna dan tidak boleh ditakrifkan dengan pelbagai makna?
 7. Adakah setiap keperluan memenuhi skop projek?

9.4.4 Penentusahan dan Pengesahihan RoboKar

- contoh semakan yang boleh dibuat pada Atur cara 9.1:
 1. Adakah atur cara telah melaksanakan reka bentuk dengan lengkap dan betul?
 2. Adakah kod telah didokumenkan dengan lengkap dan betul untuk memudahkan menyenggarakan kod?
 3. Adakah nama pemboleh ubah ditakrifkan dengan nama yang bermakna dan jelas?
 4. Adakah terdapat pemboleh ubah yang tidak digunakan?
 5. Adakah semua kawalan gelung dan pilihan lengkap dan betul?
 6. Adakah semua pemboleh ubah output telah diumpukkan atau digunakan?

9.4.4 Penentusahan dan Pengesahihan RoboKar

- Pengujian perisian adalah satu cara pengesahsahihan atur cara, salah satu teknik pengujian yang digunakan adalah pengujian berasaskan keperluan.
- Teknik ini mereka bentuk kes ujian dengan mempertimbangkan setiap keperluan dan membangunkan satu set kes ujian daripada keperluan tersebut.
- Set kes ujian dibangunkan untuk menguji sesuatu keperluan dan satu set kajian terdiri daripada satu set nilai-nilai input, jangkauan hasil dan hasil sebenar.

9.4.4 Penentusahan dan Pengesahihan RoboKar

- contohnya sebenar daripada pemerhatian satu kumpulan pelajar sekolah yang telah mereka bentuk dan mengkod atur cara mereka di bengkel RoboKar :

No kes	Nilai input readLineSensor()	Jangkaan pergerakan	Pergerakan sebenar
K3KU01	3	Boleh kekalkan haluan sekarang dengan kelajuan tinggi	Boleh kekalkan haluan sekarang dengan kelajuan tinggi
K3KU02	2	Belok ke kanan dengan kelajuan sederhana.	Dikemudi terlebih ke kanan
K3KU03	1	Belok ke kiri dengan kelajuan sederhana.	Dikemudi terlalu sedikit ke kiri

9.5 Penyenggaraan Perisian RoboKar

- Bahagian ini akan membincangkan dua keadaan yang memerlukan penyenggaraan dilakukan ke atas atur cara pengemudian RoboKar iaitu:
 1. bila struktur atur cara perlu diubah untuk memudahkan penalaan data atau parameter
 2. bila keperluan 4 diubah untuk memberi lebih kepintaran kepada RoboKar semasa menangani kes terkeluar dari jejak.

- Difasa penyenggaraan kesemua aktiviti fasa keperluan, analisa, reka bentuk, perlaksanaan dan pengujian juga perlu dilakukan.

9.5.1 Penyenggaraan Perisian untuk Memudahkan Penalaan

- penalaan - mendapatkan nilai sudut stereng dan halaju yang paling sesuai dengan pengemudian yang dilakukan.
- Untuk memudahkan proses penalaan dan pengubahan nilai sudut stereng dan halaju, arahan yang mengandungi nilai sudut stereng dan halaju perlu disetempatkan.

9.5.1 Penyenggaraan Perisian untuk Memudahkan Penalaan

- Keratan atur cara berikut membina jadual carian bagi sudut stereng dan halaju RoboKar.

```
int sudutStereng[] = { 0,          //sensor = 0, kes keluar jejak
                      60,         //sensor = 1, belok kanan
                      -60,        //sensor = 2, belok kiri
                      0           //sensor = 3, lurus
                    };
int halaju[] = { SLOW,           //sensor = 0, kes keluar jejak
                MEDIUM,        //sensor = 1
                MEDIUM,        //sensor = 2
                FAST             //sensor = 3
              };
```

9.5.1 Penyenggaraan Perisian untuk Memudahkan Penalaan

- Contoh atur cara pengemudian RoboKar dengan tatasusunan :

```
1 #include "robokarlibs.h"
2 int sudutStereng[] = { 0, 60, -60, 0};
3 int halaju[] = {SLOW, MEDIUM, MEDIUM, FAST};
4
5 void setup() {
6     roboSetup(); // asalkan perkakakasan RoboKar
7     roboHonk(); // bunyikan hon
8     while (keyPressed() == 0) // selagi suis tak ditekan
9         ; // ulang dan uji
10 }
11 void loop()
12 {
13     int sensor; // nilai penderia optik
14                // & indeks kpd jadual carian
15     int sudut; // sudut stereng
16     int laju; // halaju
17     int i;
18
19     sensor = readLineSensor();
20     sudut = sudutStereng[sensor]; // capai sudut sepadan
```

9.5.1 Penyenggaraan Perisian untuk Memudahkan Penalaan

- Contoh atur cara pengemudian RoboKar dengan tatasusunan :

```
21 laju = halaju[sensor];           // capai laju sepadan
22
23 if (sensor == 0) {               // keluar dari jejak?
24     roboStop();                  // berhentikan RoboKar
25     for (i = 0; i < 3; i++) {    // 3 kelipan
26         digitalWrite(8, HIGH); delay(500); // nyala & tunggu
27         digitalWrite(8, LOW);  delay(500); // padam & tunggu
28         roboSteerLeft(90, SLOW); // belok ke kiri
29     }
30 }
31 else {
32     if (sudut < 0) {              // belok kiri?
33         sudut = 0 - sudut;       // jadikan sudut positif
34         roboSteerLeft(sudut, laju); // belok ke kiri
35     }
36     else                          // belok kanan @ lurus
37         roboSteerRight(sudut, laju);
38 }
39 }
40
```

9.5.2 Penyenggaraan Perisian bila Keperluan Berubah

- Untuk memberi lebih kepintaran kepada RoboKar semasa menangani kes terkeluar dari jejak, keperluan 4 boleh diubah seperti berikut:
- Keperluan 4: *Jika RoboKar terkeluar dari jejak, berhenti dan kelipkan LED 3 kali dan tentukan arah pergerakan RoboKar sebelum ia terkeluar dari jejak. Lakukan pergerakan untuk memampas pergerakan sebelum ia terkeluar dari jejak. Misalnya, jika sebelum terkeluar jejak pergerakan RoboKar adalah belok kiri, maka pergerakan memampas ialah undur sedikit dan belok kanan.*

9.5.2 Penyenggaraan Perisian bila Keperluan Berubah

- Nilai sudut stereng, arah dan halaju RoboKar boleh diingat dengan mengumpulkan nilai-nilai ini di dalam satu pemboleh ubah melalui binaan struktur bahasa C.
- Keratan atur cara berikut adalah satu contoh bagaimana pemboleh ubah ini diisytiharkan:

```
struct datarobot
{
    int arah;           // ingat arah robot
    int sudutstereng;  // ingat sudut belokan
    int laju;          // ingat halaju robot
};
struct datarobot datalepas; // datalepas sebagai pemboleh ubah
```

9.5.2 Penyenggaraan Perisian bila Keperluan Berubah

- Nilai sudut stereng, arah dan halaju di dalam pemboleh ubah datalepas boleh dikemaskini dengan nilai sebenar menggunakan fungsi berikut:

```
void kemaskinidata (int arahbelok, int sudut, int halaju)
{
    datalepas.arah = arahbelok;
    datalepas.sudutstereng = sudut;
    datalepas.laju = halaju;
}
```


9.5.2 Penyenggaraan Perisian bila Keperluan Berubah

- Empat Kemungkinan bagi Arah RoboKar Sebelum Terkeluar Jejak serta Tindakan Memampas :

Arah Pergerakan Yang Mungkin Sebelum Terkeluar Jejak	Tindakan Memampas Yang Diperlukan
Lurus	Undur seketika, kemudian belok kanan dengan sudut stereng 60 darjah dan halaju perlahan.
Belok Kanan	Undur seketika, kemudian belok kiri dengan sudut stereng 60 darjah dan halaju sederhana.
Belok Kiri	Undur seketika, kemudian belok kanan dengan sudut stereng 60 darjah dan halaju sederhana.
Undur (ini hanya berlaku jika pergerakan undur sebelum ini tidak cukup untuk memasukan RoboKar ke dalam jejak semula)	Undur seketika, kemudian belok ke kiri dengan sudut stereng 90 darjah dan halaju sederhana.

9.5.2 Penyenggaraan Perisian bila Keperluan Berubah

- Atur cara pengemudian RoboKar selepas Keperluan 4 berubah :

```
1: #include "robokarlibs.h"
2: #define LURUS    0
3: #define KEKANAN 1
4: #define KEKIRI  2
5: #define UNDUR   3
6:
7: int sudutStereng[] = { 0, 60, -60, 0};
8: int halaju[] = {SLOW, MEDIUM, MEDIUM, FAST};
9: struct datarobot {
10:     int arah;           // ingat arah robot
11:     int sudutstereng;  // ingat sudut belokan
12:     int laju;          // ingat halaju robot
13: };
14: struct datarobot datalepas; // ingat keadaan RoboKar
15:
16: void kemaskinidata (int arahbelok, int sudut, int halaju)
17: {
18:     datalepas.arah = arahbelok;
```

9.5.2 Penyenggaraan Perisian bila Keperluan Berubah

- Atur cara pengemudian RoboKar selepas Keperluan 4 berubah (Sambungan):

```
19: datalepas.sudutstereng = sudut;
20: datalepas.laju = halaju;
21: }
22:
23: void setup() {
24:   roboSetup(); // asalkan perkakakan RoboKar
25:   roboHonk(); // bunyikan hon
26:   while (keyPressed() == 0) // selagi suis tak ditekan
27:     ; // ulang dan uji
28: }
29:
30: void loop()
31: {
32:   int sensor; // nilai penderia optik
33:   // & indeks kpd jadual caria
34:   int sudut; // sudut stereng
35:   int lajurobot; // halaju RoboKar
36:   int arahgerak; // arah pergerakan RoboKar
37:
```

9.5.2 Penyenggaraan Perisian bila Keperluan Berubah

- Atur cara pengemudian RoboKar selepas Keperluan 4 berubah (Sambungan):

```
38:   sensor = readLineSensor();
39:   sudut = sudutStereng[sensor]; // capai sudut sepadan
40:   lajurobot = halaju[sensor];   // capai laju sepadan
41:
42:   if (sensor == 0) {           // keluar dari jejak?
43:       roboStop();              // berhentikan RoboKar
44:       if (datalepas.arah == KEKANAN) {
45:           roboReverse(SLOW); waitSeconds(2); //undur seketika
46:           roboSteerLeft(60, MEDIUM);
47:           kemaskinidata(KEKIRI, 60, MEDIUM);
48:       }
49:       else if (datalepas.arah == KEKIRI) {
50:           roboReverse(SLOW); waitSeconds(2); //undur seketika
51:           roboSteerRight(60, MEDIUM);
52:           kemaskinidata(KEKANAN, 60, MEDIUM);
53:       }
54:       else if (datalepas.arah == LURUS) {
55:           roboReverse(SLOW); waitSeconds(2); //undur seketika
56:           roboSteerRight(60, SLOW);
```

9.5.2 Penyenggaraan Perisian bila Keperluan Berubah

- Atur cara pengemudian RoboKar selepas Keperluan 4 berubah (Sambungan):

```
57: kemaskinidata(KEKANAN, 60, SLOW);
58: }
59: else if (datalepas.arah == UNDUR) {
60:     roboReverse(SLOW); waitSeconds(2); //undur seketika
61:     roboSteerLeft(90, MEDIUM);
62:     kemaskinidata(KEKIRI, 90, MEDIUM);
63: }
64: }
65: else {
66:     if (sudut < 0) {           // belok kiri?
67:         sudut = 0 - sudut;    // jadikan sudut positif
68:         roboSteerLeft(sudut, lajurobot); // belok ke kiri
69:         kemaskinidata(KEKIRI, sudut, lajurobot);
70:     }
71:     else                       // belok kanan @ lurus
72:         roboSteerRight(sudut, lajurobot);
73:         kemaskinidata(KEKANAN, sudut, lajurobot);
74: }
75: }
```

© Copyright Universiti Teknologi Malaysia

innovative • entrepreneurial • global