

WORKSHOP PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VIRTUAL VEX

**UNTUK SMA SANTA ANGELA KELAS X DAN XI
09-10 AGUSTUS 2023**



TIM PENGABDI

Ir. Teddy Marcus Zakaria (720004/ 0418036901)

Sulaeman Santoso, S.Kom., M.T. (710096/ 0416058301)

Hendra Bunyamin, S.Si., M.T. (720001/ 0404127601)

Andreas Widjaja, S.Si., M.Sc., Ph.D. (720282/ 0419097204)

Tiur Gantini, S.T., M.T. (710071/ 0425027606)

Sendy Ferdian Sujadi, S.Kom., M.T. (730062/ 0410028802)

Dr. Bernard R. Suteja, S.Kom., M.Kom. (720062/ 0408038001)

Tjatur Kandaga, S.Si., M.T. (720080 / 0407107102)

2272004 - Jessica Alvina Luwia	2272037 - Rasyaad Hildebrand Gunawan
2272029 - Josephine Alvina Luwia	2272008 - Elmosius Suli
2273012 - Jocelyn Ellyandra Wijaya	2272007 - Kevin Owen
2173018 - Felisa Angeline Ongkowidjojo	2172030 - Femmy Friscilla Susilo
2273010 - Hana Saputri	2272013 - Benaya Andrias Kuncorobudi
2272041 - Rafael Cavin	2072021 - Jeremia Daud Halim
2273001 - Olivia Justine Audia	2072001 - Matthew Julian
2172026 - Nathan Joshua	2273019 - Raymond Surya
2172003 - Yehezkiel David Sm	2273001 - Olivia Justine Audia
2173025 - Rosi Rachel Dame	

**FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN MARANATHA
BANDUNG
2023**

Daftar Isi

DAFTAR ISI	2
BAB I MENGENAL SIMULATOR ROBOT VR VEX	4
A. Visual Programming Language (VPL)	4
B. Struktur Pemrograman atau Algoritma	5
C. Mengenal Jenis-jenis Robot.....	5
D. Bagaimana memprogram Robot?.....	8
Memulai Virtual Robot (VR Vex).....	8
Memilih Playground VR Vex	9
Tampilan Web VR Vex (VR.Vex.com).....	10
Mengerakkan Maju-Mundur, Belok Kiri-Kanan, Berhenti.....	12
E. LATIHAN MEMBUAT PROGRAM ROBOT SEDERHANA.....	13
Menggeser blok, mengganti parameter angka, menyalin blok.....	13
Memilih Warna Pen dan Menurunkan/menaikkan Pen	15
Memberi Nama, Menyimpan, Memanggil Proyek VR Vex.....	16
F. Tugas T01-Buat Kotak 5 buah.....	16
G. Kuis	18
BAB 2 CARA MEMBUAT PROGRAM ROBOT DAN ALGORITMA SEQUENCE	21
A. Cara memprogram robot (Block, Blocks with a Python Viewer dan Python)	21
Blocks	21
Blocks with a Python viewer	21
Python.....	21
B. Berkenalan Lebih Jauh Kode Blok (BLOCK CODE)	22
C. Algoritma Sequence (sekuensial).....	27
D. Tugas T02a-Buat Kotak 4 buah	27
E. Tugas T02b-Buat Segitiga 3 buah	28
F. Tugas T02c-Buat Segienam 2 buah	30
BAB 3 PENGULANGAN (ALGORITMA LOOPS)	31
A. Algoritma Pengulangan (Looping/Loops).....	31
B. T03a-Buat Kotak 5 buah (Gunakan Pengulangan Repeat).....	33
C. T03b-Buat Belah ketupat 3 buah (Gunakan Pengulangan Repeat)	34

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

D. Kuis	35
BAB 4 TIPE DATA, VARIABEL, OPERATOR DAN FUNGSI	37
A. Mengenal berbagai tipe data	37
B. Variabel	38
C. Mendefinisikan, menggunakan Variabel, List dan 2D List	41
Variabel List	43
Variabel 2D List (List 2 dimensi atau matriks).....	43
D. Mengenal operasi aritmatika, perbandingan, logika dan fungsi-fungsi	45
Operasi Aritmatika.....	46
Operasi Perbandingan	47
Operasi Logika.....	48
Fungsi Matematika dalam VR.Vex	48
Memantau (monitor) nilai variabel	51
E. T04a-Buat Segitiga Siku-siku	52
F. T04b-Buat 4 buah Segitiga Siku-siku	52
G. T04c-Buat 4 buah Segitiga Siku-siku (Gunakan List)	53
H. T04d-Buat 4 buah Segitiga Siku-siku (Gunakan 2DList).....	53
BAB 5 PEMILIHAN KASUS DAN SENSOR WARNA.....	54
A. Algoritma Pemilihan Kasus (Selection).....	54
B. T05a- Maju Zigzag 1-10, 20-11, 21-30, dst (Gunakan Repeat, Tanpa If-Then-Else).....	56
C. T05b- Maju Zigzag 1-10, 20-11, 21-30, dst (Gunakan Repeat, dan If-Then-Else).....	57
D. T05c- Maju Zigzag 1-10, 11-20, 31-40, dst (Gunakan Repeat, dan If-Then-Else)	58
E. Mengenal Sensor	59
F. Disk Maze Playground	60
BAB 6 MY BLOCKS (SUB PROGRAM)	62
A. Mengenal perintah My Blocks atau Sub-program	62
B. Proyek menggunakan My Blocks buatan sendiri	63
My Block tanpa parameter	63
My Block dengan parameter	64
C. T06a- Proyek Segi Tiga Puluh	66
D. T06b- Proyek Segi Banyak.....	67
KETENTUAN KOMPETISI “MENGGAMBAR NAMA/LAMBANG/LOGO TIM MENGGUNAKAN ROBOT VR VEX”:.....	69

BAB I

MENGENAL SIMULATOR ROBOT VR VEX

Pelajaran 1 MENGENAL SIMULATOR ROBOT VR VEX	Target Pelajaran <ol style="list-style-type: none"> 1. Mengenal jenis-jenis robot masa kini dan masa datang 2. Mengenal Robot VR Vex, Playground dan Pemrograman Visual dalam simulator Robot VR.VEX.COM 3. Membuat program sederhana untuk menggerakkan Robot, menggunakan playground Number Grid Map
	Deskripsi Pelajaran Siswa diperkenalkan berbagai jenis robot, simulator robot VR Vex dan membuat program sederhana untuk menggerakkan robot.
Durasi Belajar 45-60 menit	Bahan Pelajaran Lembar Kerja 1.1 – Jenis-jenis robot Lembar Kerja 1.2 – Tampilan web VR Vex Lembar Kerja 1.3 – Blok untuk menggerakkan robot (Drive Train) Lembar Kerja 1.4 – Playground Number Grid Map

Sebelum menggunakan simulator robot VR VEX, beberapa hal perlu dipahami, antara lain :

- Apa yang dimaksud VPL (Visual Programming Language)?
- Struktur Pemrograman atau Algoritma
- Mengenal Jenis-jenis Robot
- Bagaimana memprogram Robot VR VEX (Virtual Robot Vex)

A. Visual Programming Language (VPL)

Dalam dunia komputasi, bahasa pemrograman visual (VPL) adalah bahasa pemrograman yang memungkinkan pengguna membuat program melalui elemen program secara grafis (berbentuk blok-blok atau *puzzle*). Dengan VPL pemrograman ditampilkan secara visual melalui simbol grafis, contoh VPL yang terkenal adalah Scratch. **Scratch** adalah sebuah bahasa pemrograman visual yang memungkinkan pemula seperti siswa, guru, atau orangtua untuk belajar membuat program tanpa harus memikirkan salah-benar penulisan sintaks. Bahasa pemrograman ini dibuat oleh MIT (Media Lab dari Massachusetts Institute of Technology). Dengan Scratch, pengguna dapat membuat sendiri animasi, permainan, karya kesenian, dan lain-lain. Salah satu VPL antara lain VR VEX yang akan dibahas dalam buku ini. VPL lainnya seperti Alice, GameMaker, Kodu, Lego Mindstorms, MIT App Inventor, TinkerCad.

Berbeda dengan pemrograman reguler yang banyak menggunakan teks (sintaks tertulis) untuk membuat suatu program. Pemrograman atau yang sering kita sebut dengan

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

programming, ini merupakan proses yang memungkinkan pembuatan perangkat lunak komputer untuk suatu tujuan misal Program untuk Mengelola Penjualan, Pembelian, Persediaan. Program yang dibuat oleh manusia adalah serangkaian instruksi/kode yang diperintahkan kepada komputer untuk melakukan suatu tujuan.

B. Struktur Pemrograman atau Algoritma

Algoritma adalah metode atau langkah yang direncanakan secara tersusun dan terstruktur untuk menyelesaikan suatu permasalahan. Algoritma berasal dari kata *algoris* dan *ritmis* yang pertama kali diperkenalkan oleh Abu Ja'far Muhammad Ibn Musa Al Khwarizmi pada 825 M di dalam buku Al-Jabar Wa-al Muqabla.

Ada tiga bentuk algoritma yang umum digunakan, yaitu sequence (algoritma sekuelial/berurutan), selection (pemilihan kasus tergantung kondisi yang dihadapi atau dikenal dengan algoritma pencabangan) dan looping/loops (pengulangan, digunakan untuk menyingkat perintah yang berulang).

TIGA JENIS BENTUK ALGORITMA



Beberapa orang sering kesulitan memahami pengertian algoritma dan sering dianggap sama antara algoritma dan program. Pada nyatanya, keduanya memang berhubungan namun memiliki pemahaman yang berbeda.

C. Mengenal Jenis-jenis Robot

Kata robot berdasarkan Kamus Besar Bahasa Indonesia (KBBI) dapat berarti alat berupa orang-orangan dan sebagainya yang dapat bergerak (berbuat seperti manusia) yang dikendalikan oleh mesin. Kenyataan robot banyak jenisnya, tidak harus menyerupai orang. Berbagai jenis robot diantaranya ¹:

- **Robot Menyerupai Manusia (Humanoid).**

¹ Beberapa Contoh Jenis Robot, <http://www.robo-pal.com/know3.html>

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX



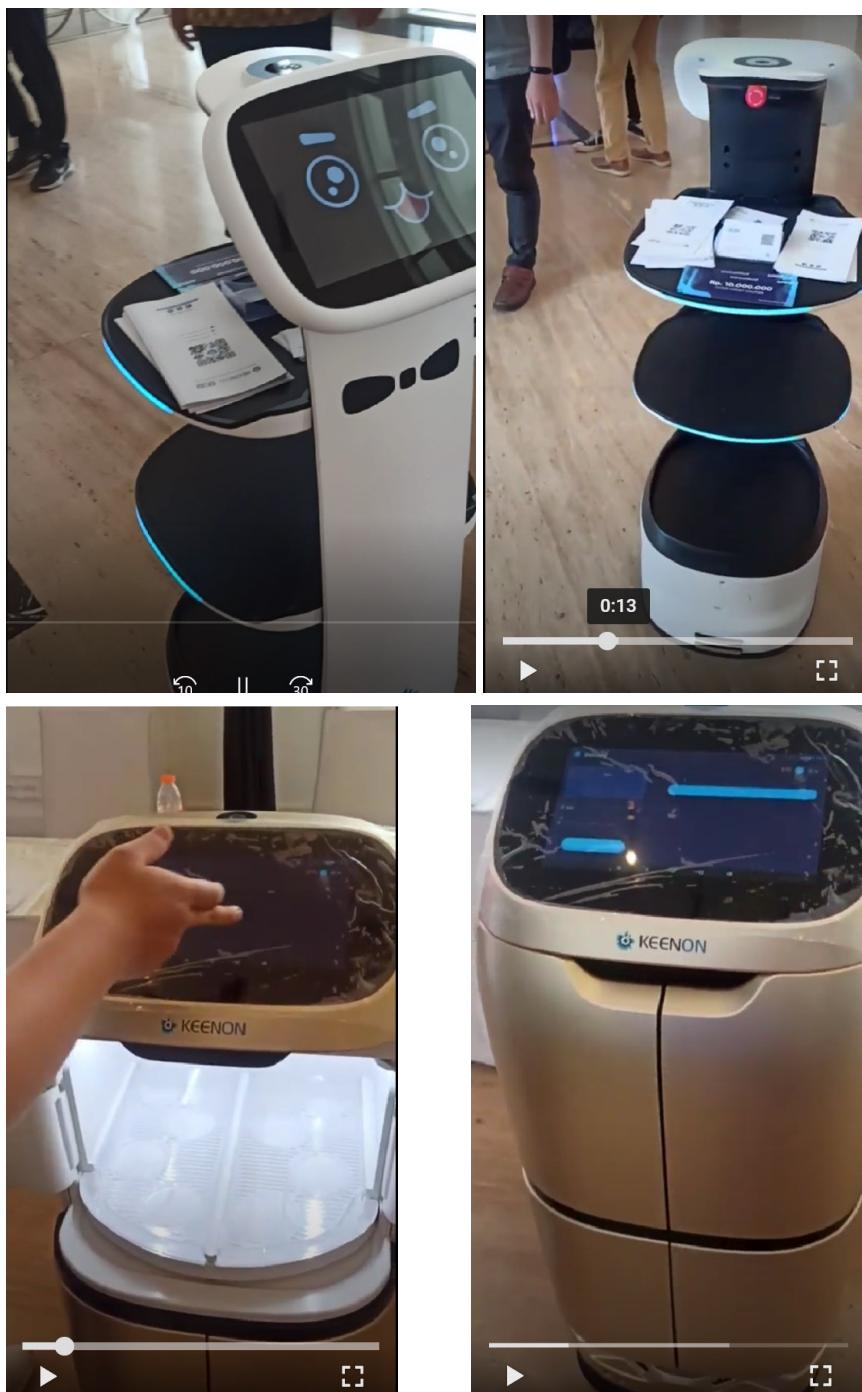
- **Robot Berkaki (Legged).**
- **Robot Terbang (Flying Robot)**
- **Robot Dalam Air (Underwater Robot).**
- **Robot Tangan (Manipulator).**



Gambar 1 TOSY Industrial Robot at Automatica 2010, Munich, Germany (Sumber : Wikimedia Common)

- **Robot Bergerak (Mobile).** Ciri khasnya adalah mempunyai aktuator berupa roda untuk menggerakan keseluruhan badan robot tersebut sehingga robot dalam melakukan perpindahan posisi dari satu titik ke titik yang lain. Contoh Robot Pengantar Barang dalam Pameran Metaverse di Jakarta, Maret 2023, pada Gambar 1. Robot A ini bisa diprogram untuk berkeliling di area pameran tanpa menabrak benda atau orang di sekitarnya. Robot B dapat diprogram untuk mengantar barang ke lokasi tertentu dalam suatu gedung berlantai-lantai, barang bisa diambil oleh penerima yang memiliki password untuk membuka pintunya.

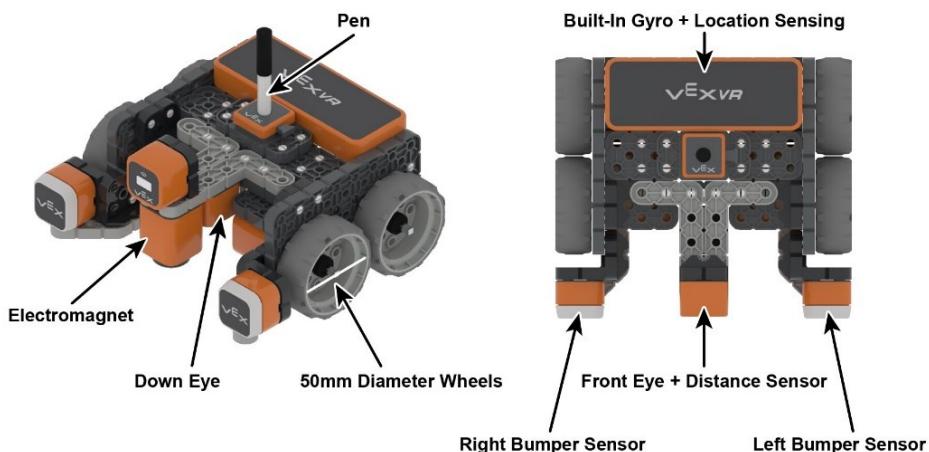
PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX



Gambar 2 Robot Pengantar Barang A dan B, Jakarta 2023 (Sumber : koleksi pribadi)

Dalam buku ini akan dibahas mengenai Robot Bergerak jenis VR VEX menggunakan simulator berbasis web atau Virtual Robot (VR), sehingga untuk mencobanya tidak memerlukan perangkat fisiknya. VR VEX tampak depan dan tampak atas seperti pada Gambar 3.

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX



Gambar 3 Virtual Robot (VR VEX) (Sumber : <https://www.vexrobotics.com/vexcode/vr>)

D. Bagaimana memprogram Robot?

Belajar pemrograman lewat robot, sangat menarik dan tidak membosankan. Yuk kita belajar membuat program robot menggunakan Virtual Robot (VR) Vex. VRVEX dapat diakses melalui browser dengan alamat VR.VEX.Com. Melalui website ini memungkinkan pengguna untuk mengendalikan robot virtual dan memprogramnya menggunakan lingkungan pemrograman yang disediakan. Melalui VR Vex, pengguna dapat belajar tentang konsep-konsep dasar pemrograman robot, seperti penggunaan sensor, pergerakan, dan navigasi. Pengguna juga dapat belajar tentang bagaimana memprogram robot agar melakukan tugas-tugas tertentu, seperti menghindari rintangan atau menyelesaikan rute tertentu.

Virtual Robot VR Vex menyediakan lingkungan simulasi (*playground*) yang realistik dan interaktif untuk belajar pemrograman robot. Dalam lingkungan ini, pengguna dapat melihat dan mengendalikan robot virtual dalam tiga dimensi, dan melihat perubahan program perilaku robot.

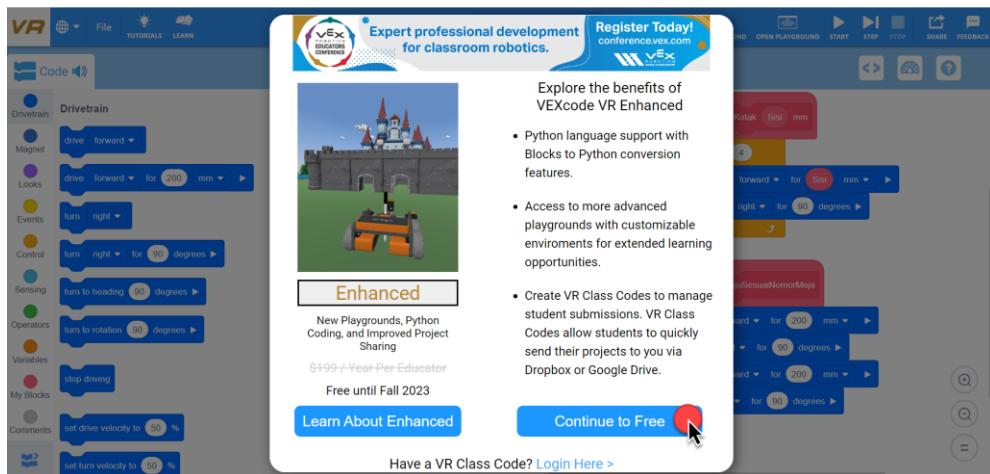
VR Vex juga dapat menjadi alat yang berguna bagi pengajar dalam mengajarkan konsep-konsep dasar pemrograman robot. Dalam lingkungan ini, pengajar dapat membuat tugas dan latihan untuk siswa yang dapat membantu mereka memperoleh pemahaman yang lebih baik tentang konsep-konsep tersebut. VR Vex memberikan pengalaman belajar yang unik dan interaktif bagi pengguna, serta membantu meningkatkan efektivitas pembelajaran di bidang pemrograman robot.

Memulai Virtual Robot (VR Vex)

Untuk memulai Virtual Robot VR Vex, Anda perlu mengunjungi situs web VRVEX di <https://vr.vex.com> dan mendaftar sebagai pengguna. Setelah mendaftar, Anda dapat mengakses Virtual Robot VR Vex dan berbagai konten dan program pelatihan lainnya yang tersedia di situs web. Berikut adalah langkah-langkah yang perlu Anda lakukan untuk memulai Virtual Robot VR Vex:

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

1. Buka situs web VRVEX di <https://vr.vex.com/> dan klik tombol "Sign Up" di sudut kanan atas halaman untuk mendaftar sebagai pengguna.
2. Isi formulir pendaftaran dengan informasi pribadi Anda, seperti nama, alamat email, dan password.
3. Setelah mendaftar, klik "Continue to Free" atau "Login here" (bila memiliki VR Class code), untuk membuka lingkungan Virtual Robot VR Vex, seperti pada Gambar 4.

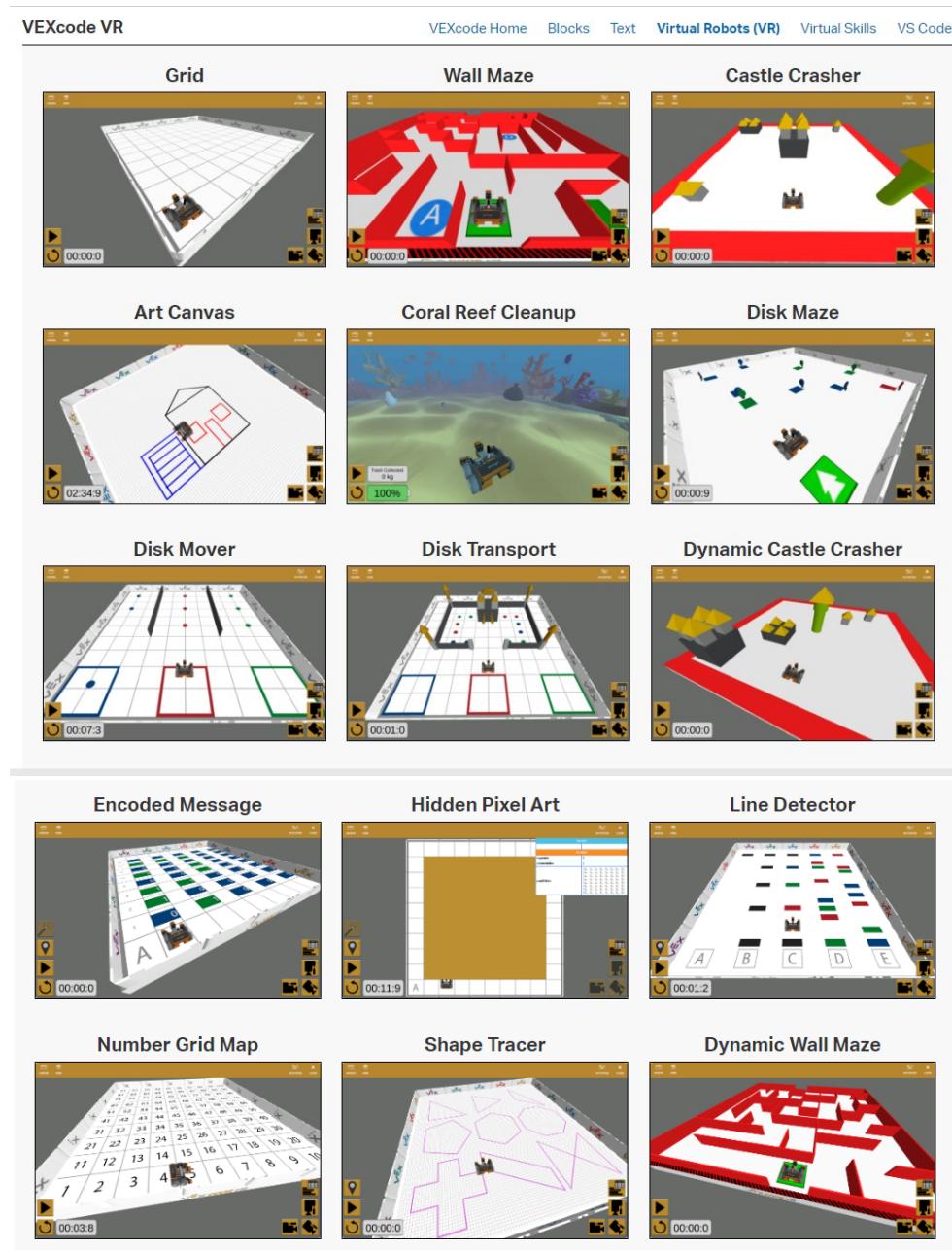


Gambar 4 Memulai VRVEX

Memilih Playground VR Vex

Di dalam lingkungan Virtual Robot, Anda dapat mempelajari konsep dasar pemrograman robot dan memprogram robot virtual untuk melakukan tugas-tugas tertentu. Gunakan lingkungan pemrograman yang disediakan untuk membuat kode pemrograman robot dan melihat bagaimana perubahan pada kode mempengaruhi perilaku robot. Ada 15 lingkungan pemrograman yang telah disediakan, bersifat gratis seperti pada Gambar 5. Penggunaan semua playground ini akan dibahas dalam buku ini pada bab selanjutnya. Anda akan belajar pemrograman yang menarik dan unik untuk setiap playground ini. Melalui Playgroud Grid, Anda dapat memprogram robot untuk bergerak sesuai yang lintasan yang diinginkan. Playgroud Wall Maze, Anda dapat memprogram robot untuk mencari jalan keluar, tanpa menabrak dinding. Playgroud Art Canvas, Anda dapat memprogram robot untuk menggambar bentuk yang biasa sampai bentuk yang rumit namun menarik.

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX



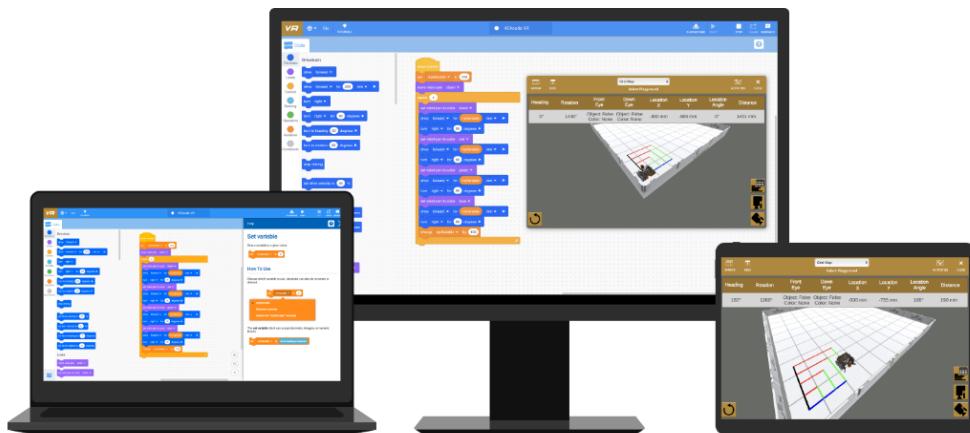
Gambar 5 Lingkungan Pemrograman VR Vex (Playground)

Playground ini diberi nama sesuai dengan fungsi robot yang akan diprogram, seperti Grid, Wall Maze, Castle Crasher, Art Canvas, Coral Reef Cleaner, Disk Maze, Disk Mover, Disk Transport, Dinamic Castle Crasher, Encoded Message, Hidden Pixel Art, Line Detector, Number Grip Map, Shape Tracer, Dinamic Wall Maze.

Tampilan Web VR Vex (VR.Vex.com)

Tampilan VR.VEX versi web seperti pada Gambar 6, bagian kiri adalah pemrograman menggunakan bahasa visual (*block*), bagian tengah tampak program dan tampilan virtual robot dan bagian kanan tampilan virtual robot yang diperbesar.

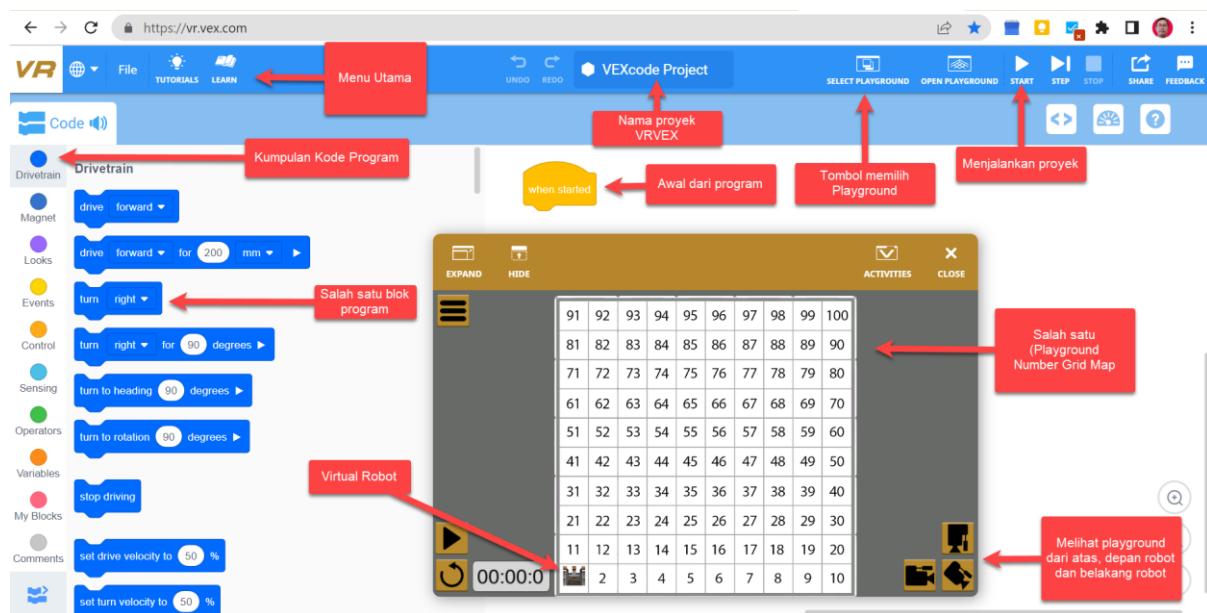
PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX



Gambar 6 Tampilan Web VR.Vex.com (Sumber : <https://www.vexrobotics.com/vexcode/vr>)

Pada Gambar 7 merupakan penjelasan sebagian fungsi-fungsi yang sering digunakan dalam lingkungan pemrograman VRVex. Anda dapat mencobanya, supaya terbiasa dengan tombol-tombol yang ada.

Bagian Atas. Bagian atas-kiri terdapat menu utama Language, File, Tutorial, Learn. Bagian atas-tengah terdapat menu Undo, Redo, Nama File Proyek VRVex. Bagian atas-kanan terdapat menu Select Playground, Open Playground, Start, Step, Stop, Share, Feedback.



Gambar 7 Tataletak fungsi-fungsi pada VRVex

Bagian Tengah. Bagian tengah-kiri terdapat kumpulan Code (Kumpulan Kode Program dalam bentuk blok visual) seperti Drivetrain, Magnet, Looks, Events, Control, Sensing, Operators, Variables, My Blocks, Comments. Bagian tengah-kanan terdapat area untuk meletakkan program, ditandai dengan blok [When Started]. Anda dapat menambahkan blok-blok di area ini sesuai proyek yang akan dibuat. Pada bagian ini ditampilkan visual dari playground dan robot VR Vex, Anda dapat melihat tampak atas, tampak samping dan tampak depan.

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

Menggerakkan Maju-Mundur, Belok Kiri-Kanan, Berhenti

Anda dapat memprogram robot untuk bergerak, belok, berhenti dan mengatur kecepatan bergerak, kecepatan belok. Blok-blok ini program ada pada Code Drivetrain seperti pada Gambar 8.



[Drive forward/reverse] : maju/mundur terus menerus
[Drive forward/reverse for 200 mm] : maju/mundur sejauh 200 mm

[Turn right/left] : belok kanan/kiri terus menerus
[Turn right/left for 90 degrees] : belok kanan/kiri sebanyak 90 derajat

[Turn to heading 90 degrees] :
Mengubah ke arah 90 derajat menggunakan sensor Gyro bawaan.

[Turn to rotation 90 degrees] :
Berputar ke arah 90 derajat menggunakan sensor Gyro bawaan.

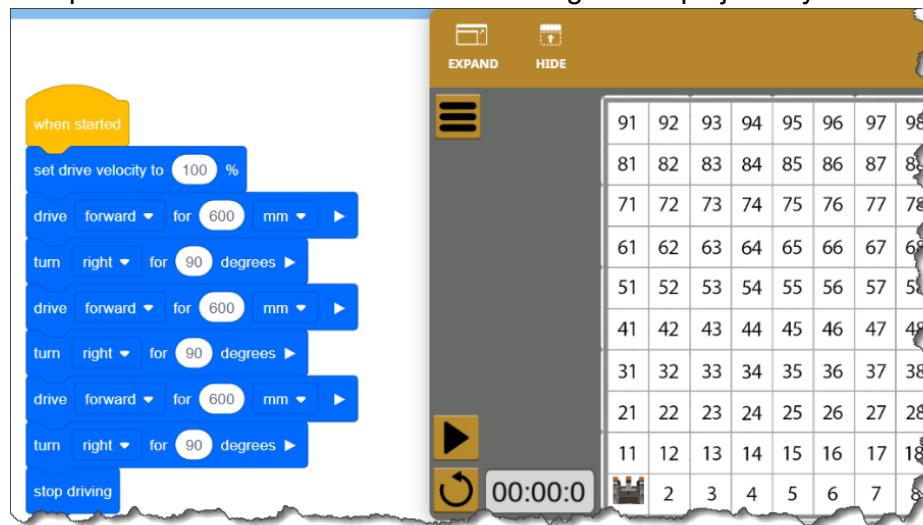
[Stop driving] : berhenti

[Set drive velocity to 50%] : Kecepatan bergerak 50% dari maksimum 100%

[Set drive rotation to 50%] : Kecepatan berputar 50% dari maksimum 100%

Gambar 8 Code Drivetrain

Berikut contoh proyek mengatur kecepatan robot bergerak, robot maju sejauh 600 mm, robot belok kanan 90 derajat, dan di bagian terakhir berhenti. Lihat Gambar 9. Playground yang digunakan Number Grid Map. Jika robot mulai pada angka1, maka pada angka berapa robot akan berhenti? Satu kotak angka berapa jaraknya?



Gambar 9 Proyek Robot bergerak dan belok dengan kecepatan 100%

E. LATIHAN MEMBUAT PROGRAM ROBOT SEDERHANA

Setelah Anda login, pilih playground “Number Grid Map”, gunakan tombol [SELECT PLAYGROUND]. Anda akan membuat program untuk menggerakkan robot membentuk sebuah kotak.

Algoritma sebagai berikut :

1. Mulai
2. Maju ke depan sejauh 1800 mm
3. Belok kanan 90 derajat
4. Maju ke depan sejauh 1800 mm
5. Belok kanan 90 derajat
6. Maju ke depan sejauh 1800 mm
7. Belok kanan 90 derajat
8. Maju ke depan sejauh 1800 mm
9. Belok kanan 90 derajat
10. Berhenti
11. Selesai

Potongan blok program yang akan dibuat seperti pada Gambar 10.



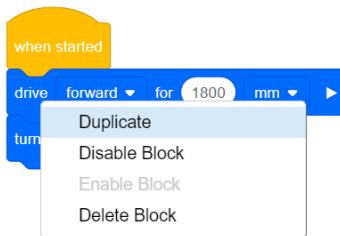
Gambar 10 Potongan program menggerakkan robot

Menggeser blok, mengganti parameter angka, menyalin blok

Langkah untuk membuat program ini sebagai berikut :

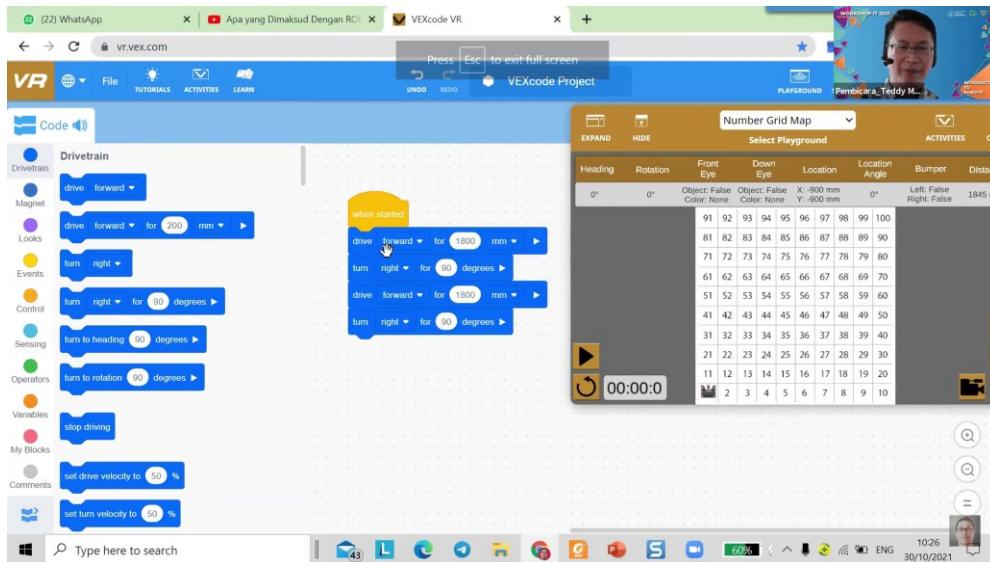
1. Pilih Code [Drivetrain]
2. Pilih blok program [drive forward for 200 mm], geser ke bagian bawah blok [When started]. Ganti angka 200 menjadi 1800.
3. Pilih blok program [turn right for 90 degrees], geser ke bagian bawah blok [drive forward for 1800 mm].
4. Sekarang Anda akan menyalin (*duplicate*) 2 baris blok yang sudah dibuat, caranya klik-kanan-mouse sehingga muncul menu, lalu pilih [Duplicate]. Setelah itu, blok yang disalin diletakkan di bagian bawahnya.

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

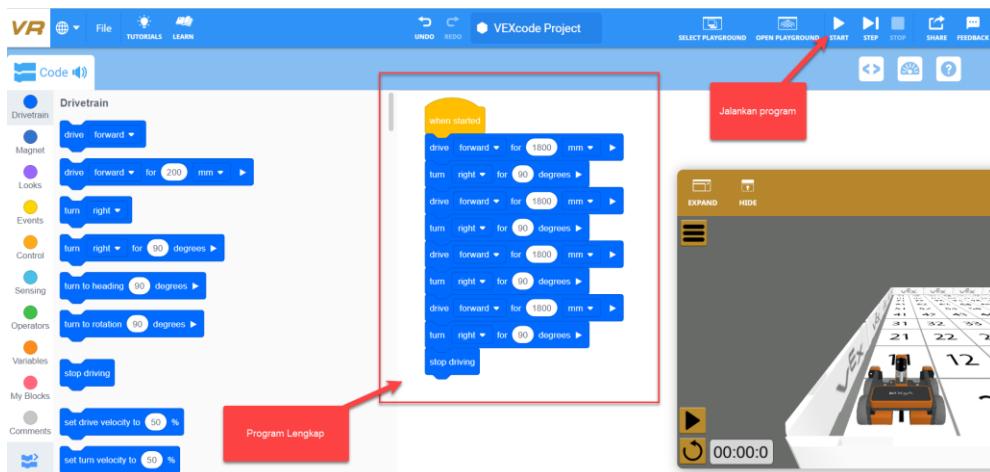


Gambar 11 Cara menyalin blok program

- Setelah disalin, maka akan tampak blok program seperti pada Gambar 12. Ulangi penyalinan sehingga blok program lengkap seperti pada Gambar 13.



Gambar 12 Tampilan Program Menggerakkan Robot

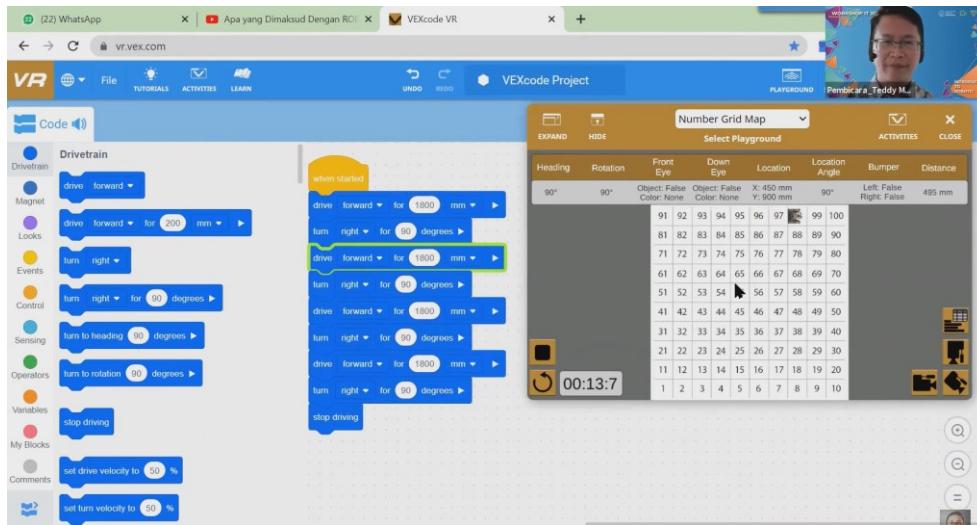


Gambar 13 Tampilan Program Menggerakkan Robot lengkap

- Setelah lengkap, tambahkan blok program [Stop Driving] untuk menghentikan robot.
- Tekan tombol [Start] untuk menjalankan proyek program robot yang sudah dibuat. Perhatikan robot akan bergerak maju, belok dan seterusnya sehingga membuat lintasan kotak.

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

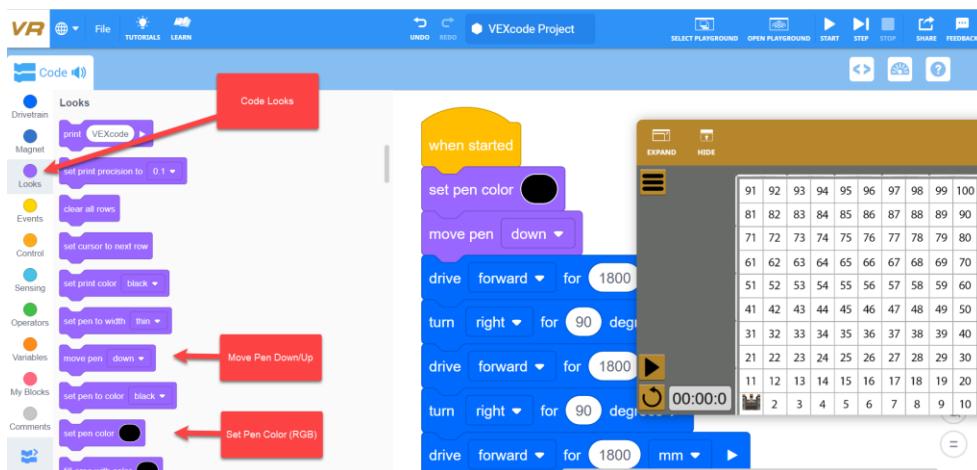
Mudah bukan? Anda sudah membuat program robot untuk membuat lintasan kotak, seperti pada Gambar 14.



Gambar 14 Tampak visualisasi robot bergerak membentuk kotak

Memilih Warna Pen dan Menurunkan/menaikkan Pen

Anda dapat menambahkan blok program, supaya saat robot bergerak sambil menggambar lintasan yang dilaluiinya. Tambahkan blok program [Set Pen Color] dan [Move Pen Down] ada pada Code **Looks**. Perhatikan Gambar 15, setelah ada penambahan set pen color dan move pen down, maka lintasan kotak yang dilewati oleh robot akan tergambar. Silahkan Anda mencobanya. Anda dapat memilih warna pen sesuai yang dikehendaki, format warna merupakan kombinasi warna Red, Green dan Blue atau disingkat RGB.

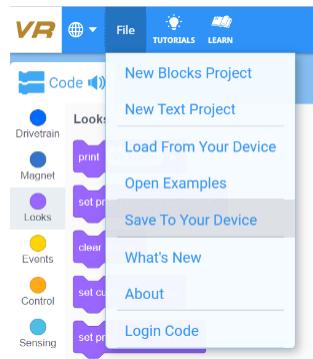
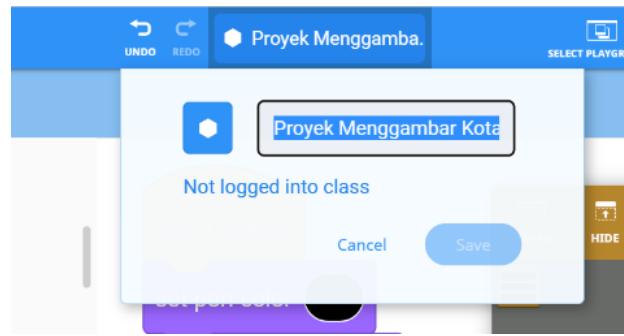


Gambar 15 Robot Menggambar Kotak

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

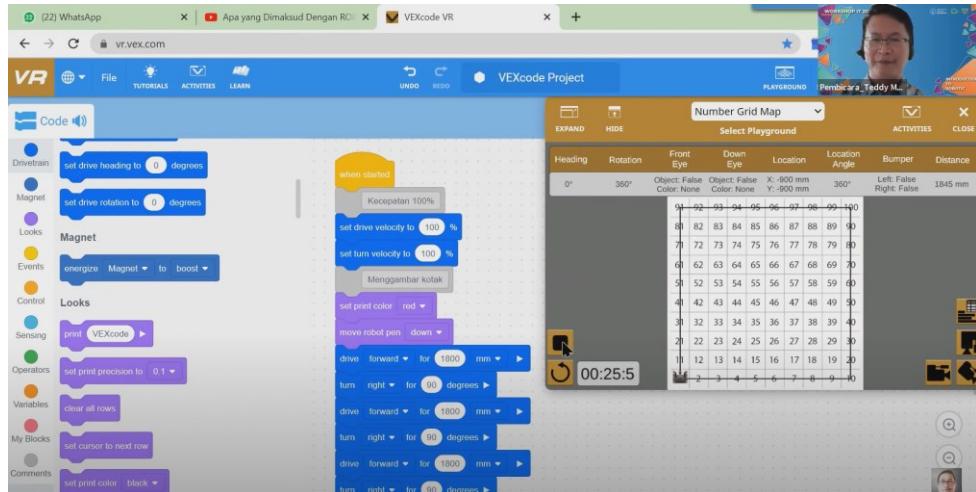
Memberi Nama, Menyimpan, Memanggil Proyek VR Vex

Setelah robot menjalankan misinya menggambar sebuah kotak, Anda harus memberi nama proyek misal "Proyek Menggambar Kotak". Kemudian menyimpan proyek tersebut ke komputer Anda, gunakan File - Save to Your Device.



Suatu kali Anda dapat memanggil proyek yang sudah ada buat, menggunakan File – Load From Your Device.

Pada Gambar 16 Robot menggambar sebuah kotak berwarna hitam. Anda dapat mencobanya.



Gambar 16 Proyek Robot Menggambar Kotak

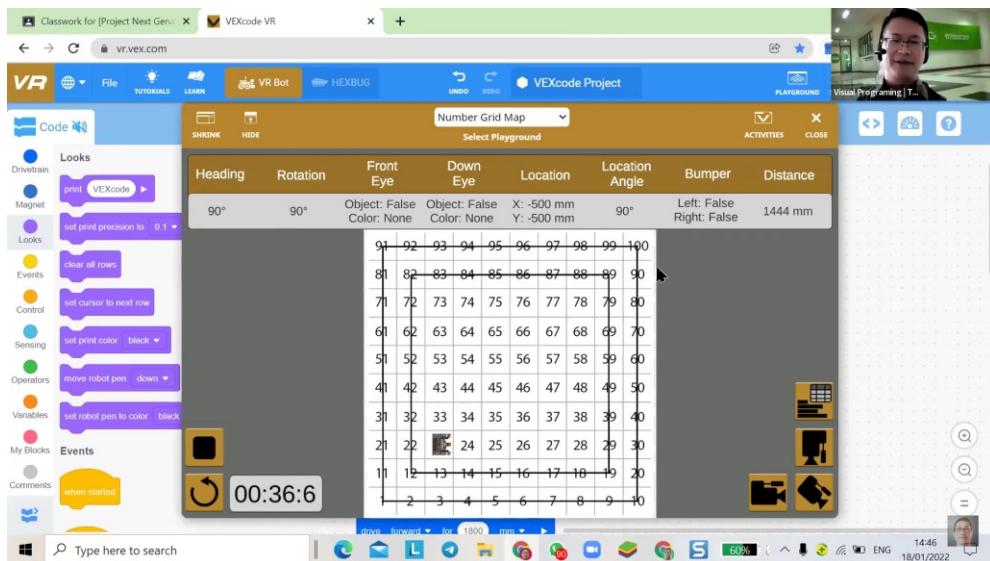
F. Tugas T01-Buat Kotak 5 buah

100 points – Kerjakan dalam 30 menit

Robot menggambar 5 kotak, menggunakan pensil warna merah.

Link : <https://www.youtube.com/watch?v=DynAn6elxz8>

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX



Algoritma dalam bentuk rangkaian berurutan (sequence) :

1. Mulai
2. Atur kecepatan robot 100%
3. Atur kecepatan belok robot 100%
4. Robot memilih pensil warna merah

5. Robot menurunkan pensilnya, siap untuk menggambar
6. Robot menggambar kotak pertama mulai dari angka 1-10, 10-100, 100-91, 91-1

7. Robot mengangkat pensilnya
8. Kemudian robot bergerak ke angka 12

9. Robot menurunkan pensilnya, siap untuk menggambar
10. Robot menggambar kotak ke-2 dari 12-19, 19-89, 89- 82, 82-12

11. Robot mengangkat pensilnya
12. Kemudian robot bergerak tanpa menggambar ke angka 23

13. Robot menurunkan pensilnya, siap untuk menggambar
14. Robot menggambar kotak ke-3 dimulai dari angka 23

15. Robot mengangkat pensilnya
16. Kemudian robot bergerak tanpa menggambar ke angka 34

17. Robot menurunkan pensilnya, siap untuk menggambar
18. Robot menggambar kotak ke-4 dimulai dari angka 34

19. Robot mengangkat pensilnya
20. Kemudian robot bergerak tanpa menggambar ke angka 45

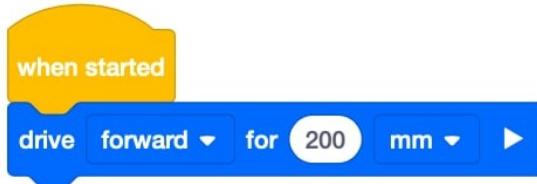
21. Robot menurunkan pensilnya, siap untuk menggambar
22. Robot menggambar kotak ke-5 dimulai dari angka 45

23. Robot berhenti

Beri nama proyek program ini “Kotak 5 buah - <Nama Anda>”, lalu simpan pada drive Anda dan capture hasil akhirnya.

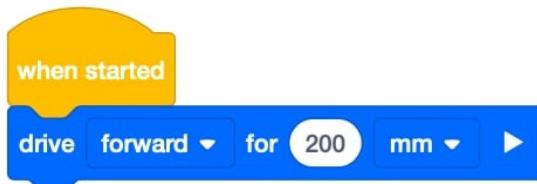
G.Kuis

1. Blok [drive for] berikut akan digunakan untuk **mundur 500 mm** (mili meter), perintah mana yang benar (jawaban bisa lebih dari satu)?



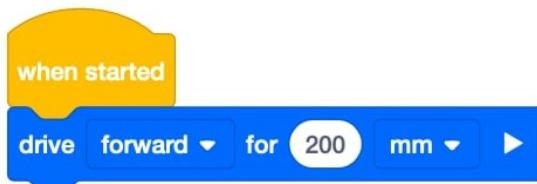
- a) Ubah forward ke reverse
- b) Ganti angka 200 menjadi -500
- c) Ubah forward ke reverse, dan ganti angka 200 menjadi -500
- d) Ganti angka 200 menjadi 500 dan ubah forward ke reverse

2. Blok [drive for] berikut akan digunakan untuk **maju 50 cm** (centi meter), perintah mana yang benar (jawaban bisa lebih dari satu)?



- a) Ganti angka 200 menjadi 500
- b) Ganti angka 200 menjadi -500
- c) Ubah forward ke reverse, dan ganti angka 200 menjadi -500
- d) Ganti angka 200 menjadi 500 dan ubah forward ke reverse

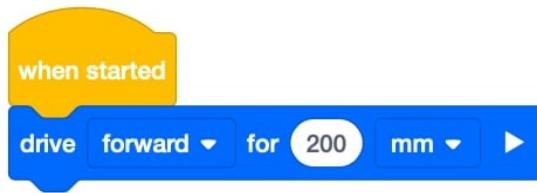
3. Blok [drive for] berikut akan digunakan untuk **belok 90 derajat dan maju 500 mm**, perintah mana yang benar (jawaban bisa lebih dari satu)?



- a) Tambahkan [Turn Right] setelah [When started]
- b) Tambahkan [Turn Right] setelah [When started] dan ganti angka 200 menjadi 500
- c) Tambahkan [Turn Right for 90 degree] setelah [When started]
- d) Tambahkan [Turn Right for 90 degree] setelah [When started] dan ganti angka 200 menjadi 500

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

4. Mana yang terbaik menjelaskan blok [drive for] berikut?



- a) Menggerakkan VR Robot mundur terus menerus
- b) Menggerakkan VR Robot ke kiri atau kanan
- c) Menggerakkan VR Robot maju atau mundur dengan jarak tertentu
- d) Menggerakkan VR Robot maju terus menerus

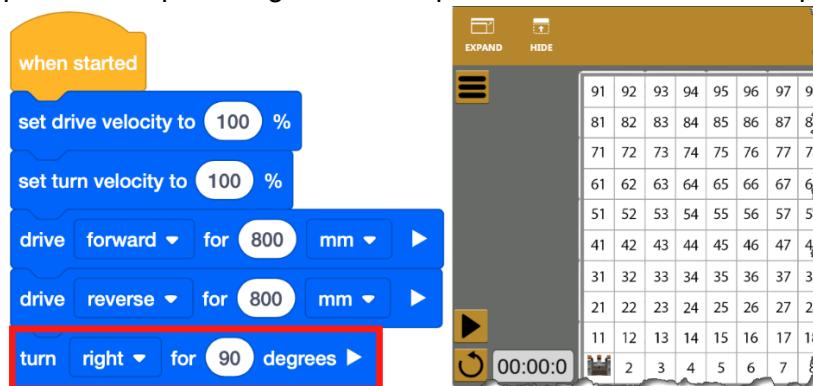
5. Blok mana yang digunakan untuk mengubah kecepatan dari drivetrain (jawaban bisa lebih dari satu)?

- a) [Set turn velocity]
- b) [Set drive velocity]
- c) [Set drive heading]
- d) [Set drive rotation]

6. Rentang nilai untuk mengatur kecepatan VR Robot bergerak?

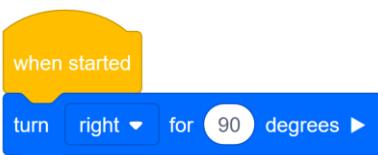
- a) 0% sampai 100%
- b) -100% sampai 100%
- c) 0% sampai 200%
- d) -100% sampai 200%

7. Apabila proyek ini dijalankan menggunakan playground Number Grid Map, dimana posisi robot pada angka 1, maka posisi terakhir robot berada pada angka?

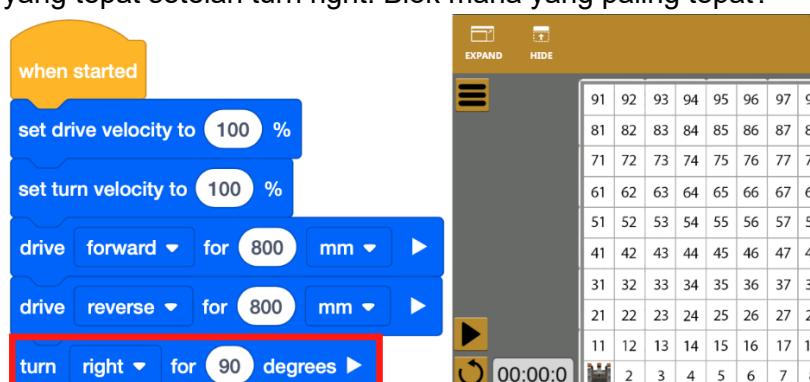


- a) 31
- b) 41
- c) 1
- d) 45

8. Mana yang terbaik menjelaskan blok [turn for] berikut?



PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

- a) Menggerakkan VR Robot ke kiri terus menerus
 - b) Menggerakkan VR Robot ke kanan terus menerus
 - c) Menggerakkan VR Robot maju atau mundur dengan jarak tertentu
 - d) Menggerakkan VR Robot ke kiri atau kanan sebesar derajat tertentu
9. Supaya posisi terakhir robot berada pada angka 5, maka perlu ditambahkan blok yang tepat setelah turn right. Blok mana yang paling tepat?
- 
- a) Drive forward for 800 mm
 - b) Drive reverse for 800 mm
 - c) Turn right for 800 mm
 - d) Drive forward

10. Apa yang dilakukan oleh VR robot, bila proyek ini dijalankan?



- a) VR Robot ke kiri terus menerus
- b) VR Robot ke kanan terus menerus
- c) VR Robot ke kiri dengan jarak tertentu
- d) VR Robot ke kanan sebesar derajat tertentu

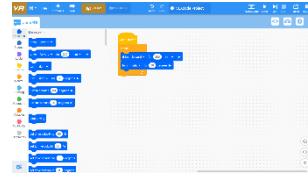
BAB 2

CARA MEMBUAT PROGRAM ROBOT DAN ALGORITMA SEQUENCE

Pelajaran 2 CARA MEMBUAT PROGRAM ROBOT DAN ALGORITMA SEQUENCE	Target Pelajaran
	<ol style="list-style-type: none"> 1. Mengenal cara memprogram robot (Block, Blocks with a Python Viewer dan Python) 2. Mengenal 3 konsep algoritma dalam pemrograman (Sequence, Loops dan Selection) 3. Memahami Algoritma Sequence (rangkaian baris demi baris) dengan membuat program Sequence. Gunakan playground Number Grid Map 4. Mendemonstrasikan program robot menggunakan sequence
Durasi Belajar 45-60 menit	Deskripsi Pelajaran Siswa diperkenalkan cara memprogram robot VR Vex dan konsep algoritma sequence yang digunakan dalam membuat sebuah program. Siswa membuat program, memantau perilaku robot dan menguji program yang telah dibuat

A. Cara memprogram robot (Block, Blocks with a Python Viewer dan Python)

Untuk memprogram Virtual Robot (VR) VEX ada tiga cara, yaitu :

Blocks	Blocks with a Python viewer	Python
		

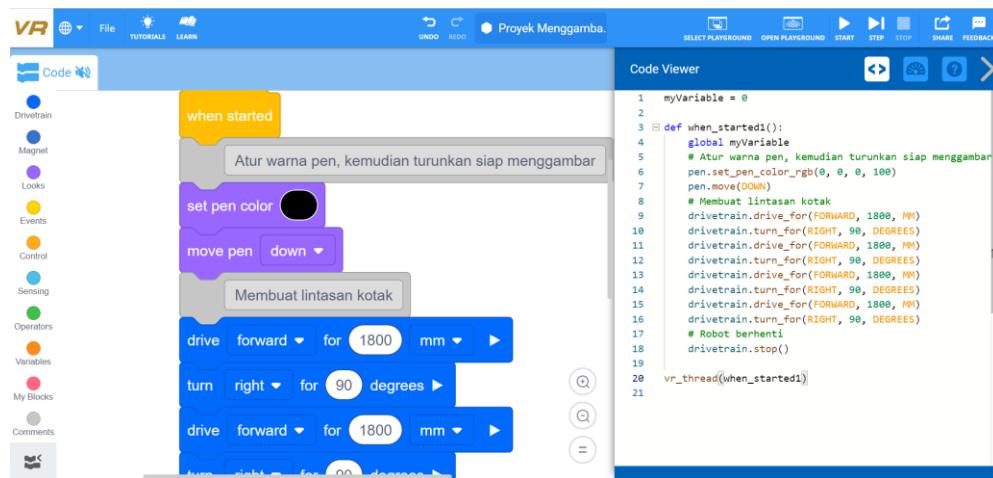
Menggunakan blok-blok yang dapat disusun mirip seperti puzzle

Gabungan blok-blok dan ditampilkan kode program dalam bahasa Python

Menggunakan bahasa Python

Berikut contoh proyek “Robot menggambar sebuah kotak” ditampilkan dalam blok dan bahasa Python, pada Gambar 17.

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX



Gambar 17 Menggambar kotak dalam blok program dan kode program Python

B. Berkenalan Lebih Jauh Kode Blok (BLOCK CODE)

Selain Code Drivetrain yang telah dijelaskan sebelumnya pada Gambar 8 Code Drivetrain, Anda akan mengenal Code lainnya Magnet, Looks, Events, Control, Sensing, Operators, Variables, My Blocks, Comments. Perhatikan warna-warna setiap kode blok : Drivetrain biru terang, Magnet biru tua, Looks ungu muda dan seterusnya. Tujuan warna ini untuk memudahkan Anda mencari letak setiap blok.

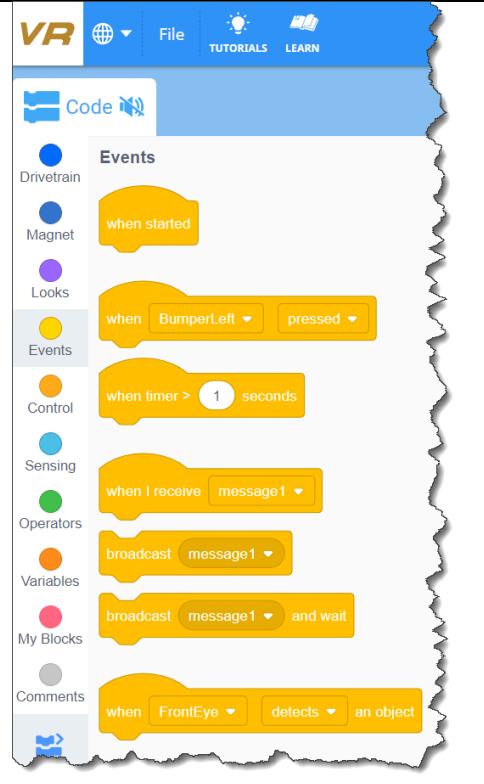
Pada bagian ini akan dijelaskan fungsi-fungsi dari setiap blok ini. Pada bab selanjutnya akan dibahas cara penggunaannya di dalam sebuah proyek.

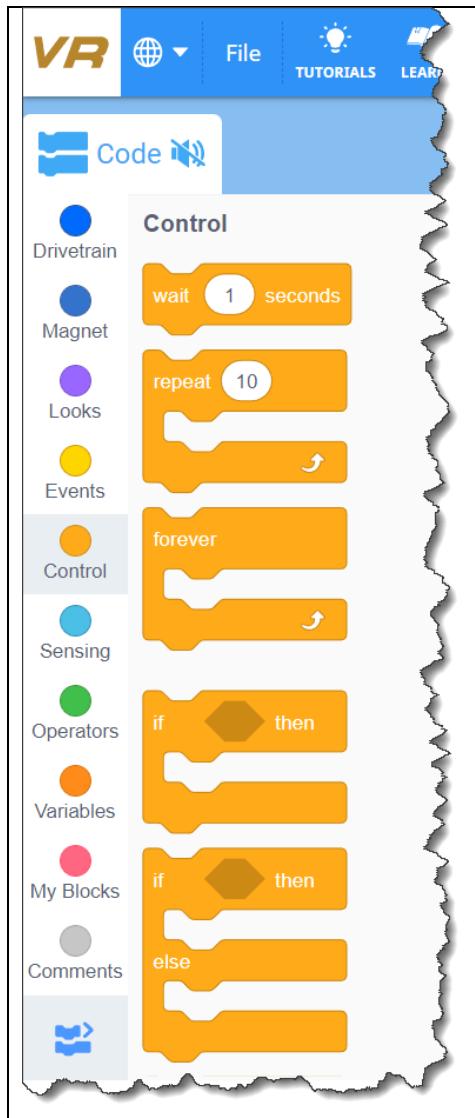
The image shows the VR VEX Code Editor with the 'Magnet' block category selected. A large callout box highlights the 'energize' block, which has three dropdown menus: 'Magnet', 'to', and 'drop'. Below the callout, a smaller box shows two options: 'boost' and 'drop'. To the right of the editor, a text box provides information about the 'energize' block:

[Energize Magnet to boost/drop] :
mengaktif/menghilangkan fungsi elektromagnet.

Magnet akan digunakan pada playground Disk Mover dan Disk Transport

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

	<p>[Print] : untuk mencetak kalimat atau isi variabel atau hasil perhitungan atau nilai suatu sensor ke layar</p> <p>[Set print precision to 0.1] : mengatur jumlah angka di belakang koma, apabila mencetak bilangan desimal. 0.1 berarti 1 angka di belakang koma atau perseruluhan. 0.01 berarti 2 angka di belakang koma atau per-seratus</p> <p>[Clear all rows] : membersihkan layar</p> <p>[Set cursor to next row] : mengatur posisi kursor pada baris selanjutnya atau turun baris</p> <p>[Set print color] : mengatur warna tulisan ke layar</p> <p>[Set pen to width extra thin/ thin/ medium/ wide/ extra wide] : mengatur lebar ujung pena saat menggambar, mulai dari ekstra tipis sampai ekstra lebar.</p> <p>[Move Pen down/up] : Menurunkan atau menaikkan posisi pena. Turun berarti siap menggambar.</p> <p>[Set pen to color] : Memilih warna pena untuk menggambar (hitam, merah, hijau atau biru)</p> <p>[Set pen color] : Memilih warna pena untuk menggambar dengan kombinasi RGB (Red-Green-Blue)</p>
	



Code Control

Blok kode kontrol adalah struktur digunakan untuk mengontrol program terhadap kondisi tertentu, seperti menunggu, mengulang blok kode, menjalankan blok kode tertentu jika suatu kondisi terpenuhi, dan menjalankan blok kode yang berbeda jika kondisi tersebut tidak terpenuhi, dsb.

[Wait 1 seconds] : menunggu 1 detik

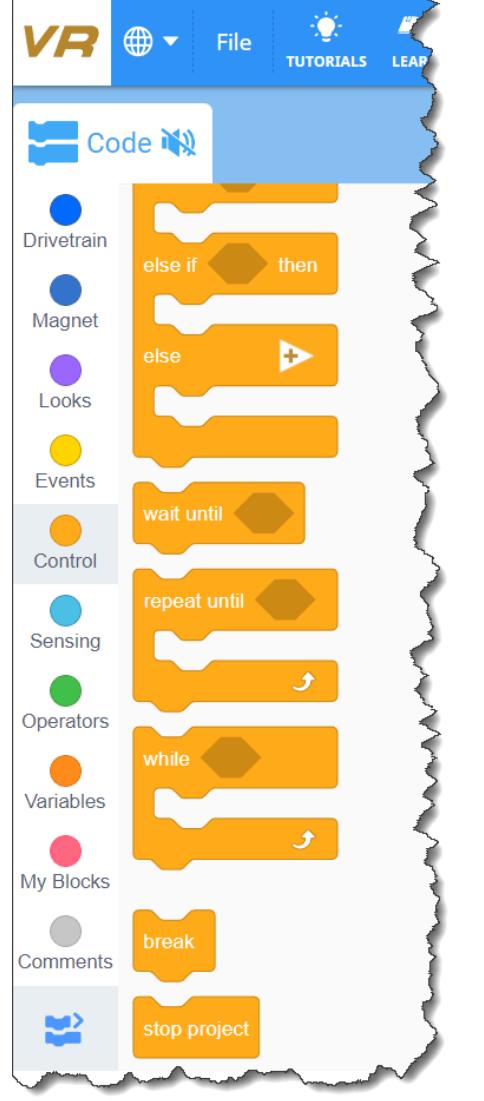
[Repeat 10] : mengulang blok perintah yang ada di dalamnya sebanyak 10 kali

[Forever] : mengulang blok perintah yang ada di dalamnya terus menerus. Pengulangan akan berhenti saat menemukan blok [Break]

[If <kondisi> then] : Jika kondisi true maka perintah-perintah yang ada di dalam then akan dijalankan

[If <kondisi> then-else] : Jika kondisi true maka perintah-perintah yang ada di dalam then akan dijalankan, bila false maka perintah-perintah yang ada dalam else akan dijalankan.

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX



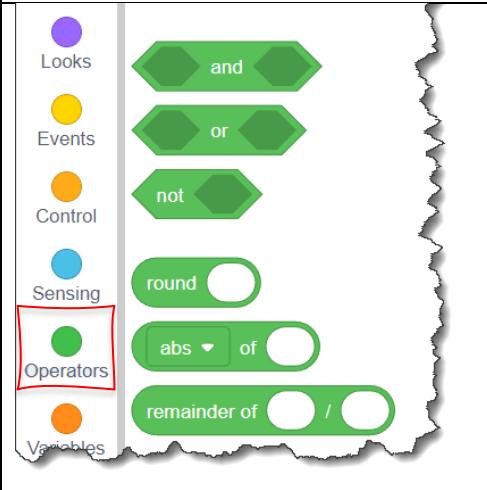
The image shows the VR VEX Code interface. On the left is a sidebar with categories: Drivetrain, Magnet, Looks, Events, Control, Sensing, Operators, Variables, My Blocks, Comments, and a Help section. The main area displays Scratch-style blocks for programming. The visible blocks include:

- [If <kondisi> then-<perintah> else if <kondisi> then-<perintah> else-<perintah>]
- [Wait until <kondisi>]
- [Repeat until <kondisi>]
- [While <kondisi>]
- [Break]
- [Stop project]

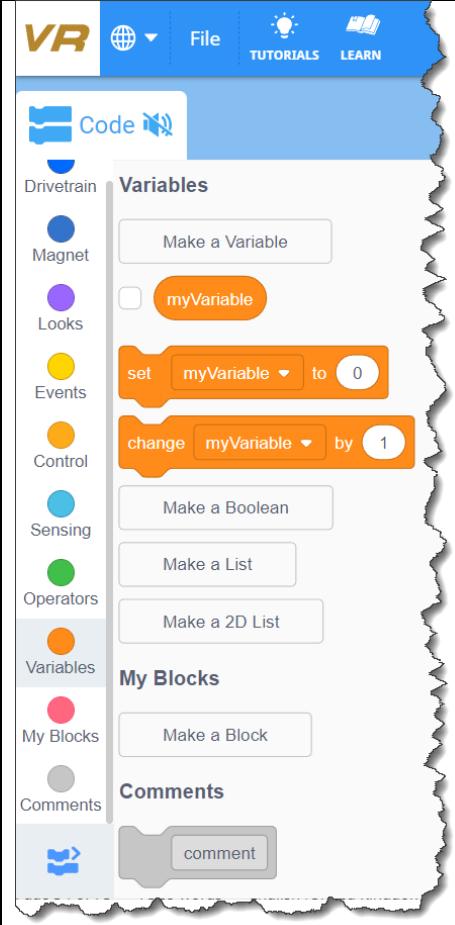
Explanations for the blocks:

- [If <kondisi> then-<perintah> else if <kondisi> then-<perintah> else-<perintah>] : Jika kondisi ke-1 true maka perintah yang ada dalam then ke-1 akan dijalankan. Jika kondisi ke-2 true maka perintah yang ada dalam then ke-2 akan dijalankan. Jika tidak maka perintah yang ada dalam else akan dijalankan.
- [Wait until <kondisi>] : menunggu sampai kondisi bernilai true
- [Repeat until <kondisi>] : mengulang perintah yang ada dalam repeat, sampai kondisi terpenuhi atau sampai kondisi bernilai true
- [While <kondisi>] : mengulang perintah yang ada dalam while, selama bernilai true
- [Break] : keluar dari pengulangan [forever]
- [Stop project] : mengakhiri project

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

	<p>Operators</p> <p>Operator adalah simbol yang digunakan dalam program untuk melakukan suatu operasi seperti operasi aritmatika (+, -, *, /), operasi perbandingan (>, <, =) dan operator logika (and, or, not). Nilai yang dioperasikan oleh operator disebut sebagai operand.</p> <p>Operator Aritmatika :</p> <ul style="list-style-type: none"> [] + [] : Operator + digunakan untuk menjumlahkan nilai ke-1 dan nilai ke-2 [] - [] : Operator - digunakan untuk mengurangi nilai ke-1 dengan nilai ke-2 [] * [] : Operator * digunakan untuk mengalikan nilai ke-1 dengan nilai ke-2 [] / [] : Operator / digunakan untuk membagi nilai ke-1 dengan nilai ke-2 <p>Pick random 1 to 10 : Mengembalikan sebuah nilai acak dari bilangan bulat antara 1 sampai dengan 10</p> <p>Operator Perbandingan :</p> <ul style="list-style-type: none"> [] > [] : Mengembalikan nilai True, jika nilai ke-1 lebih besar nilai ke-2, dan nilai False jika sebaliknya [] < [] : Mengembalikan nilai True, jika nilai ke-1 lebih kecil nilai ke-2, dan nilai False jika sebaliknya [] = [] : Mengembalikan nilai True, jika nilai ke-1 sama dengan nilai ke-2, dan nilai False jika sebaliknya
	<p>Operator Logika :</p> <ul style="list-style-type: none"> [] and [] : Mengembalikan nilai True, jika kedua nilai True [] or [] : Mengembalikan nilai True, jika salah satu nilai True not [] : Mengembalikan nilai kebalikannya <p>Fungsi Matematika :</p> <ul style="list-style-type: none"> round [] : fungsi round, membulatkan nilai abs of [] : fungsi abs, membuat nilai absolut remainder of [] / [] : fungsi remainder, sisa bagi nilai ke-1 dibagi nilai ke-2

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX



The screenshot shows the VEXcode software interface. The top menu bar includes 'File', 'TUTORIALS', and 'LEARN'. The left sidebar has categories: Drivetrain, Magnet, Looks, Events, Control, Sensing, Operators, Variables (which is selected), My Blocks, Comments, and a 'comment' block. The main workspace displays the 'Variables' block palette. It includes a 'Make a Variable' button, a variable named 'myVariable' (orange oval), a 'set myVariable to 0' block (orange rectangle), a 'change myVariable by 1' block (orange rectangle), and buttons for 'Make a Boolean', 'Make a List', 'Make a 2D List', 'My Blocks', 'Comments', and 'comment'.

Variables
Variabel digunakan untuk menampung nilai.

Tombol [Make a Variable] : Digunakan untuk membuat sebuah variabel

[Set my Variable to 0] : mengisi myVariable dengan nilai 0

[Change myVariable by 1] : mengubah nilai myVariable bertambah 1. Apabila nilai myVariable sebelumnya bernilai 5, maka setelah [Change myVariable by 1] nilainya menjadi $5 + 1$ atau 6

[Make a Boolean] : membuat sebuah variabel bertipe boolean (nilainya True atau False)

[Make a Boolean] : membuat sebuah variabel bertipe boolean (nilainya True atau False)

[Make a List] : membuat sebuah variabel bertipe list, panjang list antara 1 sd 20. Sebuah set list dapat menerima nilai desimal, integer dan blok berupa numerik. List ini serupa array 1 dimensi.

[Make a 2D List] : membuat sebuah variabel bertipe list 2D, baris list antara 1 sd 10 dan kolom list antara 1 sd 10. Sebuah set list dapat menerima nilai desimal, integer dan blok berupa numerik. List 2D serupa matriks.

C. Algoritma Sequence (sekuensial)

Algoritma sekuensial adalah langkah-langkah yang dilakukan secara berurutan sesuai dengan urutan penulisannya. Struktur ini merupakan struktur yang paling sering dilakukan dan mudah dipahami.

Contoh:

Algoritma sekuensial yang memiliki lima baris perintah, yaitu p1, p2, p3, p4 dan p5, maka semua perintah akan dilakukan secara berurutan mulai dari perintah p1 sampai p5.

D. Tugas T02a-Buat Kotak 4 buah

100 points – Kerjakan dalam 10 menit

Robot menggambar 4 kotak, menggunakan pensil warna merah, biru, hijau, hitam

Link : <https://www.youtube.com/watch?v=DynAn6elxz8>

Algoritma dalam bentuk rangkaian berurutan (sequence) :

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

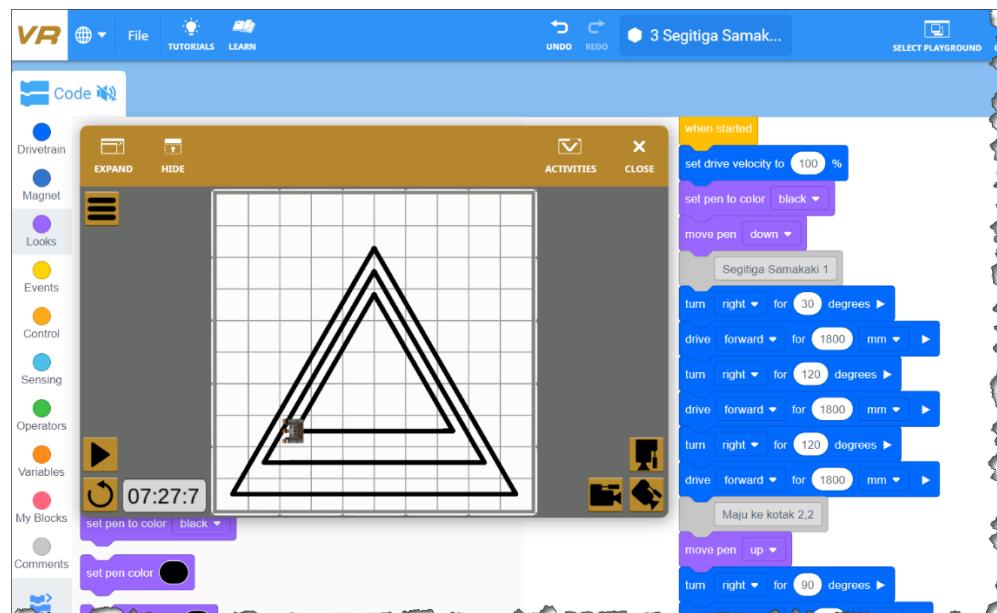
1. Robot menggambar kotak pertama dari 1-10, 10-100, 100-91, 91-1
2. Kemudian robot bergerak tanpa menggambar ke angka 12
3. Robot menggambar kotak ke-2 dari 12-19, 19-89, 89- 82, 82-12
4. Kemudian robot bergerak tanpa menggambar ke angka 23
5. Robot menggambar kotak ke-3 dimulai dari angka 23
6. Kemudian robot bergerak tanpa menggambar ke angka 34
7. Robot menggambar kotak ke-4 dimulai dari angka 34

Kumpulkan programnya diberi nama Kotak 4 buah - <Nama Anda> dan capture hasil akhirnya.

E. Tugas T02b-Buat Segitiga 3 buah

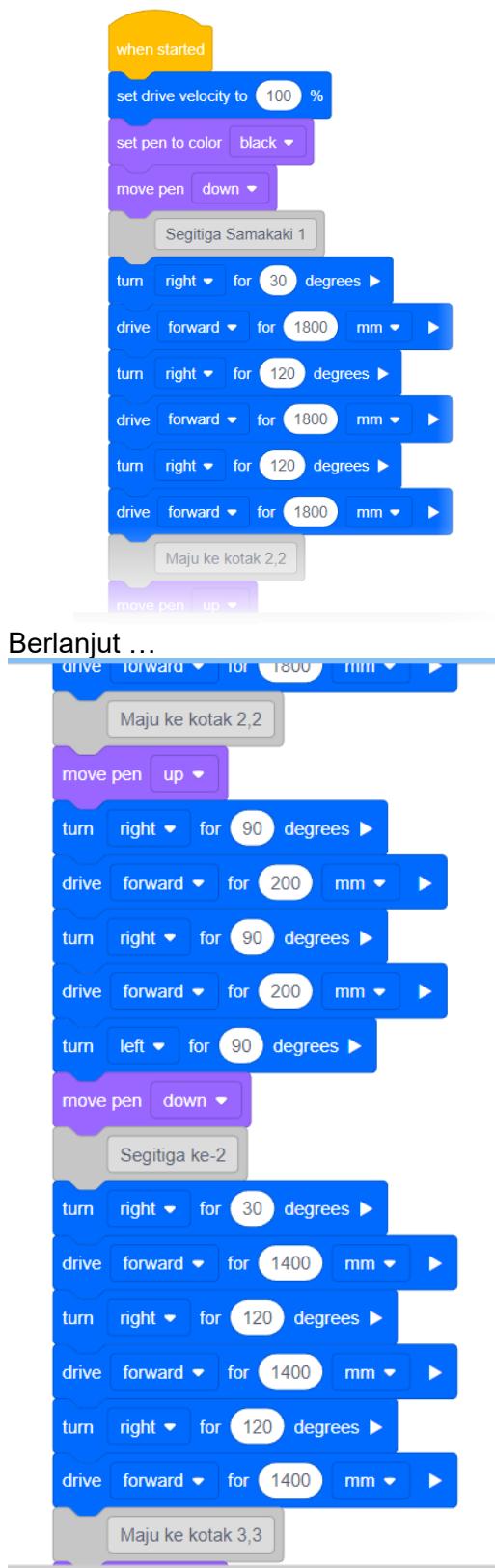
100 points – Kerjakan dalam 10 menit
Robot menggambar 3 segitiga dengan ukuran berbeda (1800mm, 1400mm dan 1000mm), menggunakan pensil warna hitam.
Sudut dalam Segitiga Sama sisi adalah 60 derajat.

Kumpulkan programnya diberi nama Segitiga 4 buah - <Nama Anda> dan capture hasil akhirnya.



Gambar 18 Tiga buah Segitiga Samakaki

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX



Dilanjutkan ke sebelah kanan ...

Mulai

Atur kecepatan robot 100%

Atur pena warna hitam

Turunkan pena (siap menggambar)

Komentar : Segitiga samakaki ke-1 di kotak 1,1

Belok kanan 30 derajat

Maju 1800 mm

Belok kanan 120 derajat

Maju 1800 mm

Belok kanan 120 derajat

Maju 1800 mm

Komentar : Maju ke kotak 2,2

Siap untuk menggambar segitiga ke-2



F. Tugas T02c-Buat Segienam 2 buah

100 points – Kerjakan dalam 10 menit

Robot menggambar segienam dengan ukuran berbeda, menggunakan pensil warna biru

Kumpulkan programnya diberi nama Segienam 2 buah - <Nama Anda> dan capture hasil akhirnya.

BAB 3

PENGULANGAN (ALGORITMA LOOPS)

Pelajaran 3 PENGULANGAN (ALGORITMA LOOPS)	Target Pelajaran <ol style="list-style-type: none"> Mengenal perintah pengulangan REPEAT dan FOREVER Membuat program menggunakan pengulangan. Gunakan playground Number Grid Map Mendemonstrasikan program robot menggunakan loops
	Deskripsi Pelajaran Siswa diminta untuk melihat perbedaan algoritma sequence dan loops, dan menyebutkan keuntungan penggunaan loops. Siswa membuat program menggunakan pengulangan
Durasi Belajar 45-60 menit	Bahan Pelajaran Lembar Kerja 3.1 – Jenis-jenis Loops Lembar Kerja 3.2 – Robot menggambar Benteng Kotak 3 Lapis Lembar Kerja 3.3 – Robot menggambar Segitiga, Segiempat, Segilima, Segienam, Segidelapan, Segiduabelas

A. Algoritma Pengulangan (Looping/Loops)

Kegiatan yang sama yang dilakukan secara berulang, dapat ditulis menggunakan algoritma sekuensial, hanya saja menjadi tidak efisien. Misal dalam algoritma sekuensial untuk menggambar kotak segi empat, sebagai berikut :

- 1. Maju 1000 mm**
- 2. Belok 90 derajat**
- Maju 1000 mm
- Belok 90 derajat
- Maju 1000 mm
- Belok 90 derajat
- Maju 1000 mm
- Belok 90 derajat

Perhatikan perintah baris 1 dan baris 2, sama dengan baris 3 dan 4, sama dengan baris 5 dan 6, sama dengan baris 7 dan 8. Artinya baris 1 dan 2 diulang sebanyak 4 kali. Jika ditulis dalam algoritma sekuensial, maka tidak efisien. Oleh karena itu perintah yang berulang dapat menggunakan algoritma pengulangan (*looping/loops*). Perhatikan algoritmanya akan diubah menjadi algoritma pengulangan, sebagai berikut :

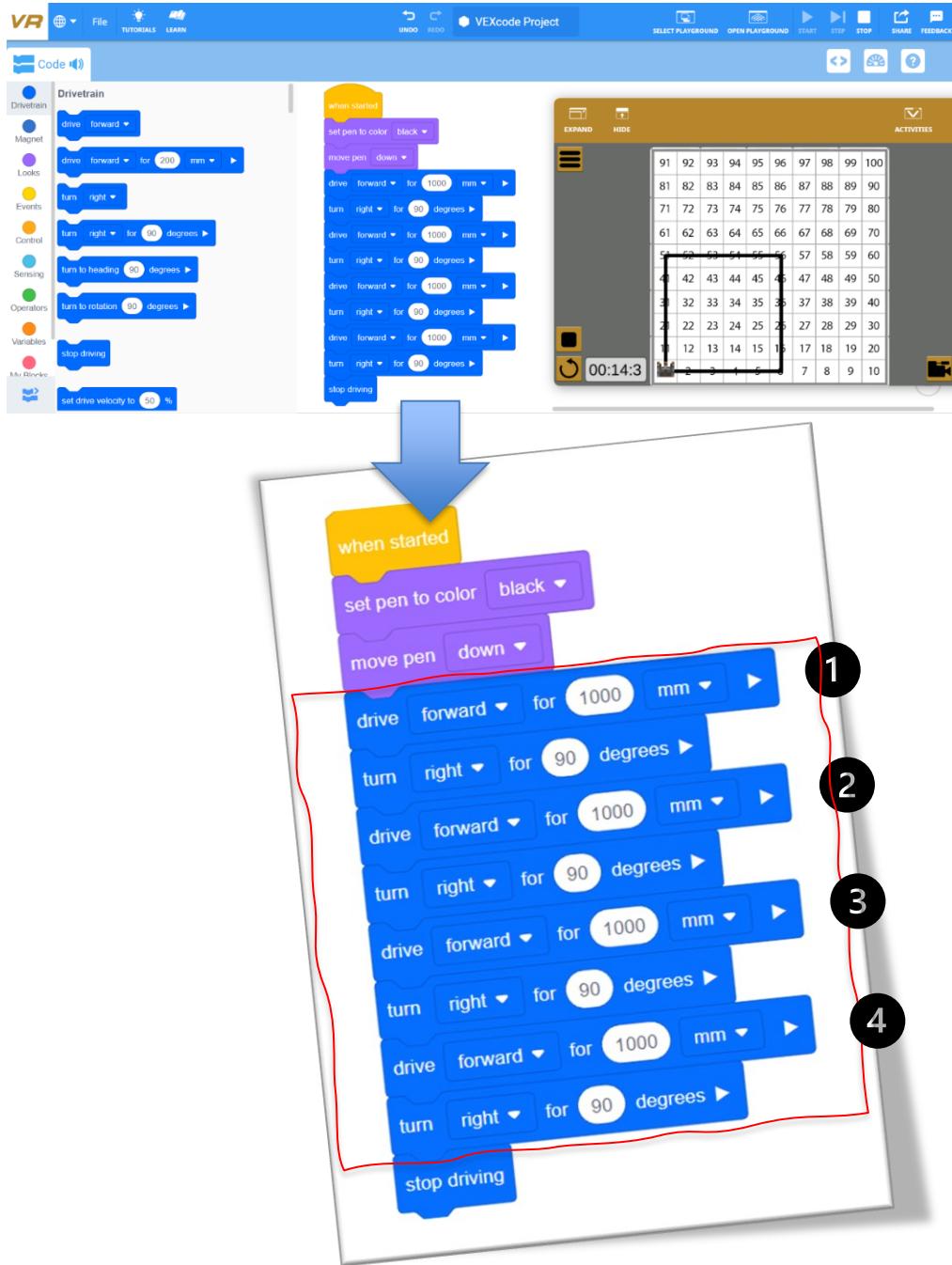
- Ulang 4 kali
- 2. Maju 1000 mm**
- 3. Belok 90 derajat**

Perintah baris 1, menyatakan bahwa perintah 2 dan 3 akan diulang sebanyak 4 kali, maka secara logika keduanya sama antara algoritma sekuensial dan algoritma pengulangan. Jika kita perhatikan algoritma pengulangan lebih ringkas, lebih mudah dibaca dan efisien. Struktur

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

pengulangan atau *looping* digunakan untuk menjalankan kegiatan yang dilakukan berulang-ulang.

Mari kita lihat penerapan dalam pemrograman robot VR Vex.

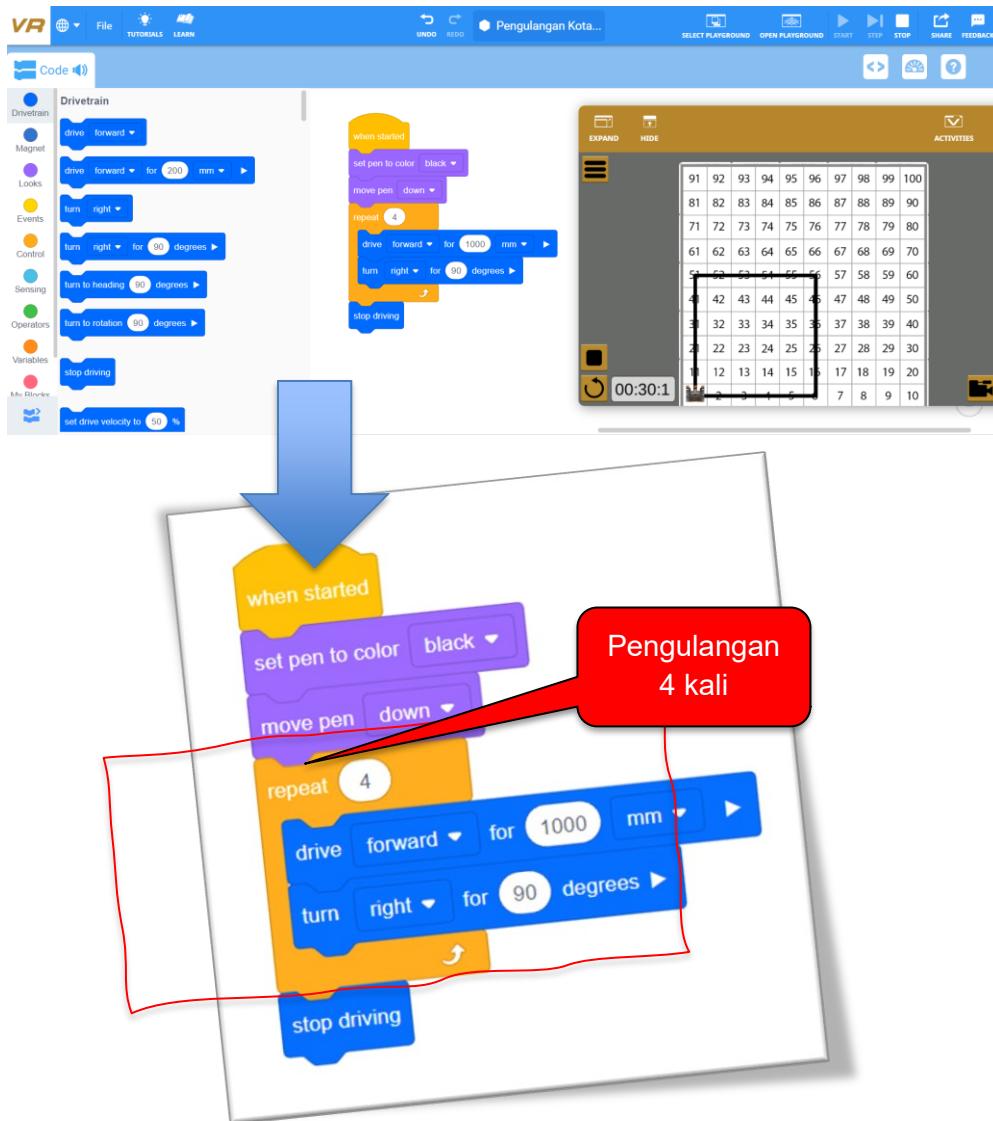


Gambar 19 Contoh program sekuensial untuk menggambar kotak 1000 mm

Pada Gambar 19 mulai baris ke-4 [**drive forward for 1000mm**] dan [**turn right for 90 degrees**] ditulis 4 kali supaya tergambar segiempat. Pada bagian akhir ditutup dengan [**stop driving**]. Bisa dibayangkan bagaimana bila robot diminta untuk menggambar

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

segisepuluh, maka akan ditulis 10 kali perintah yang sama. Bagaimana jika menggambar segiseratus?



Gambar 20 Contoh program pengulangan untuk menggambar kotak 1000 mm

Pada Gambar 20, untuk menggambar kotak atau segiempat menggunakan pengulangan yang hasilnya sama dengan program sekuensial. Terlihat perintah **[repeat 4]** membungkus 2 baris perintah **[drive forward for 1000mm]** dan **[turn right for 90 degrees]**, sehingga kedua perintah tersebut akan diulang sebanyak 4 kali. Ini membuat program lebih ringkas, lebih mudah dibaca dan efisien. Bagaimana apabila hendak menggambar segisepuluh? Cukup mengubah **[repeat 4]** menjadi **[repeat 10]**. Bagaimana dengan menggambar segiseratus?

B. T03a-Buat Kotak 5 buah (Gunakan Pengulangan Repeat)

100 poin, masing-masing 50 poin

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

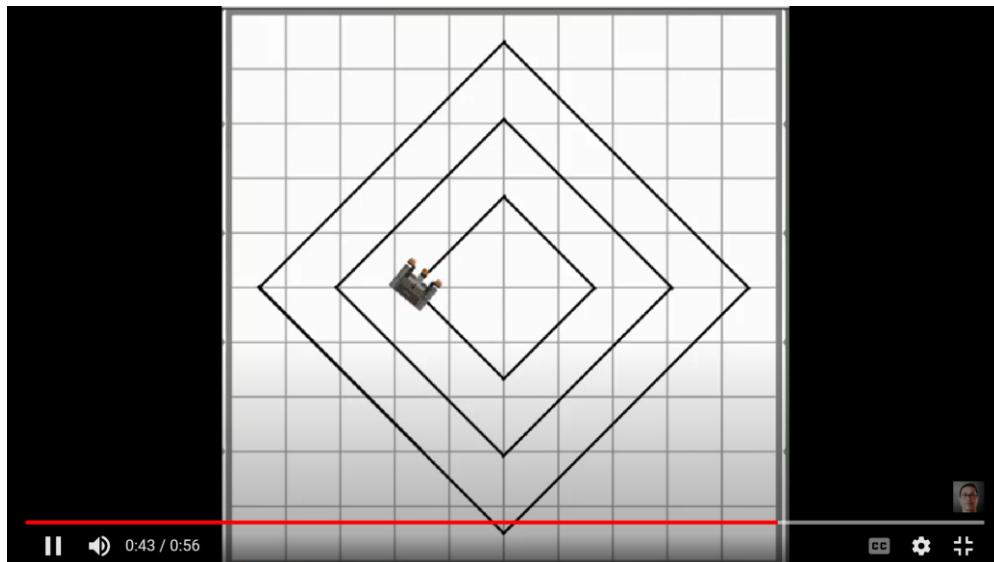
A. Robot menggambar 5 kotak, menggunakan pensil warna merah (Gunakan Repeat)

1. Robot menggambar kotak pertama dari 1-10, 10-100, 100-91, 91-1
2. Kemudian robot bergerak tanpa menggambar ke angka 12
3. Robot menggambar kotak ke-2 dari 12-19, 19-89, 89- 82, 82-12
4. Kemudian robot bergerak tanpa menggambar ke angka 23
5. Robot menggambar kotak ke-3 dimulai dari angka 23
6. Kemudian robot bergerak tanpa menggambar ke angka 34
7. Robot menggambar kotak ke-4 dimulai dari angka 34
8. Kemudian robot bergerak tanpa menggambar ke angka 45
9. Robot menggambar kotak ke-5 dimulai dari angka 45

Kumpulkan programnya diberi nama Kotak 5 buah - <Nama Anda> dan capture hasil akhirnya.

C. T03b-Buat Belah ketupat 3 buah (Gunakan Pengulangan Repeat)

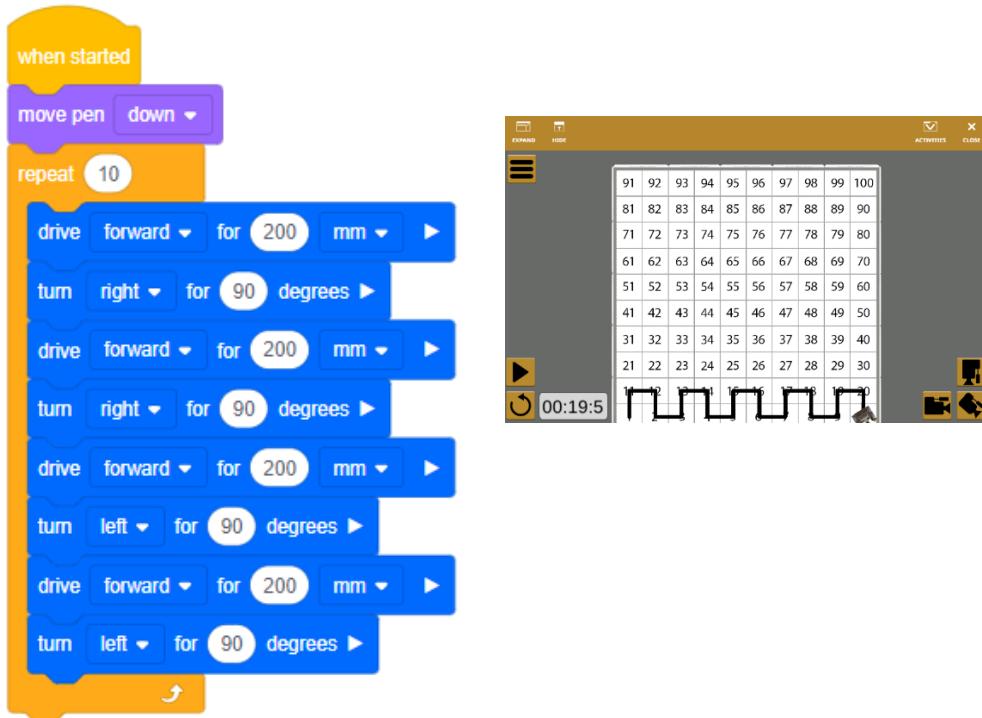
Belah Ketupat pertama sisi 1272mm, 872mm, 472mm. Lihat contoh (Link : <https://youtu.be/6CrXec9eWnA>)



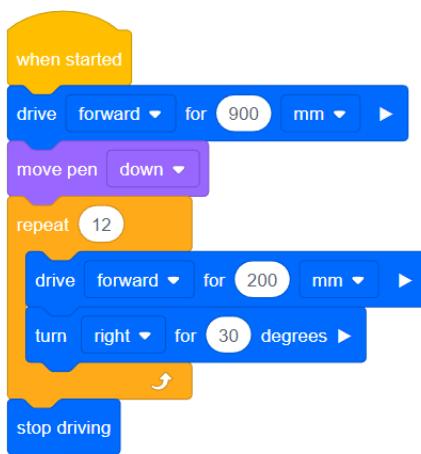
- a. Program tanpa pengulangan
- b. Program dengan pengulangan (looping)
- c. Seperti kasus ke-2, hanya warna garis pada setiap kotak : garis miring warna biru dan lainnya berwarna merah.

D. Kuis

1. Coba bayangkan proyek ini, tanpa perlu menjalankan programnya. Jawaban yang benar adalah (bisa lebih dari satu)



- a) Pengulangan Repeat 10 kali kebanyakan, robot akan menabrak dinding atau jatuh
 - b) Pengulangan Repeat 5 kali, maka robot tidak akan menabrak dinding
 - c) Semua angka 200 pada [drive forward for] diganti 100, maka robot tidak akan menabrak dinding
 - d) Pengulangan Repeat 4 kali, maka robot akan berhenti di angka 9
2. Coba bayangkan proyek ini, tanpa perlu menjalankan programnya. Jawaban yang benar adalah (bisa lebih dari satu)



- a. Robot akan menggambar setengah lingkaran
- b. Robot akan menggambar segi-duabelas
- c. Robot akan menggambar huruf p
- d. Robot akan menggambar segi-duabelas terbuka
- e. Pengulangan 12 kali untuk belok 30 derajat maka akan membentuk sudut penuh 360 derajat

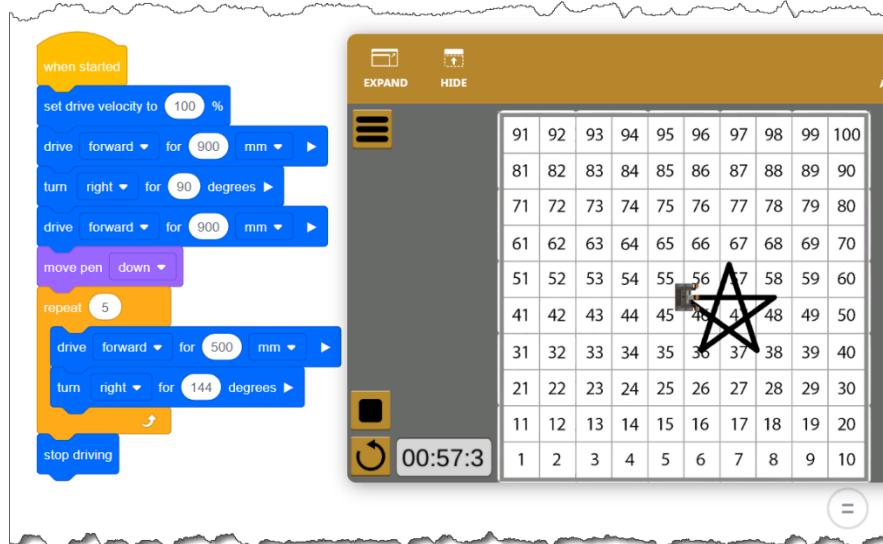
PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

3. Coba bayangkan proyek ini, tanpa perlu menjalankan programnya. Jawaban yang benar adalah (bisa lebih dari satu)



- a. Bila angka [Turn Right for 30 degrees], diganti 60, maka robot akan menggambar segi-duabelas sebanyak 2 kali
- b. Bila angka [Turn Right for 30 degrees], diganti Turn Left for 60, maka robot akan menggambar segi-duabelas sebanyak 2 kali
- c. Bila angka [Repeat 12] diganti 10, maka robot akan menggambar segi-sepuluh
- d. Bila angka [Repeat 12] diganti 10 dan angka [Turn Right for 30 degrees] diganti 36, maka robot akan menggambar segi-sepuluh

4. Sebuah proyek membuat bintang menggunakan pengulangan Repeat 5, dengan panjang sisi 500 mm. Mana pernyataan yang benar dari pilihan berikut :



- a. Supaya robot tidak menabrak maka sisi 500 mm, dapat diperbesar sampai 900 mm
- b. Jika posisi [drive forwar for 500mm] ditukar dengan [turn right for 144 degrees] tetap akan menggambar bintang juga
- c. Jika angka [turn right for 144 degrees] diganti -144 tetap akan menggambar bintang juga
- d. Jika angka [drive forwar for 500mm] diganti -500 tetap akan menggambar bintang juga

BAB 4

TIPE DATA, VARIABEL, OPERATOR DAN FUNGSI

Pelajaran 4 TIPE DATA, VARIABEL, OPERATOR, DAN FUNGSI	<p>Target Pelajaran</p> <ol style="list-style-type: none"> 1. Mengenal berbagai tipe data 2. Mendefinisikan dan menggunakan variabel, list dan 2D list dalam program robot 3. Mengenal operator aritmatika (+, -, *, /), penugasan/assignment (=), penambahan/increment dan pengurangan/decrement, pembanding (>, <, =), operator logika (and, or, not) dan fungsi (abs, floor, ceiling, sqrt, sin, cos, tan, asin, acos, atan, ln, log, e^x, 10^x) 4. Membuat program robot menggunakan loops dan variabel. Gunakan playground Number Grid Map
	<p>Deskripsi Pelajaran</p> <p>Siswa diminta membedakan jenis variabel dan tipe data. Siswa akan mendefinisikan variabel dan digunakan dalam program robot menggambar suatu bentuk dasar</p>
Durasi Belajar 45-60 menit	<p>Bahan Pelajaran</p> <p>Lembar Kerja 4.1 - Tipe Data Lembar Kerja 4.2 – Jenis operator Lembar Kerja 4.3 – Robot menggambar 5 buah kotak dengan ukuran berbeda, menggunakan loops dan variabel Lembar Kerja 4.4 – Robot menggambar kotak spiral menggunakan loops dan variabel</p>

A. Mengenal berbagai tipe data

Tipe data dalam pemrograman mengacu pada jenis nilai yang dapat disimpan dalam variabel atau digunakan dalam operasi matematika atau logika. Beberapa tipe data umum dalam pemrograman meliputi:

- a) Integer: Tipe data integer digunakan untuk menyimpan bilangan bulat. Contohnya adalah 1, 2, 3, -4, -5, dan seterusnya.
- b) Floating-point atau desimal: Tipe data floating-point digunakan untuk menyimpan bilangan pecahan. Contohnya adalah 3,14, 2,5, -1,25, dan seterusnya.
- c) Boolean: Tipe data boolean digunakan untuk nilai yang hanya dapat berupa benar atau salah (true atau false).
- d) Karakter: Tipe data karakter digunakan untuk menyimpan karakter tunggal, seperti 'a', 'b', 'c', dan seterusnya. Tipe data ini belum tersedia dalam VR.Vex.
- e) String: Tipe data string digunakan untuk menyimpan kumpulan karakter. Contohnya adalah "hello", "world", "1234", dan seterusnya. Tipe data ini belum tersedia dalam VR.Vex.
- f) Array: Tipe data array digunakan untuk menyimpan kumpulan nilai yang sama jenisnya, seperti kumpulan bilangan bulat, bilangan pecahan, karakter, dan

seterusnya. Ada 2 jenis array yang digunakan dalam VR.Vex, yaitu List atau array 1 dimensi dan 2D List atau array 2 dimensi dikenal dengan nama matriks.

Tipe data dalam pemrograman bergantung pada bahasa pemrograman yang digunakan dan dapat bervariasi antara bahasa pemrograman. Penting untuk memahami tipe data yang tersedia dan bagaimana mereka dapat digunakan untuk membangun program yang efektif dan efisien.

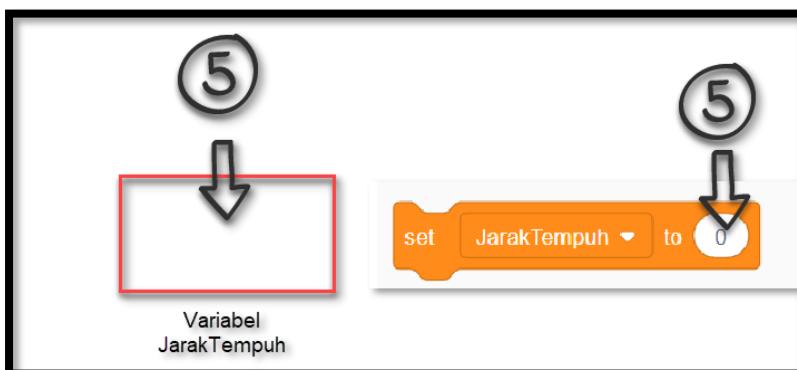
B. Variabel

Variabel dalam pemrograman digunakan untuk menyimpan nilai atau data tertentu dalam program. Fungsi variabel adalah memungkinkan programmer untuk menyimpan, mengakses, dan memanipulasi data dalam program dengan cara yang mudah dan efisien. Dalam pemrograman, variabel dapat didefinisikan dengan tipe data tertentu, seperti integer, float, string, boolean, dan sebagainya. Tipe data variabel menentukan jenis data yang dapat disimpan dalam variabel.

Programmer dapat membuat program yang lebih dinamis dan fleksibel menggunakan variabel, karena nilai atau data yang disimpan dalam variabel dapat berubah selama program dijalankan. Hal ini memungkinkan programmer untuk membuat program yang dapat menerima masukan atau input dari pengguna atau lingkungan, memprosesnya dan memberikan luaran atau output, tergantung proses yang dilakukan dalam program.

Variabel bertipe numerik digunakan untuk menyimpan nilai integer, desimal maupun blok bernilai numerik. seperti pada Gambar 21. Berikut contoh penggunaannya :

- variabel JarakTempuh bertipe data numerik dibuat, kemudian diisi dengan data 5, maka nilai dari JarakTempuh adalah 5
- variabel luasPersegi untuk menampung hasil perhitungan panjang * lebar
- variabel bilanganAcak untuk menampung nilai dari blok perintah [pick random 1 to 10]
- variabel akarKuadrat untuk menampung nilai dari blok perintah [sqrt of 16]



Gambar 21 Ilustrasi Variabel Numerik dan Kode Blok Pengisian Variabel

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX



Gambar 22 Ilustrasi Variabel Boolean dan Kode Blok Pengisian Variabel Boolean

Variabel bertipe boolean digunakan untuk menyimpan nilai true atau false, seperti pada Gambar 22. Berikut contoh penggunaannya :

- program untuk mengecek apakah sebuah bilangan prima atau tidak? Variabel boolean dapat digunakan untuk menyimpan nilai apakah bilangan tersebut prima atau tidak.
- program yang mengatur suatu fitur, variabel boolean dapat digunakan untuk menandai apakah fitur tersebut aktif atau tidak.
- program yang mendeteksi warna gelap, variabel boolean dapat digunakan untuk menandai apakah warna tersebut gelap atau tidak.



Gambar 23 Ilustrasi Variabel List (Array) dan Kode Blok Pengisian Variabel List

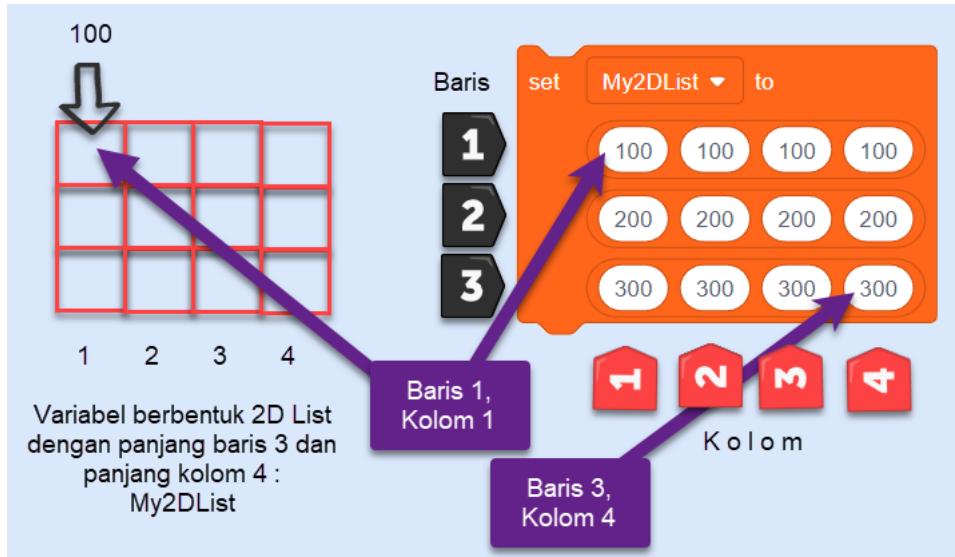
Variabel list atau dikenal dengan array 1 demesi, digunakan untuk menyimpan kumpulan nilai dalam satu variabel, setiap datanya ditandai dengan indeks atau item ke berapa, dimulai dengan item ke-1. Dalam pemrograman secara umum, penggunaan variabel array sangat penting karena memungkinkan pengelolaan sekumpulan data yang lebih efektif dan efisien. Dalam VR.Vex array diberi nama List dan maksimum menampung 20 nilai, seperti pada Gambar 23. Berikut contoh penggunaannya :

- program untuk menampung myList dengan panjang 5 = {1000, 800, 600, 400, 200 }
- program untuk menampung nilaiSiswa dengan panjang 3 = { 80, 90, 70 }
- program untuk menampung jarakTempuh dengan panjang 4 = { 120, 100, 300, 500 }
- program untuk menampung sudutBelok dengan panjang 3 = { 90, 45, 45 }

Variabel 2D List atau dikenal dengan array 2 demesi, digunakan untuk menyimpan kumpulan nilai dalam satu variabel, setiap datanya ditandai dengan baris dan kolom . Dalam VR.Vex diberi nama 2D List atau dikenal dengan matriks dan maksimum menampung 10 baris dan 10 kolom, seperti pada Gambar 24. Berikut contoh penggunaannya :

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

- program untuk menampung my2DList dengan jumlah 3 baris dan 4 kolom = {{100, 100, 100, 100}, {200, 200, 200, 200}, {300, 300, 300, 300} }
- program untuk menampung koordinatBenda dengan jumlah 4 baris dan 2 kolom = { {80, 190}, {15, 70}, {30,40}, {-50,70} }



Gambar 24 Ilustrasi Variabel 2D List (Matriks) dan Kode Blok Pengisian Variabel 2DList

Variables

Variabel digunakan untuk menampung nilai.

Tombol [Make a Variable] : Digunakan untuk membuat sebuah variabel

[Set my Variable to 0] : mengisi myVariable dengan nilai 0

[Change myVariable by 1] : mengubah nilai myVariable bertambah 1. Apabila nilai myVariable sebelumnya bernilai 5, maka setelah [Change myVariable by 1] nilainya menjadi $5 + 1$ atau 6

[Make a Boolean] : membuat sebuah variabel bertipe boolean (nilainya True atau False)

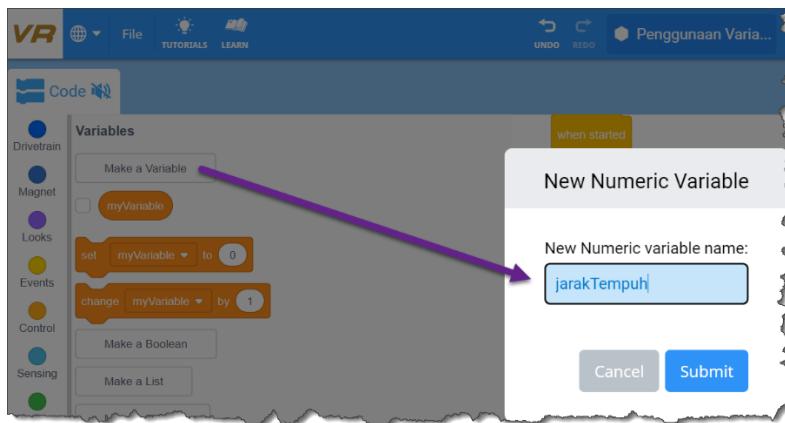
[Make a List] : membuat sebuah variabel bertipe list, panjang list antara 1 sd 20. Sebuah set list dapat menerima nilai desimal, integer dan blok berupa numerik. List ini serupa array 1 dimensi.

[Make a 2D List] : membuat sebuah variabel bertipe list 2D, baris list antara 1 sd 20 dan kolom list antara 1 sd 20. Sebuah set list dapat menerima nilai desimal, integer dan blok berupa numerik. List 2D serupa matriks.

C. Mendefinisikan, menggunakan Variabel, List dan 2D List

Sebelum digunakan variabel perlu didefinisikan (deklarasi), misal akan didefinisikan variabel yang bernama jarakTempuh, seperti pada Gambar 25. Beberapa aturan penamaan variabel yang perlu diperhatikan yaitu :

- Namanya unik, tidak boleh menggunakan nama variabel yang sudah ada
- Diawali karakter tidak boleh angka
- Tidak menggunakan spasi atau tandabaca atau operator
- Tidak menggunakan kata kunci (keyword)

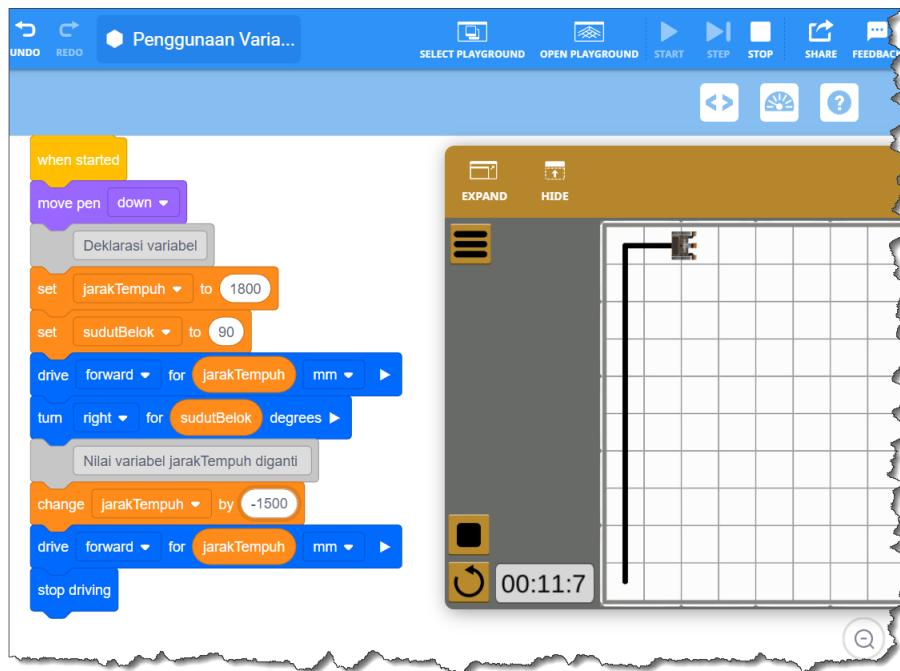


Gambar 25 Mendefinisikan sebuah variabel

Berikut contoh Proyek Penggunaan Variabel (Gambar 26), setelah dideklarasikan variabel numerik jarakTempuh dan sudutBelok.

- Baris 01, Robot siap mulai beroperasi
- Baris 02, pena hitam diturunkan, robot siap menggambar
- Baris 03 dan 08, merupakan komentar untuk memberikan keterangan dan tidak akan diproses
- Pada baris 04 dan 05, variabel diisi dengan angkat, jarakTempuh=1800 dan sudutBelok=90.
- Pada baris 06, variabel jarakTempuh digunakan dalam blok perintah [drive forward for **jarakTempuh** mm]
- Pada baris 07, variabel sudutBelok digunakan dalam blok perintah [turn right for **sudutBelok** degrees]
- Pada baris 09, variabel jarakTempuh diubah menjadi jarakTempuh = jarakTempuh – 1500, maka nilai jarakTempuh yang baru adalah 1800 – 1500 yaitu 300
- Pada baris 10, variabel jarakTempuh dengan nilai terbaru yaitu 300 digunakan dalam blok perintah [drive forward for **jarakTempuh** mm]
- Baris 11, robot berhenti

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

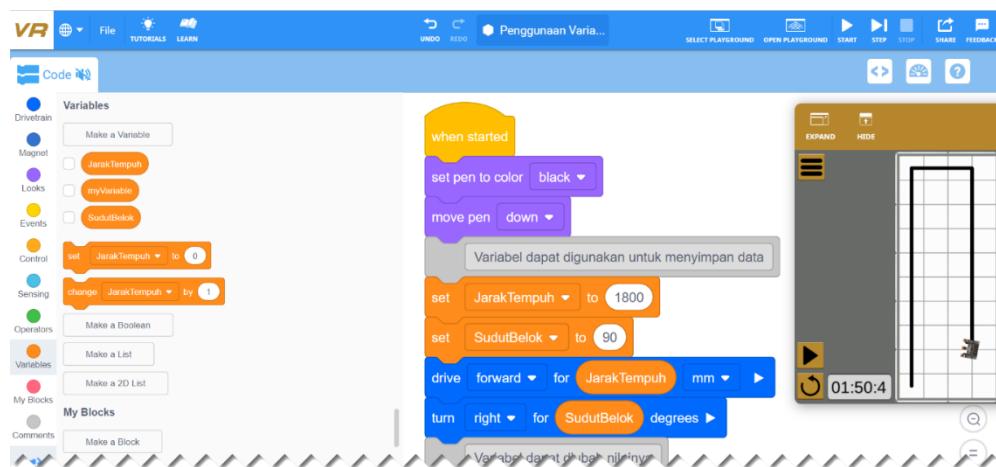


Gambar 26 Penggunaan Variabel Numerik : jarakTempuh dan sudutBelok

Saat program dijalankan, maka robot akan siap menggambar menggunakan pena hitamnya, kemudian bergerak maju 1800 mm, belok 90 derajat, dan bergerak maju 300 mm. Hasil akhir tampak pada Gambar 26, bagian kanan.

Catatan : Dalam pemrograman blok perintah [Change jarakTempuh by -1500] sering digunakan sebagai counter atau pencacah, artinya nilai variabel tersebut dapat ditambah atau dikurangi sebuah bilangan. Pada kasus ini variabel jarakTempuh diganti menjadi jarakTempuh – 1500, maka jarakTempuh yang baru diganti menjadi jarakTempuh yang lama dikurangi 1500, sehingga nilai jarakTempuh setelah perintah ini menjadi 300.

Pada Gambar 27, dapatkah Anda menambahkan potongan progam ini sehingga robot melanjutkan bergerak maju dan kemudian belok ke kanan? Mudah bukan?

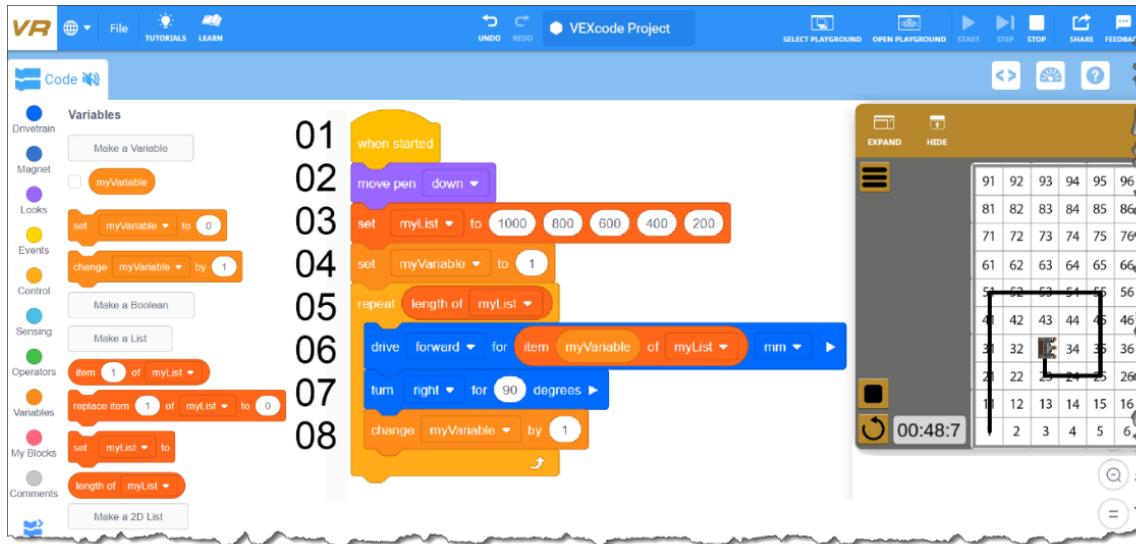


Gambar 27 Penggunaan Variabel Numerik : jarakTempuh dan sudutBelok lanjutan

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

Variabel List

Berikut contoh Proyek Penggunaan List (Gambar 28), setelah dideklarasikan List bernama myList dengan panjang 5, artinya nilai yang dapat ditampung maksimum 5 buah.



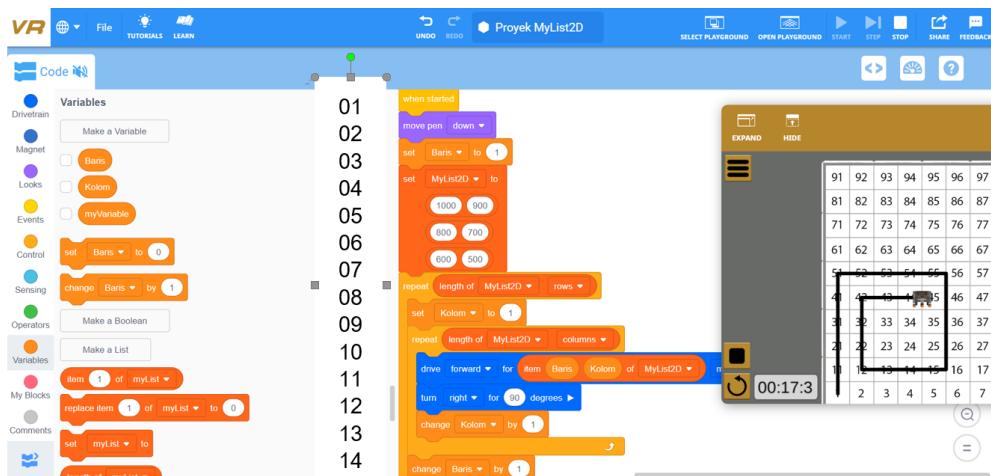
Gambar 28 Penggunaan List atau Array : myList

- Baris 01, Robot siap mulai beroperasi
- Baris 02, pena hitam diturunkan, robot siap menggambar
- Pada baris 03, myList diisi dengan 5 buah data yaitu {1000, 800, 600, 400, 200}
- Pada baris 04, variabel myVariable diisi dengan 1, variabel ini digunakan untuk mencatat posisi list yang akan digunakan nilainya.
- Pada baris 05, pengulangan Repeat sebanyak [length of myList], karena panjang myList adalah 5, maka repeat sebanyak 5 kali
- Pada baris 06, robot akan maju sejauh [item myVariable of myList]. Nilai myVariable = 1, sehingga item ke-1 dari myList adalah 1000. Robot maju sejauh 1000 mm
- Pada baris 07, robot belok 90 derajat
- Pada baris 08, nilai myVariable bertambah 1, menjadi 2
- Kemudian mengulang baris 05 [repeat] untuk pengulangan ke-2
- Proses baris 06, 07, 08 dan seterusnya sampai pengulangan [repeat] dilakukan 5 kali
- Setelah proses pengulangan selesai, robob berhenti

Variabel 2D List (List 2 dimensi atau matriks)

Berikut contoh Proyek Penggunaan 2D List (Gambar 28), setelah dideklarasikan 2D List bernama myList2D dengan panjang baris=3 dan panjang kolom=2, artinya nilai yang dapat ditampung maksimum $3 \times 2 = 6$ buah.

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

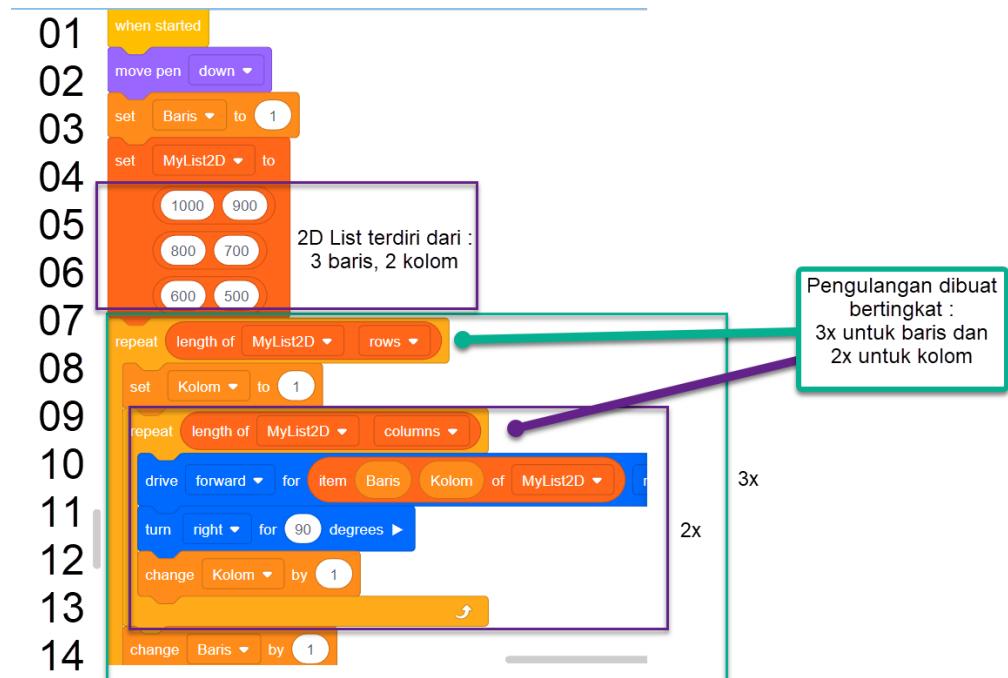


Gambar 29 Penggunaan 2D List atau matriks : myList2D

- Baris 01, Robot siap mulai beroperasi
- Baris 02, pena hitam diturunkan, robot siap menggambar
- Pada baris 03, variabel Baris diisi dengan 1
- Pada baris 04-07, myList2D diisi dengan 6 buah data yaitu {{1000, 900}, {800, 700}, {500,400}}
- Pada baris 08, pengulangan Repeat sebanyak [length of myList2D rows], karena panjang baris myList2D adalah 3, maka repeat sebanyak 3 kali
- Pada baris 09, variabel Kolom diisi dengan 1
- Pada baris 10, pengulangan Repeat sebanyak [length of myList2D coloms], karena panjang kolom myList2D adalah 2, maka repeat sebanyak 2 kali
- Pada baris 11, robot akan maju sejauh [item baris, kolom of myList2D]. Nilai baris = 1 dan kolom = 1, sehingga item baris, kolom dari myList2D adalah 1000. Robot maju sejauh 1000 mm
- Pada baris 12, robot belok 90 derajat
- Pada baris 13, nilai kolom bertambah 1, menjadi 2
- Kemudian mengulang baris 10 [Repeat [length of myList2D coloms]], untuk pengulangan ke-2
- Proses baris 11, 12,13
- Pada baris 14, nilai kolom bertambah 1, menjadi 2
- Kemudian mengulang baris 08 [Repeat [length of myList2D rows]], untuk baris = 2. Proses berlanjut, sampai 3 baris x 2 kolom = 6 pengulangan

Perhatikan pola pengulangan untuk pemrosesan 2D List, karena memiliki baris dan kolom, maka pengulangan dibuat bertingkat (bersarang) atau pengulangan dalam pengulangan seperti penjelasan pada Gambar 30

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX



Gambar 30 Pengulangan dalam pengulangan untuk memproses 2D List

D. Mengenal operasi aritmatika, perbandingan, logika dan fungsi-fungsi

Dalam VR.Vex tersedia beberapa operasi seperti aritmatika, operasi perbandingan, operasi logika dan beberapa fungsi yang mempermudah dalam pemecahan masalah numerik, seperti tertera pada Gambar 31.

Operators

Operator adalah simbol yang digunakan dalam program untuk melakukan suatu operasi seperti operasi aritmatika ($+$, $-$, $*$, $/$), operasi perbandingan ($>$, $<$, $=$) dan operator logika (and, or, not). Nilai yang dioperasikan oleh operator disebut sebagai operand.

Operator Aritmatika :

- $[] + []$: Operator $+$ digunakan untuk menjumlahkan nilai ke-1 dan nilai ke-2
- $[] - []$: Operator $-$ digunakan untuk mengurangi nilai ke-1 dengan nilai ke-2
- $[] * []$: Operator $*$ digunakan untuk mengalikan nilai ke-1 dengan nilai ke-2
- $[] / []$: Operator $/$ digunakan untuk membagi nilai ke-1 dengan nilai ke-2

Pick random 1 to 10 : Mengembalikan sebuah nilai acak dari bilangan bulat antara 1 sampai dengan 10

Operator Pembanding :

- $[] > []$: Mengembalikan nilai True, jika nilai ke-1 lebih besar nilai ke-2, dan nilai False jika sebaliknya
- $[] < []$: Mengembalikan nilai True, jika nilai ke-1 lebih kecil nilai ke-2, dan nilai False jika sebaliknya
- $[] = []$: Mengembalikan nilai True, jika nilai ke-1 sama dengan nilai ke-2, dan nilai False jika sebaliknya

Operator Pembanding :

- $[] and []$: Mengembalikan nilai True, jika kedua nilai True
- $[] or []$: Mengembalikan nilai True, jika salah satu nilai True
- $not []$: Mengembalikan nilai kebalikannya

round [] : fungsi round, membulatkan nilai, jika nilai desimal lebih atau sama dengan 0.5 menjadi 1, jika dibawah 0.5 menjadi 0. Contoh $round(10.5) \rightarrow 11$, $round(10.4) \rightarrow 10$

abs of [] : fungsi abs, membuat nilai absolut

remainder of [] / [] : fungsi remainder, sisa bagi nilai ke-1 dibagi nilai ke-2

Gambar 31 Operators dalam VR.Vex

Operasi Aritmatika

Operasi aritmatika adalah proses matematika yang melibatkan penjumlahan, pengurangan, perkalian, pembagian, dan operasi lainnya untuk memanipulasi angka atau variabel numerik. Operasi aritmatika dapat digunakan dalam berbagai konteks, termasuk ilmu matematika, ilmu pengetahuan alam, teknologi, keuangan, dan sebagainya.

Beberapa contoh operasi aritmatika yang umum adalah:

- Penjumlahan: Menambahkan dua atau lebih angka atau variabel numerik.
Contohnya, $2 + 3 = 5$.
- Pengurangan: Mengurangkan satu angka atau variabel numerik dari yang lain.
Contohnya, $5 - 2 = 3$.

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

- c) Perkalian: Mengalikan dua atau lebih angka atau variabel numerik. Contohnya, $2 \times 3 = 6$.
- d) Pembagian: Membagi satu angka atau variabel numerik dengan angka atau variabel numerik lainnya. Contohnya, $6 \div 2 = 3$.
- e) Modulus: Menemukan sisa pembagian dari dua angka atau variabel numerik. Contohnya, $7 \% 3 = 1$, karena 7 dibagi dengan 3 menghasilkan sisa 1.
- f) Pangkat: Mengangkat suatu angka atau variabel numerik ke suatu pangkat tertentu. Contohnya, $2^3 = 8$, karena 2 dipangkatkan dengan 3 menghasilkan 8.
- g) Akar: Mencari akar dari suatu bilangan. Contoh: akar kuadrat dari $25 = 5$.
- h) Pembulatan: membulatkan suatu bilangan menjadi bilangan bulat tertentu atau dengan jumlah angka desimal tertentu. Contoh: pembulatan 3.6 menjadi 4.
- i) Faktorial: Menghitung hasil perkalian dari bilangan bulat positif dan semua bilangan bulat positif yang lebih kecil darinya. Contoh: $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$.
- j) Logaritma: Mencari eksponen yang diperlukan untuk menghasilkan suatu bilangan tertentu. Contoh: logaritma basis 2 dari $8 = 3$, karena $2^3 = 8$.
- k) Trigonometri: Menghitung hubungan antara sudut dan sisi segitiga. Contoh: $\sin(30) = 0.5$, karena $\sin(30)$ adalah rasio antara sisi segitiga yang bersebrang dengan sudut 30 dan sisi miring.

Dalam matematika, operasi aritmatika digunakan untuk menyelesaikan berbagai macam masalah seperti menghitung luas, volume, mengukur jarak, waktu, kecepatan, dan masih banyak lagi. Dalam pemrograman komputer, operasi aritmatika untuk menyelesaikan berbagai macam tugas, seperti penghitungan angka pada aplikasi keuangan, perhitungan dalam permainan, dan bahkan pada pembuatan algoritma kecerdasan buatan. Oleh karena itu, penting bagi kita untuk memahami dan menguasai operasi aritmatika dasar serta operasi aritmatika yang lebih kompleks.

Operasi Perbandingan

Operasi perbandingan adalah operasi dalam matematika dan pemrograman komputer yang digunakan untuk membandingkan dua buah bilangan atau nilai, dan menghasilkan nilai kebenaran (true atau false) sebagai hasil dari perbandingan tersebut.

Berikut ini adalah beberapa contoh operasi perbandingan yang sering digunakan:

- a) Lebih besar ($>$): operator ini digunakan untuk membandingkan apakah suatu bilangan atau nilai lebih besar dari bilangan atau nilai yang lain. Contoh: $5 > 3$ akan menghasilkan nilai kebenaran true karena 5 lebih besar dari 3.
- b) Lebih kecil ($<$): operator ini digunakan untuk membandingkan apakah suatu bilangan atau nilai lebih kecil dari bilangan atau nilai yang lain. Contoh: $2 < 7$ akan menghasilkan nilai kebenaran true karena 2 lebih kecil dari 7.
- c) Sama dengan ($==$): operator ini digunakan untuk membandingkan apakah suatu bilangan atau nilai sama dengan bilangan atau nilai yang lain. Contoh: $3 + 4 == 7$ akan menghasilkan nilai kebenaran true karena 3 ditambah 4 sama dengan 7.
- d) Tidak sama dengan ($!=$): operator ini digunakan untuk membandingkan apakah suatu bilangan atau nilai tidak sama dengan bilangan atau nilai yang lain. Contoh: $5 - 2 != 2$ akan menghasilkan nilai kebenaran true karena 5 dikurangi 2 tidak sama dengan 2. VR.Vex belum menyediakan operasi ini.

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

- e) Lebih besar atau sama dengan (\geq): operator ini digunakan untuk membandingkan apakah suatu bilangan atau nilai lebih besar atau sama dengan bilangan atau nilai yang lain. Contoh: $5 + 2 \geq 6$ akan menghasilkan nilai kebenaran true karena 5 ditambah 2 lebih besar atau sama dengan 6. VR.Vex belum menyediakan operasi ini.
- f) Lebih kecil atau sama dengan (\leq): operator ini digunakan untuk membandingkan apakah suatu bilangan atau nilai lebih kecil atau sama dengan bilangan atau nilai yang lain. Contoh: $2 * 3 \leq 7$ akan menghasilkan nilai kebenaran true karena 2 dikali 3 lebih kecil atau sama dengan 7. VR.Vex belum menyediakan operasi ini.

Operasi Logika

Operasi logika adalah operasi dalam matematika dan pemrograman komputer yang digunakan untuk memanipulasi nilai kebenaran (true atau false) dan menghasilkan nilai kebenaran baru sebagai hasil dari operasi tersebut.

Berikut ini adalah beberapa contoh operasi logika yang sering digunakan:

- a) AND: operator AND ($\&\&$) menghasilkan nilai kebenaran true **hanya jika kedua nilai yang dibandingkan sama-sama true**. Jika salah satu atau kedua nilai tersebut false, maka hasilnya akan false. Contoh: $(5 > 3) \&\& (4 < 7)$ akan menghasilkan nilai kebenaran true karena keduanya true.
- b) OR: operator OR ($\|$) menghasilkan nilai kebenaran **true jika salah satu atau kedua nilai yang dibandingkan adalah true**. Jika keduanya false, maka hasilnya akan false. Contoh: $(5 > 3) \| (4 > 7)$ akan menghasilkan nilai kebenaran true karena salah satu dari keduanya true.
- c) NOT: operator NOT (!) digunakan untuk **membalikkan nilai kebenaran dari sebuah ekspresi**. Jika nilai kebenaran awalnya true, maka NOT akan menghasilkan nilai false, dan sebaliknya. Contoh: $!(5 > 3)$ akan menghasilkan nilai kebenaran false karena ekspresi awalnya true.

Operasi logika sering digunakan dalam pengambilan keputusan dalam pemrograman, seperti dalam percabangan if-else atau loop while. Selain itu, operasi logika juga digunakan dalam algoritma kecerdasan buatan dan pemrosesan data.

Fungsi Matematika dalam VR.Vex

Fungsi matematika adalah hubungan matematis antara sebuah input dan output, yang ditentukan oleh sebuah aturan atau persamaan matematika. Dalam matematika, fungsi ditulis sebagai $f(x)$, yang menggambarkan bahwa output (y) tergantung pada input (x).

Berikut ini adalah beberapa jenis fungsi matematika yang tersedia dalam VR.Vex :

- a) Fungsi abs : Mengembalikan nilai absolut. Contoh $\text{abs}(10) \rightarrow 10$, $\text{abs}(-10) \rightarrow 10$
- b) Fungsi floor : Membulatkan kebawah bilangan desimal ke bilangan bulat terdekat
- c) Fungsi ceiling : Membulatkan keatas bilangan desimal ke bilangan bulat terdekat
- d) Fungsi sqrt : Mencari akar dari suatu bilangan
- e) Fungsi trigonometri : fungsi matematika yang melibatkan sin, cos, tan, asin, acos dan atan. Fungsi trigonometri sering digunakan dalam geometri dan fisika untuk menghitung sudut, jarak, dan kecepatan.

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

- f) Fungsi eksponensial dan logaritma: Melakukan operasi eksponensial dan logaritma, seperti fungsi e^x untuk menghitung nilai eksponensial, fungsi \log untuk menghitung nilai logaritma basis 10, fungsi \ln natural untuk menghitung nilai logaritma basis e. E adalah basis bilangan Euler yang bernilai sekitar 2.71828.

Proyek Menggambar Segitiga Siku-siku, jika diketahui Tinggi 1000 mm dan Alas 500 mm

Segitiga siku-siku, sudut-sudutnya terdiri dari sudut 90 derajat (sudut siku-siku) dan dua sudut lainnya yang totalnya 90 derajat juga. Jadi total sudutnya 180 derajat. Oleh karena itu, jika tinggi dan alas segitiga siku-siku diketahui, maka kita dapat menggunakan fungsi trigonometri **atan** untuk menghitung nilai dari kedua sudut tersebut.

Dalam kasus ini, tinggi segitiga siku-siku adalah 1000 dan alasnya adalah 500. Kita dapat menghitung sudut pertama (yang berada di samping tinggi) menggunakan fungsi trigonometri **arctan** (atau **atan**) sebagai berikut:

```
tinggi = 1000
alas = 500
sudut1 = atan(tinggi/alas)
```

Nilai sudut1 = 63.434 derajat dan sudut ke-2, maka dapat dihitung $90 - 63.434 = 26.566$.

Melalui Teorema Pythagoras dapat dihitung panjang sisi miring dari segitiga siku-siku dengan cara menghitung akar kuadrat dari jumlah kuadrat dari kedua sisi lainnya. Jika kita menandai sisi miring dengan c, dan kedua sisi lainnya dengan a dan b, maka teorema Pythagoras dapat dituliskan $c^2 = a^2 + b^2$ atau $c = \sqrt{a^2 + b^2}$

Dalam kasus segitiga siku-siku dengan alas 500 dan tinggi 1000, kita dapat menggunakan teorema Pythagoras untuk menghitung panjang sisi miringnya sebagai berikut:

```
sisiMiring = sqrt(tinggi*tinggi + alas*alas)
```

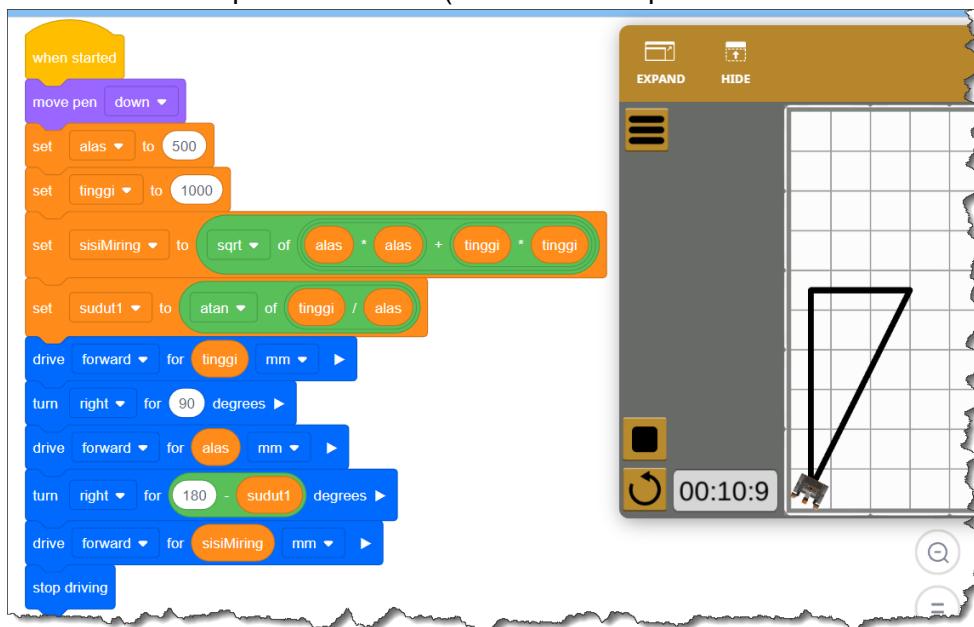
Lihat potongan program menghitung sisiMiring dan sudut1 pada Gambar 32.



Gambar 32 Penggunaan Pythagoras dan Fungsi atan untuk segitiga siku-siku

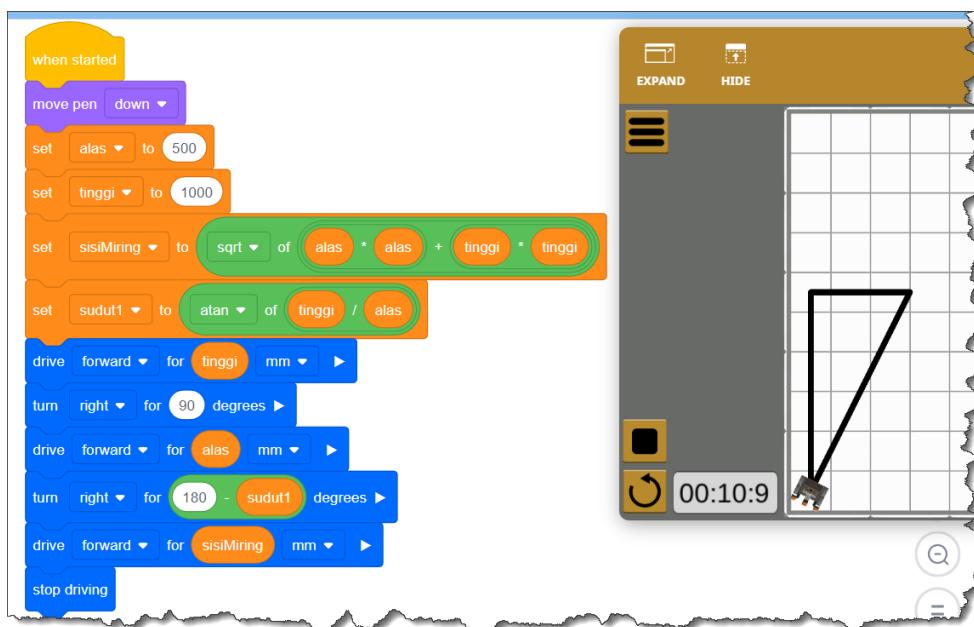
PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

Setelah mengetahui sudut1 dan sisiMiring, dilanjutkan untuk menggambar segitiganya. Tambahkan blok perintah berikut (lihat hasil akhir pada



Gambar 33) :

- [move pen down]
- [drive forward for tinggi mm]
- [turn right for 90 degrees]
- [drive forward for alas mm]
- [turn right for 180 – sudut1 degrees] //untuk siap menggambar sudut1
- [drive forward for sisiMiring mm] //untuk menggambar sisiMiring
- [stop driving]



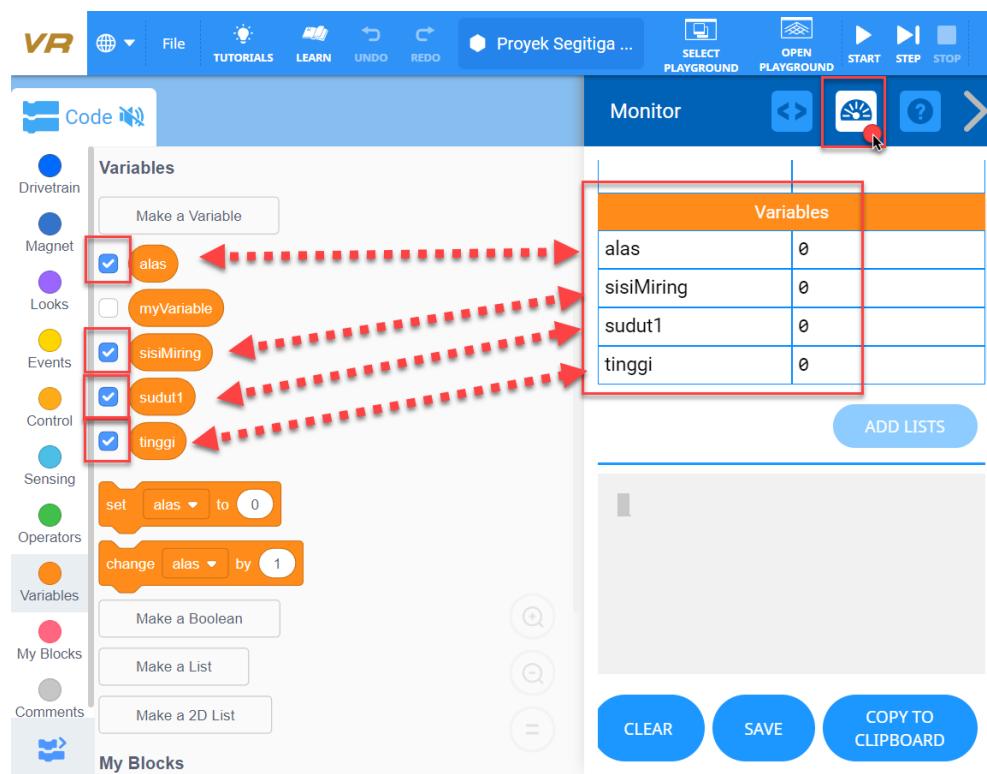
Gambar 33 Proyek Segitiga Siku-siku dengan Tinggi=1000 dan Alas 500

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

Memantau (monitor) nilai variabel

Nilai variabel yang sudah dideklarasikan dapat dilihat menggunakan fasilitas Monitor. Semua variabel yang akan dipantau, berikan tanda cek (Gambar 34), kemudian aktifkan tombol [Monitor]. Saat program dijalankan nilai variabel akan ditampilkan di layar.

Misal variabel-variabel ini akan dipantau : **tinggi, alas, sudut1, sisiMiring**. Variabel dapat dipantau sebelum, sedang dan setelah program berjalan. Saat kondisi awal program belum berjalan semua variabel bernilai 0. Saat program berjalan dapat dilihat perubahan nilai pada variabel tersebut. Setelah program berakhir nilai variabel dapat dilihat pula, seperti pada Gambar 32.

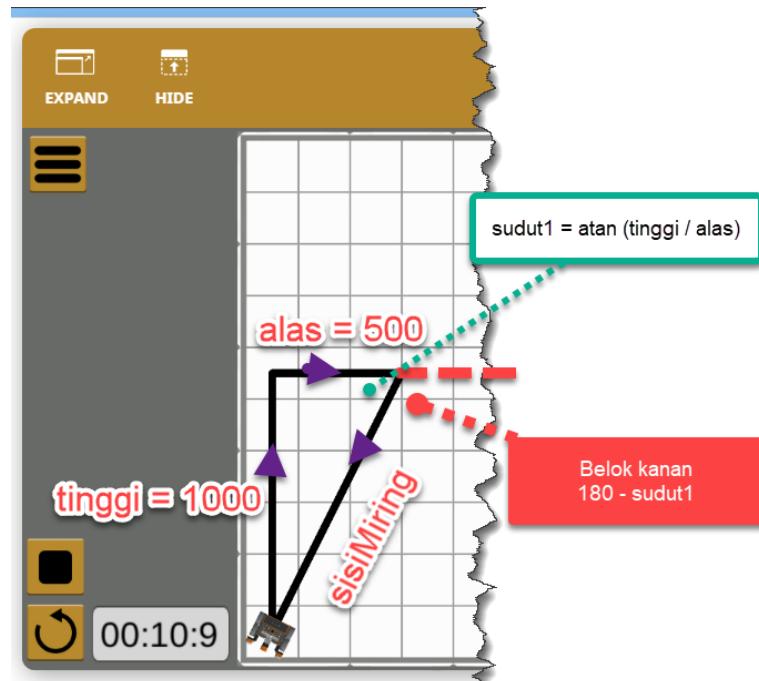


Gambar 34 Monitor (Memantau) Nilai Variabel

E. T04a-Buat Segitiga Siku-siku

Buat proyek untuk menggambar segitiga siku-siku, gunakan playground Grid Map dengan ketentuan sbb (lihat Gambar 35) :

- tinggi = 1000
- alas = 500
- sudut1 = $\text{atan}(\text{tinggi}/\text{alas})$
- sisiMiring = $\sqrt{\text{tinggi}^2 + \text{alas}^2}$



Gambar 35 Proyek Menggambar Segitiga Siku-siku

F. T04b-Buat 4 buah Segitiga Siku-siku

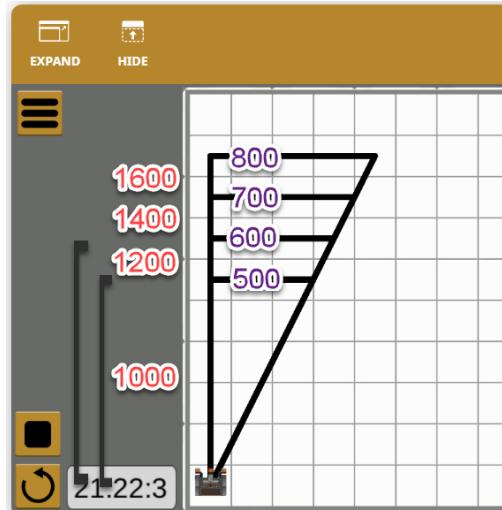
Buat proyek untuk menggambar segitiga siku-siku, gunakan playground Grid Map dengan ketentuan sbb (lihat Gambar 36). Modifikasi dari soal sebelumnya :

Segitiga ke-1 :
tinggi = 1000
alas = 500

Segitiga ke-2 :
tinggi = 1200
alas = 600

Segitiga ke-3:
tinggi = 1400
alas = 700

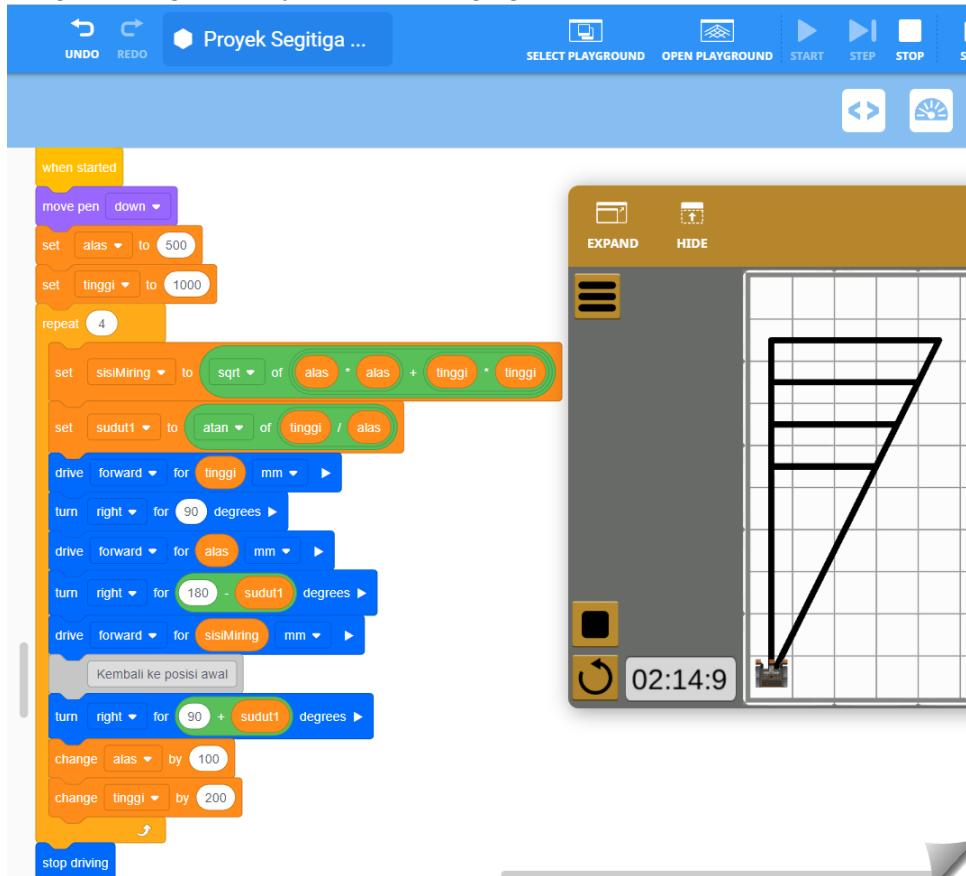
Segitiga ke-4 :
tinggi = 1600
alas = 800



Gambar 36 Proyek Menggambar 4 buah Segitiga Siku-siku

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

Program lengkap proyek 4 buah Segitiga Siku-siku, tampak pada Gambar 37.



Gambar 37 Hasil akhir Proyek 4 buah Segitiga Siku-siku

G.T04c-Buat 4 buah Segitiga Siku-siku (Gunakan List)

Bagaimana proyek yang sama dengan T04b, namun menggunakan variabel listTinggi untuk menampung tinggi ke empat segitiga {1000, 1200, 1400, 1600} dan variabel listAlas untuk menampung alas {500, 600, 700, 800}

H.T04d-Buat 4 buah Segitiga Siku-siku (Gunakan 2DList)

Bagaimana proyek yang sama dengan T04b, namun menggunakan variabel list2D untuk menampung tinggi dan alas ke empat segitiga {{1000, 500}, {1200, 600}, {1400, 700}, {1600, 800}}

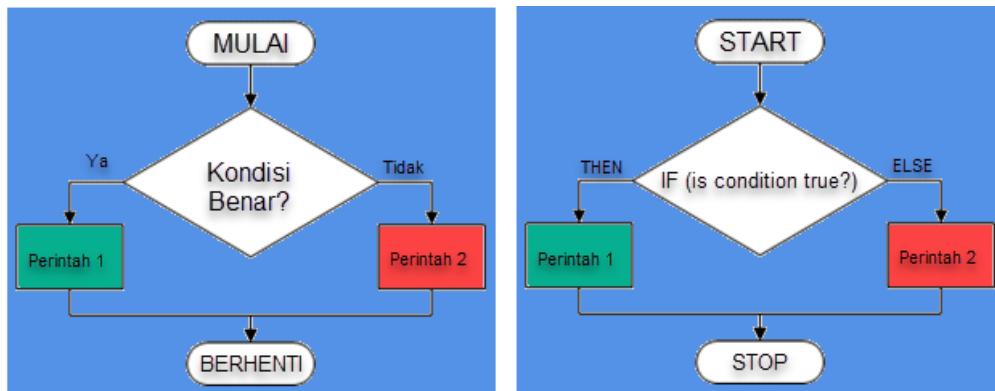
BAB 5

PEMILIHAN KASUS DAN SENSOR WARNA

Pelajaran 5 PEMILIHAN KASUS DAN SENSOR WARNA	Target Pelajaran
	<ol style="list-style-type: none"> 1. Mengenal bagaimana struktur perintah pemilihan kasus atau kondisi (IF-THEN, IF-THEN-ELSE) 2. Mengenal Eye Sensor (Front dan Down) 3. Membuat program robot untuk memilih bergerak berdasar kondisi yang dihadapi. Kondisi dapat dibaca menggunakan eye sensor. Gunakan playground Disk Maze 4. Mendemonstrasikan program
Durasi Belajar 45-60 menit	Deskripsi Pelajaran Siswa diminta untuk menelusuri struktur pemilihan kasus atau SELECTION dan bagaimana penggunaan dalam pemrogramma. Siswa melanjutkan pembuatan program menggunakan playground Disk Maze untuk mengatur robot bergerak sesuai kondisi yang dihadapi.

A. Algoritma Pemilihan Kasus (Selection)

Struktur algoritma pemilihan atau dikenal dengan pencabangan (*branching*) ini digunakan untuk mengerjakan satu aksi dari beberapa pilihan yang diberikan. Dengan algoritma pemilihan, ada kalanya satu atau beberapa aksi akan dikerjakan dan yang lain tidak dikerjakan tergantung kondisi yang dihadapi. Struktur pencabangan digambarkan dalam bentuk *flowchart* (lihat Gambar 38) untuk memudahkan pengertian, sebagai berikut :

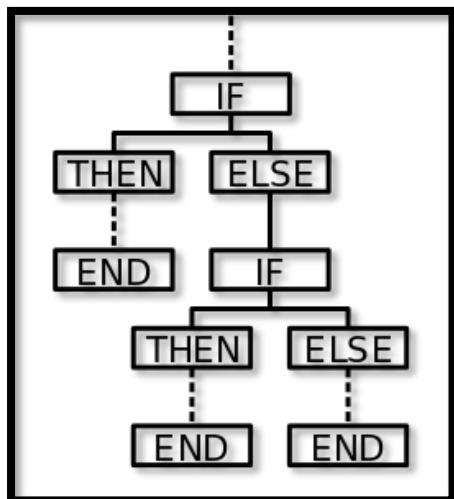


Gambar 38 Flowchart IF-THEN-ELSE

Pada Gambar 38, jika kondisi dalam belah ketupat adalah benar, maka perintah ke-1 yang akan dijalankan, dan jika kondisi salah maka perintah ke-2 yang akan dijalankan. Tampak jelas bahwa hanya salah satu perintah yang dijalankan, apakah perintah ke-1 atau perintah ke-2. Inilah fungsi dari pencabangan atau pemilihan kasus atau atau perintah kondisional.

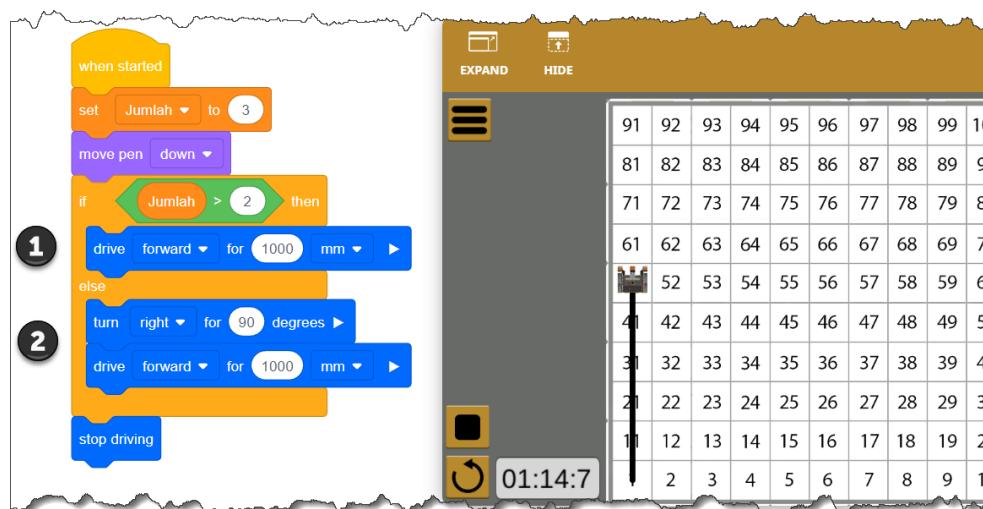
PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

Dalam kasus tertentu pencabangan dapat lebih dari dua pilihan, bagaimana menanganinya? Apabila lebih dari 2 pilihan, dapat digunakan IF-THEN-ELSE lebih dari satu. Contoh jika pilhan ada 10 buah, maka paling tidak digunakan 9 perintah IF-THEN-ELSE. Jika kondisinya 3 pilihan, maka digunakan 2 perintah IF-THEN-ELSE, strukturnya seperti Gambar 39.



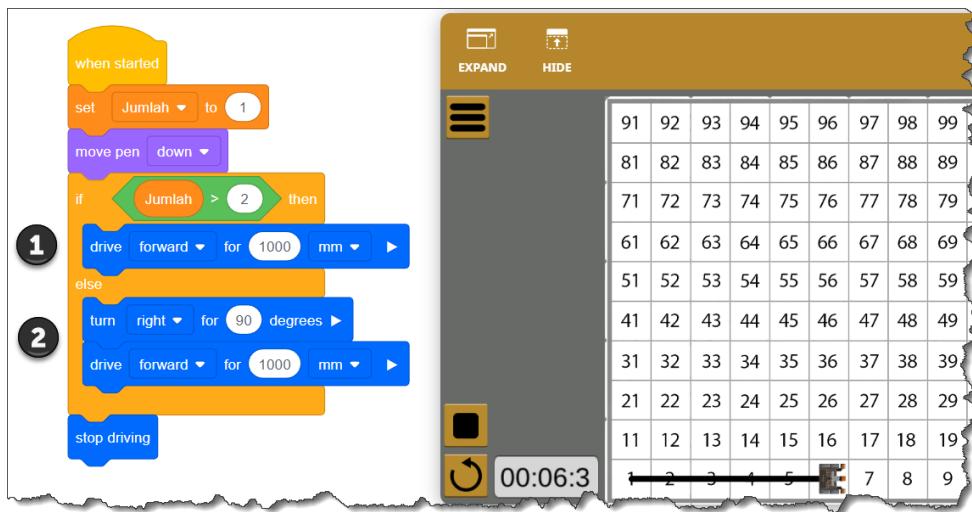
Gambar 39 Struktur Pencabangan dengan 3 kondisi

Sebuah projek sederhana untuk memperlihatkan bagaimana perintah IF-THEN-ELSE bekerja. Apabila nilai Jumlah > 2 maka perintah ke-1 akan dijalankan (lihat Gambar 40) dan sebaliknya bila kondisi tidak terpenuhi (atau Jumlah ≤ 2) maka perintah ke-2 akan dijalankan (lihat Gambar 41).



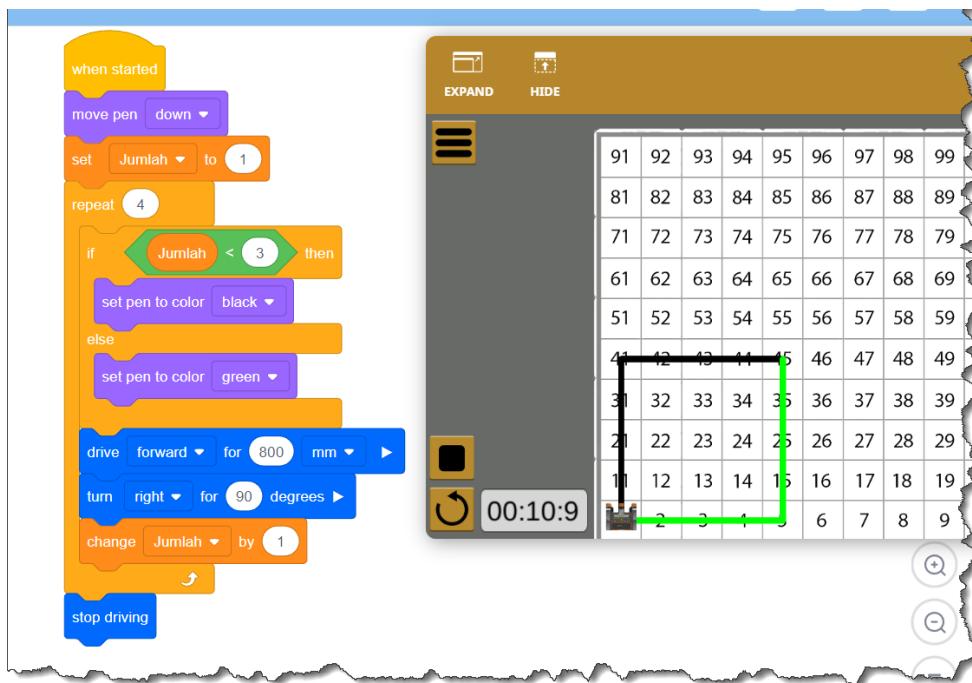
Gambar 40 Program IF-THEN-ELSE, apabila nilai Jumlah > 2

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX



Gambar 41 Program IF-THEN-ELSE, apabila nilai Jumlah <= 2

Sebuah program sederhana untuk menggambar persegi panjang dengan warna pen yang berbeda. Saat nilai variabel Jumlah < 3, menggunakan pen berwarna hitam (*black*) dan selebihnya pen berwarna hijau (*green*), seperti pada Gambar 42. Perhatikan struktur programnya, if-then-else berada dalam pengulangan repeat. Apa yang terjadi apabila blok [change Jumlah by 1] diganti angkanya menjadi [change Jumlah by 2]?



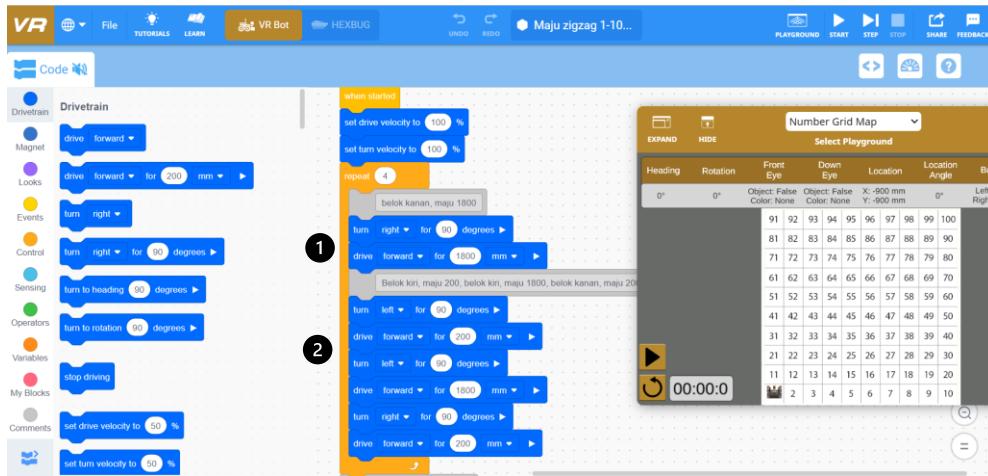
Gambar 42 Contoh Pencabangan (IF-THEN-ELSE) digabung dengan Pengulangan (Repeat)

B. T05a- Maju Zigzag 1-10, 20-11, 21-30, dst (Gunakan Repeat, Tanpa If-Then-Else)

Latihan Robot menjelajahi dan menggambar angka dari 1 sampai 100, dengan pola jelajah 1-10, 20-11, 21-30, dan seterusnya sampai angka 100, tampak pola membentuk zigzag. Anda

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

dapat menggunakan pengulangan Repeat atau Repeat Until atau While, tanpa menggunakan if-then-else. Lihat program pada Gambar 43 dan lanjutannya pada Gambar 44.



Gambar 43 Maju Zigzag 1-10, 20-11, 21-30, dst (Gunakan Repeat, Tanpa If-Then-Else) bagian A



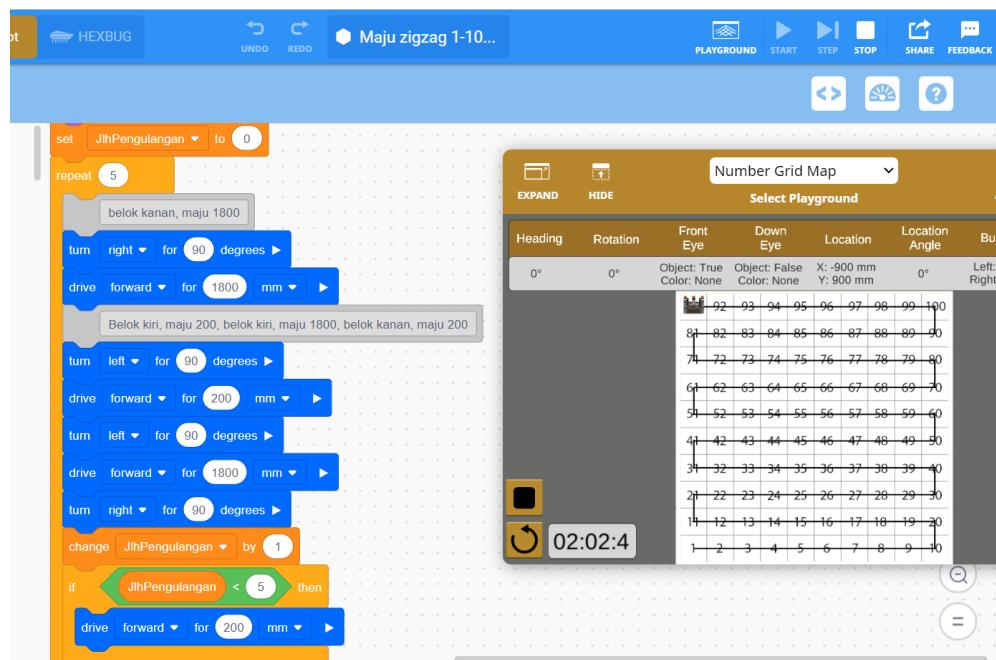
Gambar 44 Maju Zigzag 1-10, 20-11, 21-30, dst (Gunakan Repeat, Tanpa If-Then-Else) bagian B

Analisis : Bila diperhatikan blok nomor 1 dan 2, mirip dengan 3 dan 4. Mengapa blok 3 dan 4 dibuang saja, kemudian blok [repeat 4] menjadi [repeat 5]? Sudah tahu jawabannya? Diujung pada angka 91, robot akan maju menjalankan [drive forward for 200 mm] sehingga menabrak dinding. Blok perintah ini yang tidak diperlukan, saat sudah mencapai titik ujung.

C. T05b- Maju Zigzag 1-10, 20-11, 21-30, dst (Gunakan Repeat, dan If-Then-Else)

Modifikasi proyek penjelajahan robot sebelumnya. Robot menjelajahi dan menggambar angka dari 1 sampai 100, dengan pola jelajah 1-10, 20-11, 21-30, dan seterusnya, tampak pola membentuk zigzag. Anda dapat menggunakan pengulangan Repeat atau Repeat Until atau While, dan if-then-else, seperti pada Gambar 45. Jangan lupa blok [move pen down], agar robot menggunakan penanya.

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX



Gambar 45 Maju Zigzag 1-10, 20-11, 21-30, dst (Gunakan Repeat, dan If-Then-Else)

Analisis : Bila diperhatikan adanya variabel JlhPengulangan. Variabel ini digunakan untuk menyimpan berapa kali pengulangan. Variabel ini akan digunakan saat pengulangan ke-5. Jika JlhPengulangan < 5 maka blok [drive forward for 200 mm] akah dikerjakan, tapi saat JlhPengulangan = 5 maka akan dilewat.

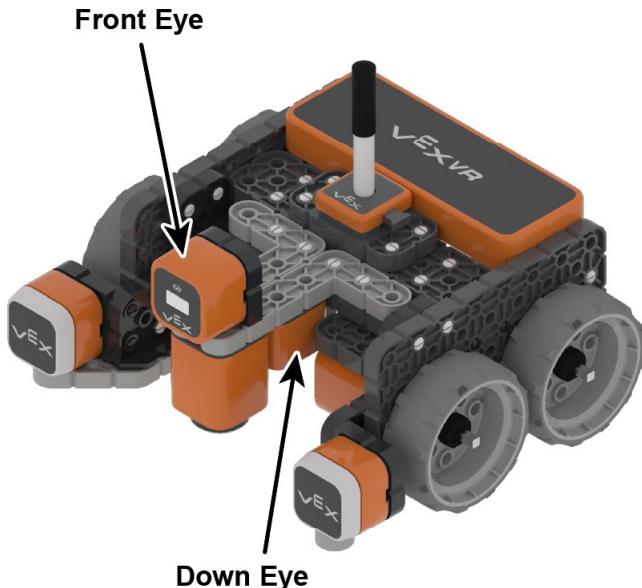
Ini salah satu tip, apabila ada bagian yang tidak ingin dijalankan oleh sistem bila kondisi tertentu. Variabel jlhPengulangan dapat dijadikan pencatat.

D. T05c- Maju Zigzag 1-10, 11-20, 31-40, dst (Gunakan Repeat, dan If-Then-Else)

Modifikasi proyek penjelajahan robot sebelumnya. Robot menjelajahi dan menggambar angka dari 1 sampai 100, dengan pola jelajah 1-10, 11-20, 21-30, dan seterusnya sampai angka 100. Mudah bukan?

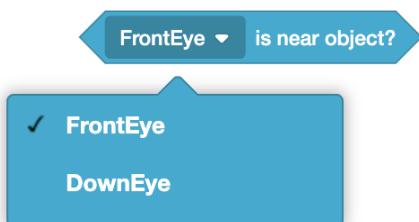
E. Mengenal Sensor

Robot VR.Vex memiliki 2 Eye Sensors, yang satu menghadap ke depan (*Front Eye*) dan yang lain menghadap ke bawah (*Down Eye*). Sensor-sensor ini dapat mendeteksi apakah ada objek didekatnya dan dapat mengenali warna seperti warna merah, hijau, biru dan tanpa warna. Berikut letak kedua sensor tersebut (*Eye Sensors*) pada Gambar 46.

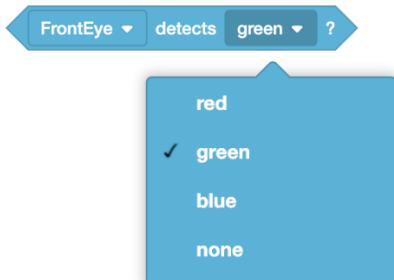


Gambar 46 Sensor untuk mendeteksi benda dan warna (sumber : VR.Vex.com)

Blok [FrontEye atau DownEye] dapat digunakan untuk mendeteksi adanya objek didekatnya.



Blok [FrontEye/DownEye detects red/green/blue/none] digunakan untuk mendeteksi warna.



Beberapa proyek ke depan, sensor-sensor ini akan digunakan untuk mendeteksi objek seperti benda, tembok, bangunan, garis, batas lokasi. Selain itu dapat mendeteksi warna primer (merah, hijau, biru) dan warna terang atau warna gelap.

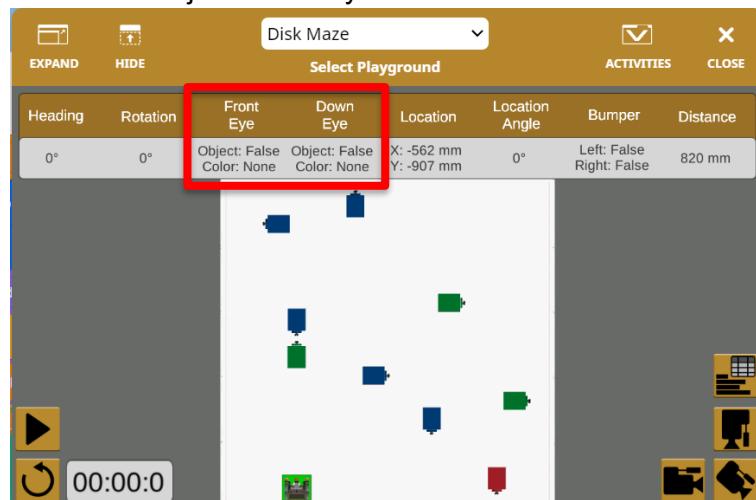
F. Disk Maze Playground

Pada playground Disk Maze, anda diminta untuk menggerakkan robot sampai ke simbol merah. Gunakan sensor depan (*FrontEye*) atau sensor bawah (*DownEye*), karena simbol warna diletakkan berupa plat berwarna vertikal dan plat berwarna di lantai, seperti pada Gambar 47 dan

Gambar 48. Adapun algoritmanya sebagai berikut :

- Jika tidak mendeteksi, maka maju
- Jika mendeteksi hijau, maka belok kanan 90 derajat
- Jika mendeteksi biru, maka belok kiri 90 derajat
- Jika mendeteksi merah, maka berhenti

Perhatikan Gambar 47, pada kotak merah untuk memantau apakah sensor melihat ada objek dan mendeteksi warna tertentu. Front Eye – Object : False, Color : None, artinya belum ada objek didekatnya dan belum ada warna terdeteksi.

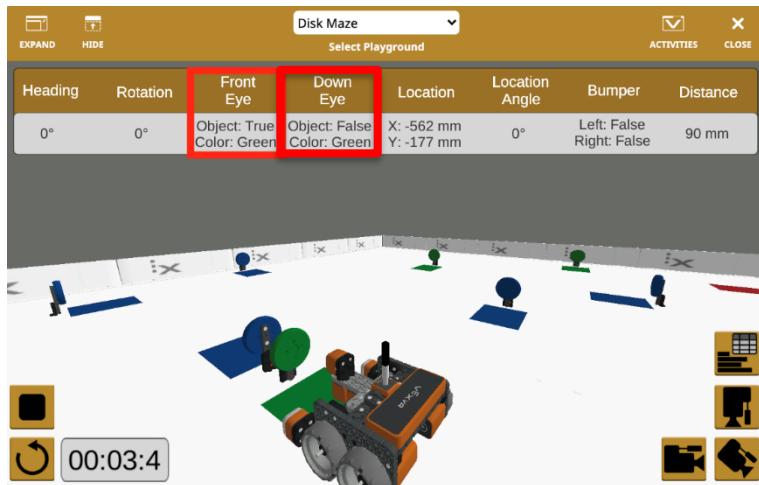


Gambar 47 Disk Maze Playground tampak dari atas

Perhatikan

Gambar 48, pada kotak merah Front Eye – Object : True, Color : Green, artinya ada objek didekatnya dan warna hijau. Untuk sensor Down Eye - Object : False, Color : Green, artinya tidak ada objek didekatnya namun lantai warna hijau.

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX



Gambar 48 Disk Maze Playgroud tampak dari samping

BAB 6

MY BLOCKS (Sub Program)

Pelajaran 6 MY BLOCKS (Sub Program)	<p>Target Pelajaran</p> <ol style="list-style-type: none"> 1. Mengenal perintah My Blocks atau Sub-program. 2. Membuat program menggunakan My Blocks yang dibuat sendiri 3. Mendemonstrasikan program, menggunakan playground Shape Tracer
	<p>Deskripsi Pelajaran</p> <p>Siswa diminta untuk membuat beberapa my block dengan parameter input</p> <p>Siswa membuat program sesuai bentuk (shape) yang tertera di playground</p>
Durasi Belajar 45-60 menit	<p>Bahan Pelajaran</p> <p>Lembar Kerja 6.1 – Playground Shape Tracer</p> <p>Lembar Kerja 6.2 – Tabel pengujian</p>

A. Mengenal perintah My Blocks atau Sub-program

Sub program dalam pemrograman (atau disebut My Block dalam VR.Vex) adalah blok kode yang dapat dipanggil dari program utama dan memiliki tugas yang spesifik. Sub program sering digunakan untuk mengelompokkan logika yang sama, sehingga kode program menjadi lebih mudah dibaca, dipelihara, dan dikelola. Dalam bahasa pemrograman, sub program dikenal dengan nama fungsi, prosedur, atau subrutin. Fungsi biasanya mengembalikan nilai ke program utama, sedangkan prosedur dan subrutin tidak mengembalikan nilai.

Pemanggilan sub program biasanya dengan menyebutkan nama sub program yang dipanggil dari program utama untuk menjalankan sub program tersebut. Setelah sub program selesai dijalankan, program utama akan kembali berjalan dari titik pemanggilan sub program tersebut. Hal ini membuat program lebih modular dan dapat dipecah menjadi bagian-bagian yang lebih kecil untuk memudahkan pengembangan dan perawatan.

Sub program juga dapat digunakan untuk melakukan tugas yang kompleks secara terpisah dari program utama, yang memungkinkan programmer memecahkan masalah yang lebih besar menjadi bagian-bagian yang lebih kecil dan mudah diatur. Beberapa keuntungan penggunaan sub program dalam pemrograman adalah:

- a) **Modularitas:** Sub program memungkinkan kode program dibagi menjadi bagian yang lebih kecil dan terorganisir, sehingga mudah dipahami dan diatur.
- b) **Penggunaan Kembali :** Jika sub program telah dibuat, kode tersebut dapat digunakan kembali di beberapa bagian program yang berbeda, sehingga menghemat waktu.
- c) **Pemeliharaan:** Kode sub program dapat diperbarui atau dimodifikasi secara terpisah, tanpa harus memodifikasi keseluruhan program utama. Hal ini membuat perawatan dan pengembangan kode program menjadi lebih mudah.
- d) **Abstraksi:** Sub program dapat digunakan untuk menyembunyikan detail implementasi yang kompleks dan hanya menekankan fungsionalitas yang

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

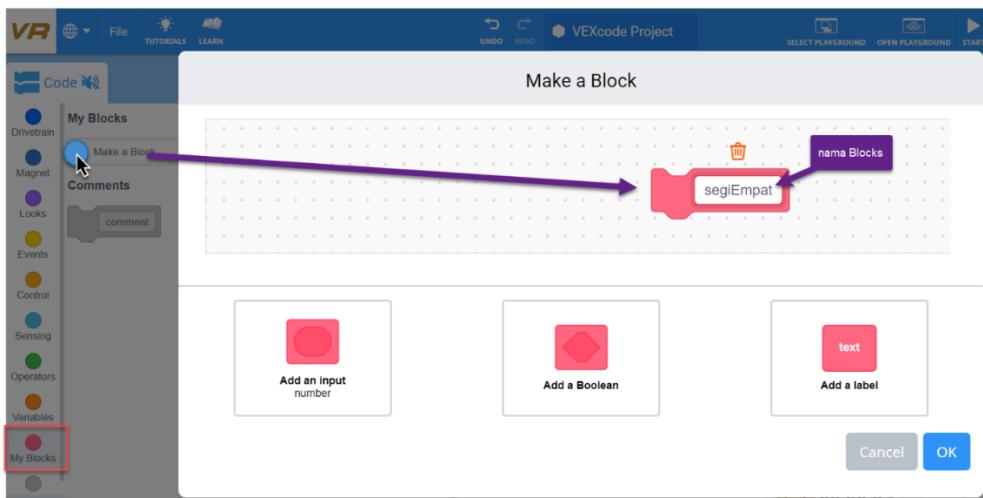
diperlukan kepada pengguna. Hal ini membuat kode program lebih mudah digunakan dan dipahami.

Sub program merupakan salah satu fitur penting dalam pemrograman, dan hampir semua bahasa pemrograman modern mendukung penggunaannya, termasuk VR.Vex.

B. Proyek menggunakan My Blocks buatan sendiri

My Block tanpa parameter

Seperti sudah disebutkan manfaat yang banyak bila menggunakan sub program (my blocks). Kita akan mencoba membuat My Blocks segiEmpat, intinya program ini akan menggambar segiempat dengan sisi 500 mm. Kita akan gunakan my blocks tanpa parameter, dengan playground Art Canvas. Langkahnya pilih My Blocks – Make a Block, kemudian isikan nama Block : segiEmpat, kemudian Ok (lihat Gambar 49).



Gambar 49 Membuat my Blocks segiEmpat

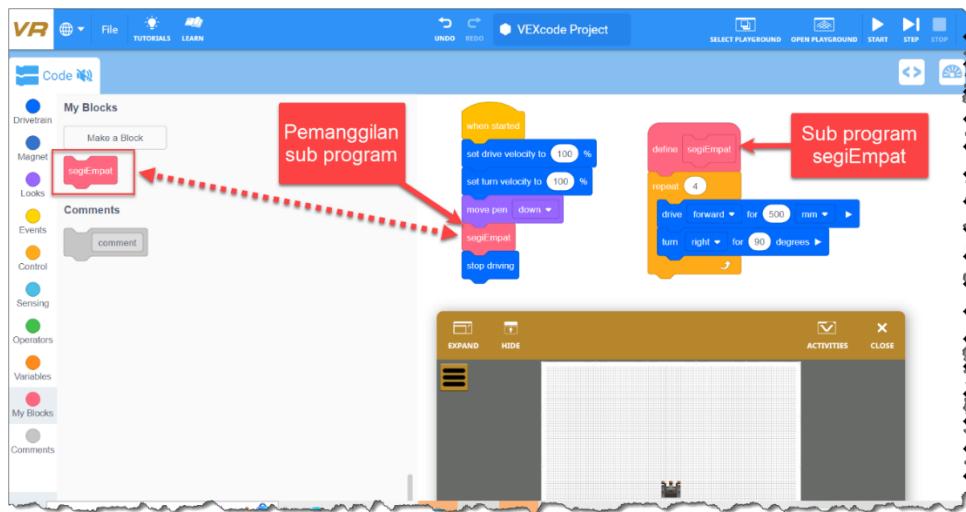
Setelah pembuatan sub program (block) segiEmpat, akan diisikan blok perintah untuk membuat segiempat dengan sisi 500 mm (lihat Gambar 50) :

- **Define segiEmpat** //Definisikan isi block segiEmpat
- Repeat 4
 - Drive forward for 500 mm
 - Turn right for 90 degrees

Pada program utama block segiEmpat dapat dipanggil dengan menyertakan blok perintah [segiEmpat] :

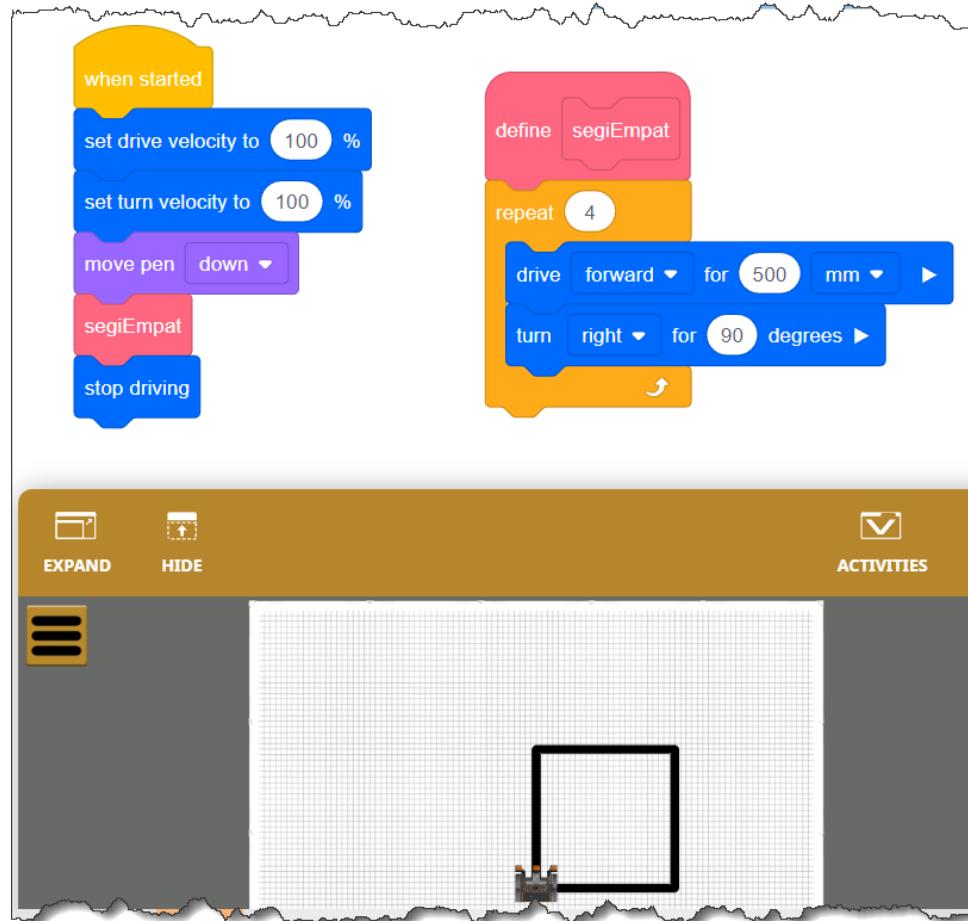
- When started
- Set drive velocity to 100%
- Set turn velocity to 100%
- Move pen down
- **segiEmpat** //Pemanggilan segiEmpat

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX



Gambar 50 Pembuatan dan Pemanggilan Block [segiEmpat]

Setelah dijalankan, tampak robot akan bergerak membentuk segiEmpat, seperti pada Gambar 51.



Gambar 51 Pembuatan segiEmpat menggunakan my Blocks

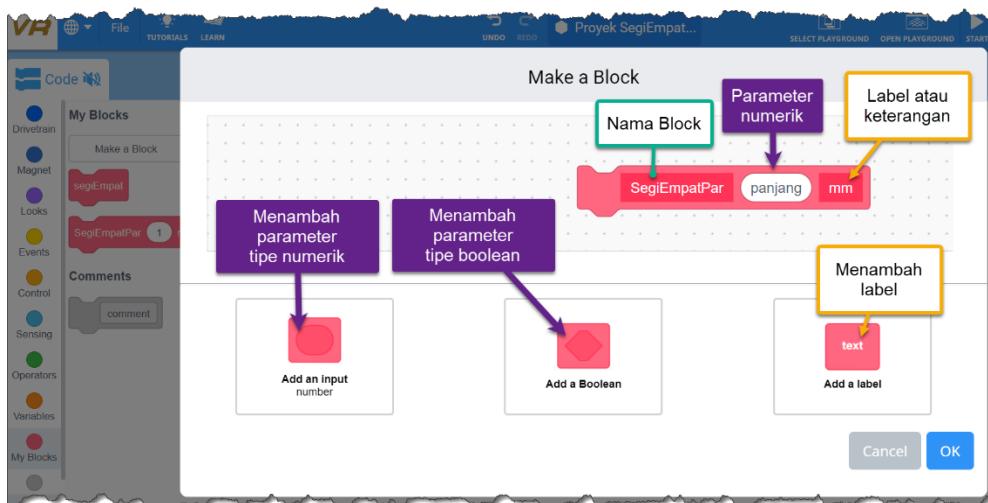
My Block dengan parameter

Sub program dengan parameter adalah sub program yang menerima satu atau beberapa nilai sebagai argumen saat dipanggil dari program utama. Parameter ini kemudian

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

digunakan di dalam sub program untuk melakukan tugas tertentu. Dalam bahasa pemrograman, parameter juga sering disebut dengan istilah argumen. Penggunaan sub program dengan parameter sangat berguna ketika ingin membuat kode yang dapat digunakan kembali dengan variasi nilai yang berbeda. Sebagai contoh, jika kita ingin membuat sub program untuk menggambar segiEmpat dengan panjang yang berbeda, dengan menggunakan parameter untuk menerima nilai panjang. Langkahnya sebagai berikut (lihat Gambar 52) :

- Pilih Code My Block, klik [Make a block]
- Setelah muncul dialog Make a Block (lihat Gambar 52), ganti nama blok menjadi **SegiEmpatPar**
- Klik [Add an input] untuk menambahkan parameter tipe numerik, setelah itu ganti parameter number menjadi **panjang**
- Klik [Add a label] untuk menambahkan label, setelah itu ganti label menjadi **mm**
- Klik [OK]



Gambar 52 Make a block with parameter

Setelah pembuatan sub program (block) SegiEmpatPar, akan diisiakan blok perintah untuk membuat segiempat dengan sisi 500 mm (lihat Gambar 50) :

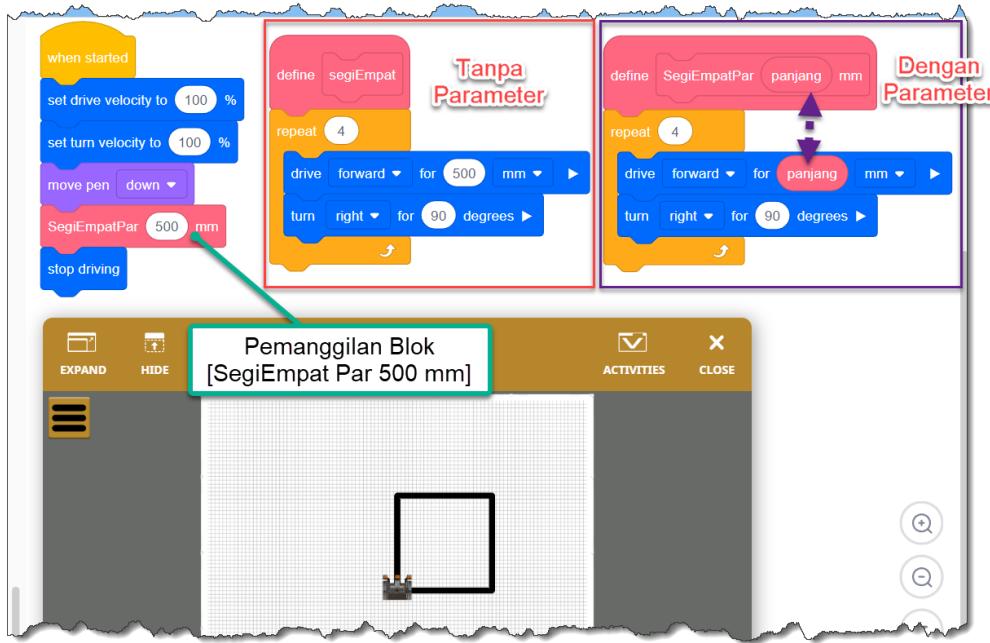
- **Define SegiEmpatPar panjang mm** //Definisikan isi block segiEmpat
- Repeat 4
 - Drive forward for [panjang] mm
 - Turn right for 90 degrees

Pada program utama block segiEmpatPar dapat dipanggil dengan menyertakan blok perintah disertai nilai untuk parameter [segiEmpatPar 500 mm] :

- When started
- Set drive velocity to 100%
- Set turn velocity to 100%
- Move pen down
- **segiEmpatPar [500] mm** //Pemanggilan segiEmpatPar

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

Contoh sub program untuk menggambar Segi Empat dengan parameter, seperti pada Gambar 53.



Gambar 53 Pembuatan segiEmpat menggunakan my Blocks dengan Parameter

Analisis :

Perhatikan cara pemanggilan Block bila ada parameternya, maka nilai untuk parameter disertakan juga. Nilai bisa diatur, misal bila ingin menggambar segi empat dengan sisi 500, maka ditulis 500. Bila ingin menggambar segi empat dengan sisi 700, maka ditulis 700.

Perhatikan juga blok segiEmpat dan SegiEmpatPar, walau tampak sama, blok SegiEmpatPar lebih fleksibel dalam menggambar berbagai segi empat dengan berbagai ukuran, dikarenakan apa parameternya. Inilah salah satu kegunaan parameter untuk blok atau sub program. Coba Anda tambahkan pemanggilan sub program segiEmpatPar dengan nilai parameter 300, kemudian buat juga untuk nilai parameter 100. Bagaimana hasilnya?

C. T06a- Proyek Segi Tiga Puluh

Buat Block (sub program) untuk membuat lintasan SegiTigaPuluhan dengan parameter sisi (lihat Gambar 54).

Sub program SegiTigaPuluhan sebagai berikut :

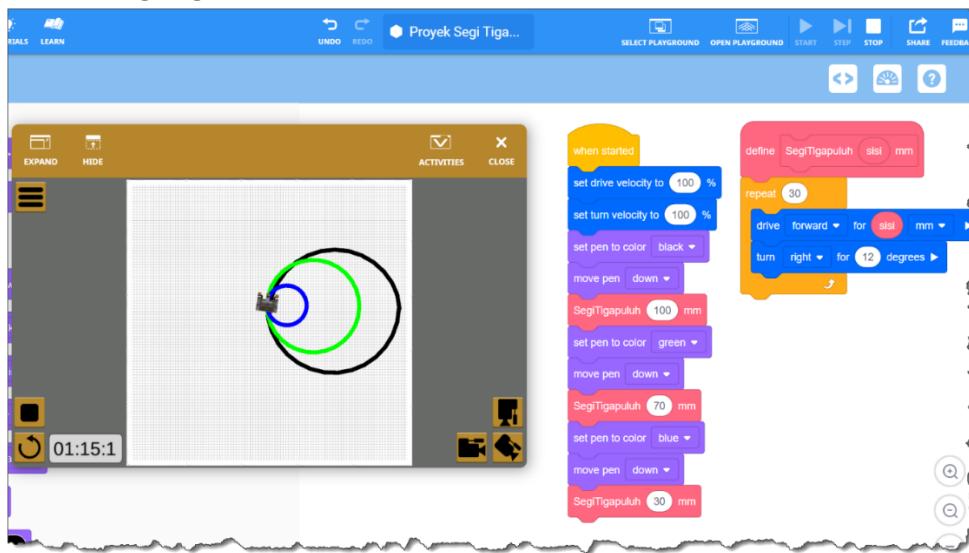
- Define SegiTigaPuluhan sisi mm
- Repeat 30
 - Drive forward for sisi mm
 - Turn right for 12 degrees

Program Utama sebagai berikut :

- When Started

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

- Set drive velocity to 100%
- Set turn velocity to 100%
- Set pen to color black
- Move pen down
- **SegiTigaPuluhan [100] mm**
- Set pen to color green
- Move pen down
- **SegiTigaPuluhan [70] mm**
- Set pen to color blue
- Move pen down
- **SegiTigaPuluhan [30] mm**



Gambar 54 Proyek Segi Tiga Puluh dengan Blocks

Analisis : Perhatikan blok SegiTigaPuluhan dipanggil 3 kali, dengan parameter yang berbeda-beda yaitu 100, 70 dan 30. Hasil eksekusi program, tampak segitigapuluhan dengan 3 ukuran berbeda. SegiTigaPuluhan tampai seperti bentuk lingkaran.

D. T06b- Proyek Segi Banyak

Satu putaran penuh atau lingkaran memiliki 360 derajat. Ini merupakan ukuran standar untuk mengukur sudut dalam matematika dan fisika. Beberapa contoh penggunaan putaran, antara lain :

- a) Jam tangan analog : jarum jam akan menunjukkan satu putaran atau lingkaran penuh setiap 12 jam, sehingga satu putaran sama dengan 360 derajat.
- b) Roda mobil: roda mobil melakukan satu putaran penuh setiap kali berputar 360 derajat, atau ketika titik tertentu pada roda kembali ke posisi awal.
- c) Bumi mengelilingi matahari: Bumi melakukan satu putaran penuh mengelilingi matahari 360 derajat, yang memakan waktu sekitar 365 hari (1 tahun).

Dengan aturan ini, bahwa satu putaran memiliki 360 derajat, maka dapat dibuat sebuah sub program **segiBanyak** dengan parameter **jlhSegi** dan **pjgSisi** dalam **mm**. Jika jumlah segi = 6, maka sudut yang diperlukan = $360/60$ derajat. Jika jumlah segi = 8, maka sudah yang diperlukan = $360/8$ derajat.

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

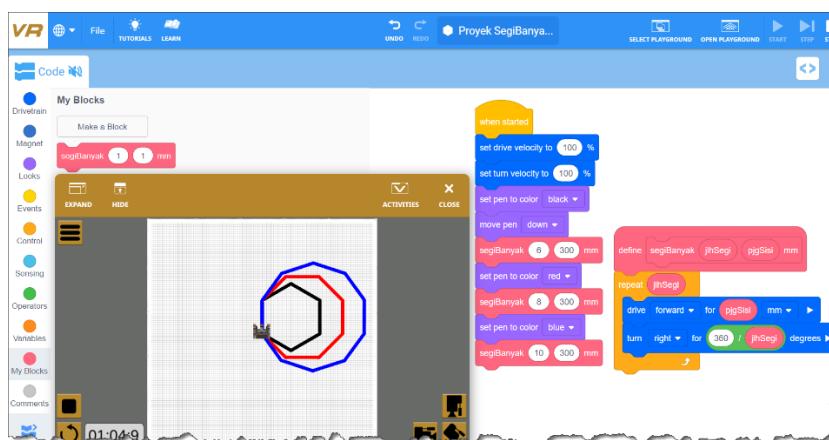
Sub program segiBanyak sebagai berikut :

Define segiBanyak [jlhSegi] [pjgSisi] mm

- Repeat jlhSegi //jumlah sisi yang akan dibuat
 - Drive forward for pjgSisi mm
 - Turn right for $360/jlhSegi$ degrees //1 putaran = 360 derajat

Program Utama sebagai berikut :

- When Started
- Set drive velocity to 100%
- Set turn velocity to 100%
- Set pen to color black
- Move pen down
- **segiBanyak [6] [300] mm**
- Set pen to color red
- **segiBanyak [8] [300] mm**
- Set pen to color blue
- **segiBanyak [10] [300] mm**



Gambar 55 Proyek SegiBanyak menggunakan Block

Ketentuan Kompetisi

“Menggambar Nama/Lambang/Logo Tim Menggunakan Robot VR VEX”:

1. Tim Peserta:
 - Kompetisi khusus kelas X dan XI SMA Santa Angela untuk tim kelompok, dengan setiap tim terdiri dari 2-4 peserta.
 - Setiap peserta hanya boleh menjadi anggota dalam satu tim saja.
2. Pendaftaran:
 - Setiap tim harus mendaftarkan diri sebelum batas waktu pendaftaran yang telah ditentukan.
 - Tim harus memberikan informasi lengkap mengenai nama tim, anggota tim, dan informasi kontak yang valid (HP dan email).
3. Alat dan Bahan:
 - Setiap tim akan diberikan akses ke Robot VR VEX yang sama untuk digunakan selama kompetisi.
 - Tim dapat menggunakan peralatan tambahan seperti pena, kertas, atau alat bantu lainnya yang sesuai.
4. Tema Kompetisi:
 - Tema kompetisi menggambar “Nama Tim” atau “Lambang Tim” atau “Logo Unik Tim”.
 - Tim diharapkan untuk membuat gambar yang mencerminkan tema dengan kreativitas dan originalitas.
5. Waktu Kompetisi:
 - Waktu yang disediakan untuk setiap tim dalam menggambar menggunakan Robot VR VEX. (60 menit)
6. Aturan Teknis:
 - Gunaan perangkat lunak VR VEX untuk menggambar.
 - Playground yang digunakan ART CANVAS
7. Penilaian:
 - Kriteria penilaian mencakup kreativitas, kesesuaian dengan tema, kerapuhan, dan ketepatan waktu.
8. Tim Juri:
 - Tim juri untuk kompetisi per kelas adalah Pengajar dan asisten
 - Tim juri untuk kompetisi final antar kelas adalah seluruh pengajar
9. Pengumuman Pemenang:
 - Pemenang kompetisi per kelas diumumkan saat akhir sesi workshop
 - Pemenang kompetisi final antar kelas diumumkan saat akhir sesi workshop
10. Hadiah:
 - Seluruh peserta mendapatkan Sertifikat Elektronik yang dapat diunduh di bit.ly/Sertif_StAngela2023
 - Hadiah satu pemenang per kelas berupa sertifikat dan hadiah dari FIT Maranatha
 - Hadiah satu pemenang final antar kelas berupa voucher belanja
11. Etika Peserta, Juri dan Panitia:
 - Tim peserta diharapkan untuk menjunjung tinggi etika dalam berkompetisi dan menghormati peserta lain serta juri.
 - Panitia berhak membatalkan kompetisi jika terdapat alasan yang sah dan di luar kendali.
 - Panitia berhak melakukan perubahan pada ketentuan kompetisi jika diperlukan dengan memberikan pemberitahuan sebelumnya kepada peserta.

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

Formulir Evaluasi Pelatihan Pemrograman Visual Menggunakan VR VEX (Nama Peserta akan digunakan untuk pembuatan Sertifikat Elektronik)

Nama Peserta: _____
Kelas : _____
Email : _____
HP : _____
Tanggal Pelatihan: _____

Instruksi: Silakan berikan penilaian dan tanggapan Anda terhadap setiap pertanyaan di bawah ini dengan memberi tanda centang (✓) atau menuliskan komentar pada kolom yang disediakan.

1. Bagaimana pendapat Anda tentang kesesuaian materi pelatihan dengan tingkat pengetahuan dan kebutuhan peserta?
 - Sangat Sesuai
 - Sesuai
 - Cukup Sesuai
 - Tidak Sesuai
 - Tidak Tahu
2. Seberapa jelas dan terstruktur penyampaian materi pelatihan?
 - Sangat Jelas dan Terstruktur
 - Jelas
 - Cukup Jelas
 - Kurang Jelas
 - Tidak Tahu
3. Bagaimana pendapat Anda tentang kualitas dan kecukupan materi presentasi atau modul yang digunakan?
 - Sangat Memuaskan
 - Memuaskan
 - Cukup Memuaskan
 - Tidak Memuaskan
 - Tidak Tahu
4. Apakah Anda merasa tertarik dan termotivasi selama mengikuti pelatihan ini?
 - Sangat Tertarik dan Termotivasi
 - Tertarik dan Termotivasi
 - Cukup Tertarik
 - Kurang Tertarik
 - Tidak Tahu
5. Bagaimana penilaian Anda terhadap kemampuan fasilitator dalam menyampaikan materi dan menjawab pertanyaan peserta?
 - Sangat Kompeten
 - Kompeten
 - Cukup Kompeten
 - Kurang Kompeten
 - Tidak Tahu
6. Seberapa bermanfaatkah kegiatan praktik menggunakan VR VEX dalam pemahaman dan penguasaan pemrograman visual?
 - Sangat Bermanfaat
 - Bermanfaat
 - Cukup Bermanfaat
 - Kurang Bermanfaat
 - Tidak Tahu
7. Bagaimana penilaian Anda terhadap durasi waktu pelatihan? Apakah cukup atau perlu diperpanjang?
 - Cukup
 - Perlu Diperpanjang

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

- Tidak Tahu
8. Seberapa baik suasana dan fasilitas pelatihan yang disediakan?
- Sangat Baik
 - Baik
 - Cukup Baik
 - Kurang Baik
 - Tidak Tahu
9. Apakah ada masukan atau saran lain yang ingin Anda berikan untuk perbaikan dan pengembangan pelatihan di masa mendatang?
- (Kolom Komentar)
10. Apakah Anda tertarik untuk melanjutkan studi ke perguruan tinggi di bidang Teknologi Informasi?
- Ya
 - Mungkin
 - Tidak
11. Apakah Anda tertarik untuk mengikuti kegiatan pelatihan serupa atau lanjutan di masa mendatang?
- Sangat Tertarik
 - Tertarik
 - Kurang Tertarik
 - Tidak Tertarik

Terima kasih atas partisipasi Anda dalam mengisi formulir evaluasi ini. Umpan balik dari Anda sangat berarti bagi kami untuk meningkatkan kualitas pelatihan di masa mendatang.

Tanda Tangan: _____ Tanggal: _____

PEMROGRAMAN VISUAL MENGGUNAKAN ROBOT VR VEX

Formulir Pendaftaran Kompetisi Kelompok Menggambar Menggunakan VR Vex

Judul Kompetisi: Kompetisi Menggambar Nama/Lambang/Logo Kelompok Menggunakan VR Vex

Tanggal Kompetisi: 09 / 10 Agustus 2023

Nama Kelompok: _____

Anggota Kelompok:

1. Nama Peserta: _____ Kelas: _____
2. Nama Peserta: _____ Kelas: _____
3. Nama Peserta: _____ Kelas: _____
4. Nama Peserta: _____ Kelas: _____

Keterangan:

- Setiap kelompok harus terdiri dari 2 hingga 4 peserta.

Persetujuan dan Syarat

Dengan menandatangani formulir ini, kami, peserta dari kelompok yang disebutkan di atas, menyatakan bahwa kami telah membaca, memahami, dan menyetujui syarat dan ketentuan dari kompetisi "Kompetisi Menggambar Menggunakan VR Vex".

1. Kami bersedia untuk mengikuti aturan dan petunjuk yang telah ditetapkan oleh panitia kompetisi.
2. Kami sepenuhnya memahami bahwa hasil keputusan dewan juri adalah mutlak dan tidak dapat diganggu gugat.
3. Kami memberikan izin kepada panitia untuk menggunakan hasil karya kami untuk keperluan promosi dan dokumentasi kegiatan.
4. Kami setuju bahwa panitia berhak membatalkan pendaftaran kami jika ditemukan adanya pelanggaran atau informasi yang tidak benar pada formulir ini.

Tanda Tangan Anggota Kelompok

1. Nama dan tanda tangan	2. Nama dan tanda tangan
3. Nama dan tanda tangan	4. Nama dan tanda tangan

Terima kasih atas partisipasi Anda dalam kompetisi ini. Semoga sukses dan selamat menggambar menggunakan VR Vex!