Lembar Kegiatan Peserta Didik (LKPD)

**Mata Kuliah** : Bahasa Pemrograman Terstruktur  
**Semester** : 1  
**SKS** : 2

# Chapter 2 - Struktur dan Logika Bahasa Pemrograman

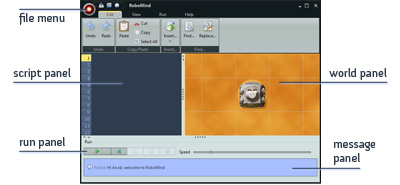
**Tujuan Praktikum:**

1. Mahasiswa memahami dan mampu menggunakan struktur sekuensial dalam pemrograman menggunakan Robomind
2. Mahasiswa memahami dan mampu menggunakan struktur kondisional dalam pemrograman menggunakan Robomind
3. Mahasiswa memahami dan mampu menggunakan struktur perulangan dalam pemrograman menggunakan Robomind

# Handout

Robomind merupakan perangkat lunak yang dapat digunakan untuk pembelajaran pemrograman komputer, bahkan juga pembelajaran robotika. Fokus utama dari perangkat lunak ini adalah bagaimana mengendalikan sebuah robot melalui serangkaian instruksi dalam bentuk kode program sehingga robot bisa menyelesaikan misi yang diinginkan.

Gambar berikut ini adalah tampilan interface perangkat lunak Robomind beserta penjelasan bagian-bagiannya.



Keterangan:

* **File menu** berisi submenu yang digunakan untuk pengaturan file, seperti membuat file baru, melakukan save atau save as file, dan juga membuka file
* **Script panel** merupakan area di mana kode program (kumpulan instruksi) dituliskan. Kode program yang ditulis inilah yang akan dijalankan untuk pengendalian robot
* **Run panel** berisi serangkaian tombol yang digunakan untuk mengeksekusi kode program yang telah dibuat, menghentikan proses eksekusi sementara (pause), atau menghentikan sepenuhnya, dan juga mereset eksekusi
* **World panel** merupakan area map yang akan dilalui oleh robot
* **Message panel** merupakan bagian dari interface yang nantinya akan memunculkan beberapa pesan seperti pesan kesalahan sintaks, atau pesan yang muncul dari robot dikarenakan ketidakmampuannya untuk menjalankan instruksi yang diberikan

## Perintah Dasar Robomind

Untuk mengatur gerakan robot, diperlukan beberapa instruksi yang diperhatikan. Tabel berikut ini adalah daftar perintah/instruksi dasar yang bisa diberikan kepada robot.

|  |  |
| --- | --- |
| **Perpindahan (move)** | |
| forward(n) | Menggerakkan robot untuk maju sebanyak n langkah (n kotak dalam map) |
| backward(n) | Menggerakkan robot untuk mundur sebanyak n langkah |
| left | Menggerakkan robot untuk hadap kiri relatif terhadap posisi depannya |
| right | Menggerakkan robot untuk hadap kanan relatif terhadap posisi depannya |
| west(n) | Menggerakkan robot ke arah barat (sesuai arah map) sebanyak n langkah |
| east(n) | Menggerakkan robot ke arah timur (sesuai arah map) sebanyak n langkah |
| north(n) | Menggerakkan robot ke arah utara (sesuai arah map) sebanyak n langkah |
| south(n) | Menggerakkan robot ke arah selatan (sesuai arah map) sebanyak n langkah |

|  |  |
| --- | --- |
| **Mengecat (paint)** | |
| paintWhite | Mengeluarkan kuas cat warna putih |
| paintBlack | Mengeluarkan kuas cat warna hitam |
| stopPainting | Memasukkan kembali kuas cat |

|  |  |
| --- | --- |
| **Gerakan lainnya** | |
| pickUp | Mengambil benda (beacon) |
| putDown | Meletakkan benda (beacon) yang sebelumnya diambil |
| flipCoin | Melakukan toast coin yang secara random akan menghasilkan nilai TRUE atau FALSE |

|  |  |
| --- | --- |
| **Pengecekan Kondisi (selalu menghasilkan nilai TRUE atau FALSE)** | |
| leftIsObstacle | Mengecek apakah sebelah kirinya ada halangan |
| leftIsClear | Mengecek apakah sebelah kirinya tidak ada halangan |
| leftIsBeacon | Mengecek apakah sebelah kirinya ada beacon |
| leftIsWhite | Mengecek apakah sebelah kirinya ada cat warna putih |
| leftIsBlack | Mengecek apakah sebelah kirinya ada cat warna hitam |
| frontIsObstacle | Mengecek apakah depannya ada halangan |
| frontIsClear | Mengecek apakah depannya tidak ada halangan |
| frontIsBeacon | Mengecek apakah depannya ada beacon |
| frontIsWhite | Mengecek apakah depannya ada cat warna putih |
| frontIsBlack | Mengecek apakah depannya ada cat warna hitam |
| rightIsObstacle | Mengecek apakah sebelah kanannya ada halangan |
| rightIsClear | Mengecek apakah sebelah kanannya tidak ada halangan |
| rightIsBeacon | Mengecek apakah sebelah kanannya ada beacon |
| rightIsWhite | Mengecek apakah sebelah kanannya ada cat warna putih |
| rightIsBlack | Mengecek apakah sebelah kanannya ada cat warna hitam |

# Praktikum 1

**Tujuan:**

Mahasiswa memahami konsep sekuensial proses dalam pemrograman.

**Langkah kerja:**

1. Buka aplikasi Robomind
2. Buka map *default.map* melalui menu **open map**
3. Tuliskan perintah-perintah program berikut ini di bagian script panel

forward(1)

forward(2)

east(3)

south(4)

pickUp

right

putDown

1. Kemudian simpan file program dengan nama **sekuensial01.irobo**
2. Kemudian jalankan program dengan mengklik tombol Run
3. Amati dan simpulkan maksud dan tujuan dari program di atas

………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………

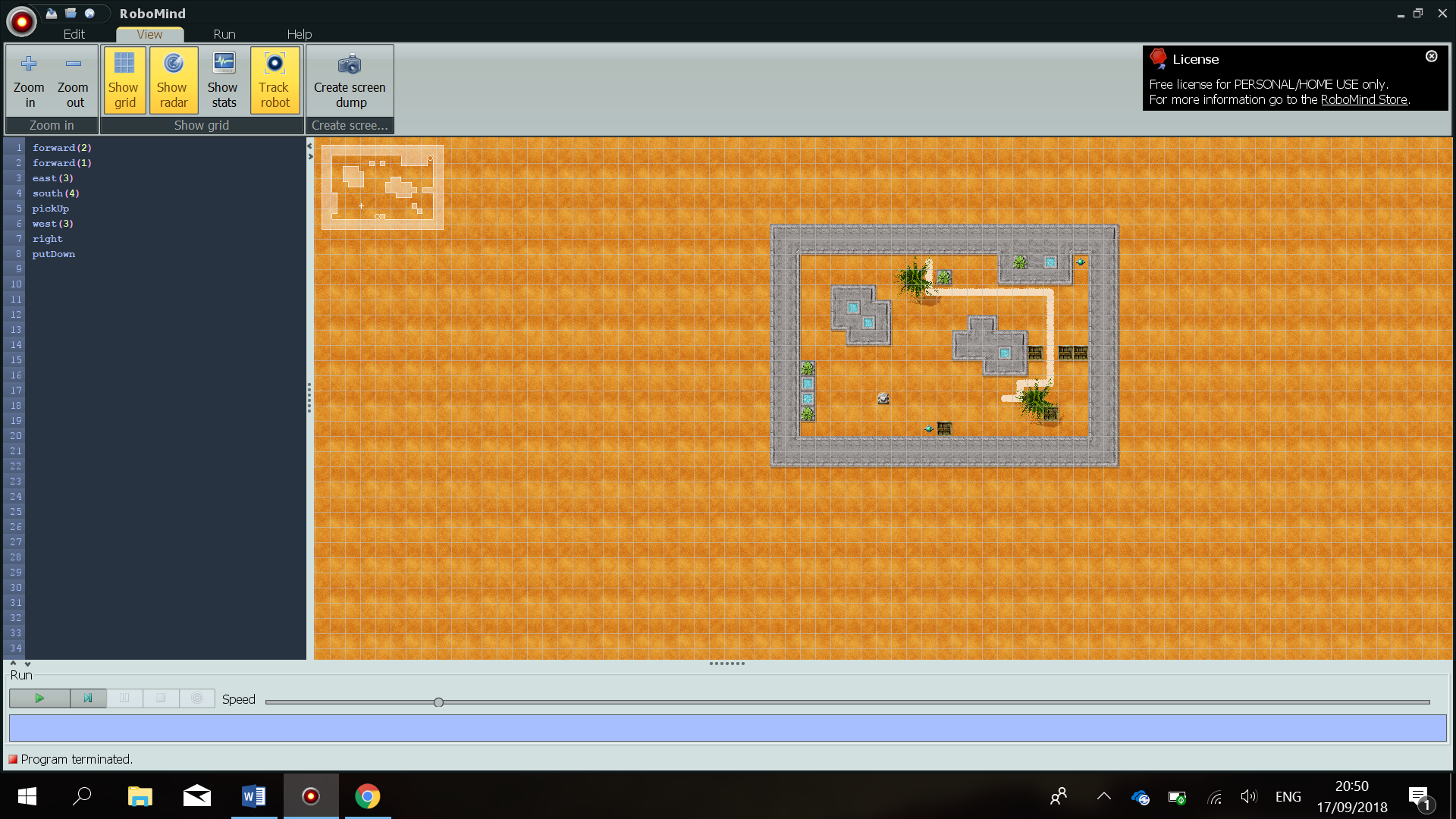
**Latihan:**

1. Tulislah kembali program di atas dengan susunan perintah yang berbeda sedemikian hingga diperoleh hasil yang sama dengan program di atas! Lakukan *save file as* dengan nama **sekuensial02.irobo**

………………………………………………………………………………………………………………………………………………  
………………………………………………………………………………………………………………………………………………

1. Tuliskan program dengan baris seminimal mungkin untuk memprogram si robot sedemikian hingga dia bisa membawa beacon yang posisinya di paling bawah pada bagian map (lihat tanda pada gambar), kemudian meletakkannya di titik lokasi pada tanda gambar. Adapun ketentuannya adalah bahwa rute pergerakan robot mengikuti garis warna putih.

Posisi akhir beacon



Posisi awal beacon

Simpan dengan nama file **sekuensial03.irobo**

………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………

1. Buka map bernama **followLine.map**. Kemudian tulislah program bagi robot supaya bisa mengambil beacon yang ada di pojok kanan atas, lalu membawanya dan meletakkan beacon di titik yang ditentukan berikut ini. Simpan dengan nama file **sekuensial04.irobo**

Posisi awal beacon



Posisi akhir beacon

Ketentuan:

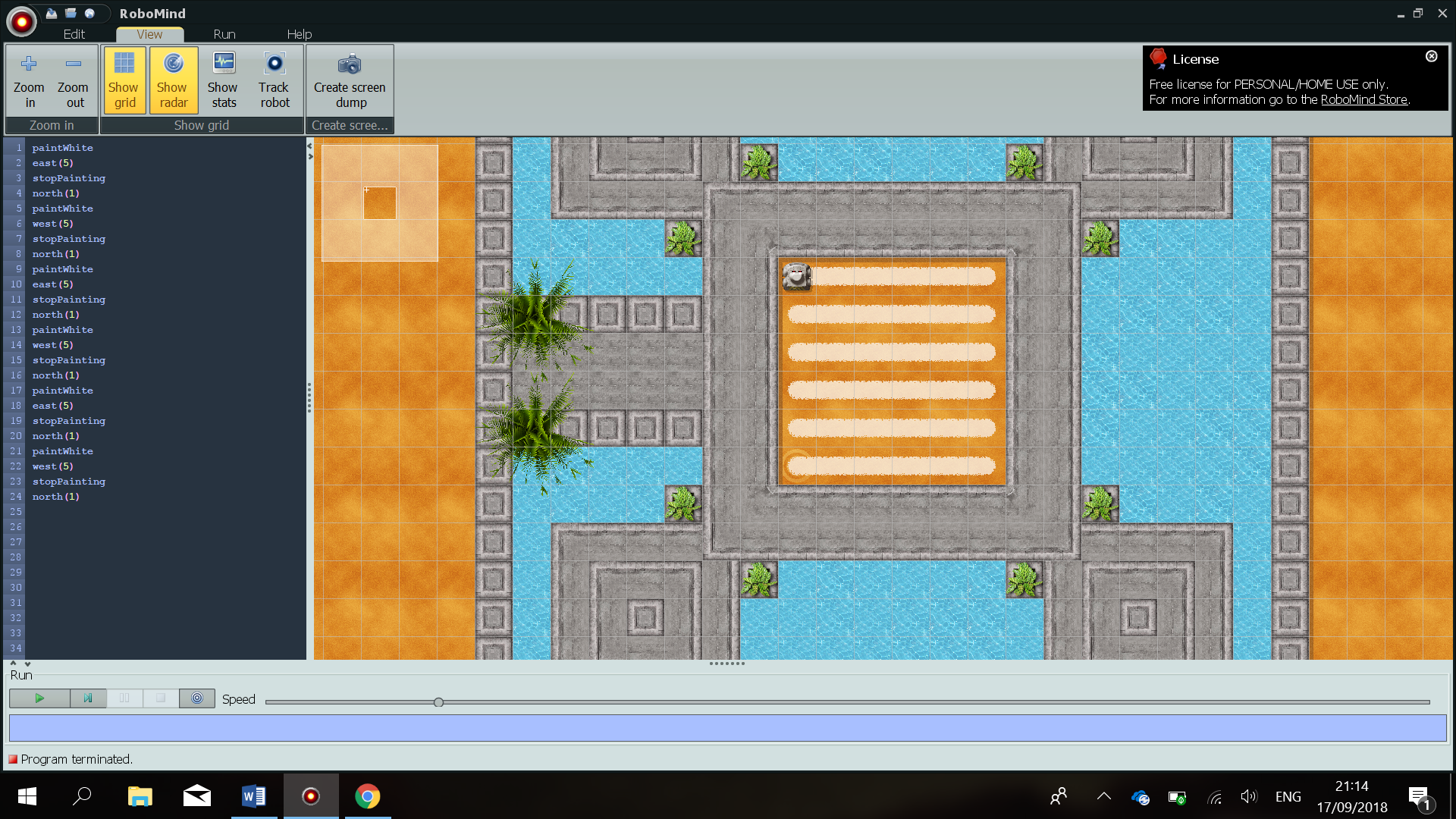
* Selama robot bergerak harus mengikuti garis warna putih
* Buat program dengan jumlah baris seminimal mungkin

1. Buka map bernama **passBeacon.map**. Kemudian tulislah program bagi robot supaya bisa melalui serangkaian beacon dan berhenti di titik putih yang ada dalam map! (jangan menggunakan perulangan terlebih dahulu). Upayakan jumlah baris program seminimal mungkin! Simpan dengan nama file **sekuensial05.irobo**

………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………

1. Buka map bernama **castle.map**. Kemudian tulislah program supaya robot bisa mengecat warna putih pada map menjadi seperti tampilan berikut ini! Simpan dengan nama file **sekuensial06.irobo**



………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………

1. Modifikasilah program dari nomor 5, sehingga diperoleh tampilan map seperti pada gambar berikut ini! Lakukan *save file as* dengan nama file **sekuensial07.irobo**



………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………

1. Bukalah map **changeBelt12.map**. Kemudian buatlah program sehingga robot bisa memindahkan semua beacon yang ada di track warna putih ke track warna hitam. Simpan dengan nama file **sekuensial08.irobo**

………………………………………………………………………………………………………………………………………………  
………………………………………………………………………………………………………………………………………………