

**DESIGN *IMMERSIVE TOOL* PADA GAME SEPEDA ELEKTRIK
BERBASIS STATISTIK DAN KOMUNIKASI
DATA ASINKRON**

SKRIPSI

Oleh:

MAYA SITI MAYSAROH

NIM. 11650100



**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2016**

**DESIGN *IMMERSIVE TOOL* PADA GAME SEPEDA ELEKTRIK
BERBASIS STATISTIK DAN KOMUNIKASI DATA ASINKRON**

SKRIPSI



**Diajukan Kepada:
Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN)
Maulana Malik Ibrahim Malang
Untuk Memenuhi Salah Satu Persyaratan Dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)**

**Oleh :
Maya Siti Maysaroh
NIM. 11650100**

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2016**

**DESIGN IMMERSIVE TOOL PADA GAME SEPEDA ELEKTRIK
BERBASIS STATISTIK DAN KOMUNIKASI DATA ASINKRON**

SKRIPSI

Oleh :

MAYA SITI MAYSAROH

NIM. 11650100

Telah disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

YUNIFA MIFTACHUL ARIF, M.T

NIP. 19830616 201101 1 004

Dr. M. Faisal, M.T

NIP. 19740510 200501 1 007

Tanggal, 13 Mei 2016

**Mengetahui,
Ketua Jurusan Teknik Informatika**

Dr. Cahyo Crysdian, M.CS

NIP. 19740424 200901 1 008

DESIGN IMMERSIVE TOOL PADA GAME SEPEDA ELEKTRIK BERBASIS STATISTIK DAN KOMUNIKASI DATA ASINKRON

SKRIPSI

Oleh :

**MAYA SITI MAYSAROH
NIM. 11650100**

**Telah Dipertahankan di Depan Dewan Penguji Skripsi dan
Dinyatakan Diterima Sebagai Salah Satu Persyaratan Untuk
Memperoleh Gelar Sarjana Komputer (S.Kom)**

Tanggal, 13 Mei 2016

Susunan Dewan Penguji

Tanda Tangan

- | | | |
|------------------|---|-----|
| 1. Penguji Utama | : <u>Hani Nurhayati, M.T</u>
NIP. 19780625 200801 2 006 | () |
| 2. Ketua | : <u>Fachrul Kurniawan, M.MT</u>
NIP. 19771020 200901 1 001 | () |
| 3. Sekretaris | : <u>Yunifa Miftachul Arif, M.T</u>
NIP. 19830616 201101 1 004 | () |
| 4. Anggota | : <u>Dr. Muhammad Faisal, M.T</u>
NIP. 19740510 200501 1 007 | () |

**Mengetahui dan Mengesahkan
Ketua Jurusan Teknik Informatika**

**Dr.Cahyo Crysdian, M.CS
NIP. 19740424 200901 1 008**

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : MAYA SITI MAYSAROH

NIM : 11650100

Fakultas/ Jurusan : Sains dan Teknologi / Teknik Informatika

Angkatan tahun/semester : **DESIGN IMMERSIVE TOOL PADA GAME SEPEDA ELEKTRIK BERBASIS STATISTIK DAN KOMUNIKASI DATA ASINKRON**

Menyatakan dengan sebenar-benarnya bahwa hasil penelitian saya ini tidak terdapat unsur-unsur penjiplakan karya penelitian atau karya ilmiah yang pernah dilakukan atau dibuat oleh orang lain, kecuali yang secara tertulis dikutip dalam naskah ini dan disebutkan dalam sumber kutipan dan daftar pustaka.

Apabila ternyata hasil penelitian ini terbukti terdapat unsur-unsur penjiplakan, maka saya bersedia untuk mempertanggungjawabkan, serta diproses sesuai peraturan yang berlaku.

Malang, 13 Mei 2016

Yang membuat pernyataan

Maya Siti Maysaroh

NIM. 11650100

MOTO

"Tiada Kesuksesan tanpa ada kemauan dan kerja keras"

Proses menuju tangga kesuksesan adalah Bersabar,

Ikhlas, pantang menyerah dan tawakal

~Maya Siti Maysaroh~

HALAMAN PERSEMBAHAN

Bacalah dengan menyebut nama Tuhanmu

Dia telah menciptakan manusia dari segumpal darah Bacalah, dan Tuhanmulah yang maha mulia yang mengajar manusia dengan pena,

Dia mengajarkan manusia apa yang tidak diketahuinya (QS. Al- 'Alaq 1-5)

Maka nikmat Tuhanmu yang manakah yang kamu dustakan ? (QS. Ar- Rahman 13)

Niscaya Allah akan mengangkat (derajat) orang-orang yang beriman diantaramu dan orang-orang yang diberi ilmu beberapa derajat (QS. Al- Mujadilah 11)

Ya Allah,

Waktu yang sudah kujalani dengan jalan hidup yang sudah menjadi takdirku, sedih, bahagia, dan bertemu orang-orang yang memberiku sejuta pengalaman bagiku, yang telah memberi warna-warni kehidupanku. Ku bersujud dihadapan Mu, Engkau berikan aku kesempatan untuk bisa sampai di penghujung awal perjuanganku

Segala Puji Bagi Mu ya Allah,

Alhamdulillah....

Sujud Syukurku persembahkan kepadamu Allah yang mulia atas takdir Mu telah kau jadikan aku manusia yang senantiasa berpikir, berilmu, dan beriman. Ikhlas dan bersabar dalam menjalani kehidupan ini, Semogha keberhasilan ini menjadi satu langkah awal bagiku untuk meraih mimpi.

Lantunan Al-fatehah beriring Shalawat dalam silahku merintih, menadahkan doa dalam syukur yang tiada terkira, terima kasihku untukmu. Kupersembahkan sebuah

karya kecil buat Bapak Emak yang maya Sayangi, yang tiada pernah hentinya selama ini memberiku semangat, doa, dorongan nasehat, dan kasih sayang serta pengorbanan yang tak tergantikan hingga aku selalu kuat menjalani setiap rintangan yang ada didepanku.

Dalam silah di lima waktu mulai fajar terbit hingga terbenam seraya tanganku menadah "Ya Allah Terimakasih telah kau tempatkan aku diantara kedua malaikatmu yang setiap waktu ikhlas menjagaku, mendidikku, membimbingku dengan baik, ya Allah berikanlah balasan setimpal Syurga firdaus untuk mereka dan jauhkanlah mereka nanti dari panasnya sengat hawa api nerakamu..."

Untukmu Bapak (Abdullah) Emak (Juminah)... Terimakasih

We always Loving you (ttd Si Bungsu)

Dalam setiap langkahku aku berusaha mewujudkan harapan-harapan yang kalian impikan didiriku, meski belum semua terwujud, insallah atas doa dan restu semua mimpi itu akan terjawab di masa penuh kehangatan nanti. Untuk itu kupersembahkan ungkapan terimakasihku

Kepada kakakku yang sering aku panggil dalam singkatan dari inisial awal nama teteh sama Aa "NheJomy" (**Ceu Nani, Ceu Halimah, Ceu Eha, Aa Jamal, Ceu Omah, dan Ceu Yoyoh**) terimakasih yang selalu ngasih doa dan kabar semangat nya, akhirnya si bungsu ini bisa wisuda juga ^_^ untuk keponakan kecilku yang selalu memberi tawa yang lepas (**Teteh Iff, Akmal, Najwa, Fahmi, Nizam, De' Arsil, De' fadhil**), Mogha kalian bisa menuntut ilmu lebih tinggi dan menjadi anak sholeh sholehah

Hidupku terlalu berat untuk mengandalkan diri sendiri tanpa melibatkan bantuan Allah dan orang lain, kepada orang yang sudah berjasa dalam hidup, menyalurkan ilmunya, mengamalkan dan membimbingku sampai bisa selesai skripsi ini. Terimakasihku kepada

:

Dosen Bimbingan Pak Faisal, Pak Yunifa, dan pak Syauqi, yang mengajari banyak hal, Terimakasih juga untuk Mas Elang dan Mba April yang sudah meluangkan waktunya untuk membantu dalam pengerjaan skripsi ini, dan juga kepada Mas Ulin, Mas Rahmat, dan Mas Misbah (Fisika '11) dan kepada Mas Oksali, mas Haris, dan Mas Pogal (Informatika 2010) yang juga sering membimbingku dalam pengerjaan penelitian. teman-teman seperjuanganku Integer 2011 Untuk sahabatku Lafnhidita Farosanti (Ocha) yang selalu nemenin dan nyemangatin aku, indra, Emil, Hafidz. Alifian.

Dan tak lupa Kepada seseorang yang sudah menjadi Bapak kedua dalam hidupku di Malang yaitu Abah chusaini el hafidz, dan ummi yang sudah memberikan banyak ilmu dan pengalaman yang tak bisa aku balas jasa mereka, dan untuk teman - teman di penjara suci khususnya di kamer Khodijah, Untuk dua mamah, (Mba binti & mayang), Oby, Mba pas, Mba dhina, Mba Asri, Mba Uji, terimakasih atas segala bantuan dan motivasinya. Mogha kalian bisa meraih mimpi yang belum terwujudkan.

Kepada kementrian Agama RI yang sudah banyak membantu dalam memfasilitasi kebutuhan material di Malang, dan kepada Mushohih yang selalu sabar membimbing Pak nasichuddin, Pak Zaiz, Teman-teman Css Mora yang dari awal masuk UIN sampai sekarang yang selalu nemenin dalam suka cita yang sering di panggil Spectrum Eigen (2011),

Spesial !!!

Untuk seseorang yang masih menjadi rahasia Illahi, tak bisa ku tulis nama ini karena masih di rahasikan, selalu yakin dan percaya bahwa allah sudah menentukan takdir masing-masing ada satu nama yang selalu kusbut dalam benih-benih doa yang akan menjadi impian di masa depan, mogha Allah mempertemukan di waktu yang tepat dengan rahmat
Nga

Untuk ribuan tujuan yang harus dicapai, untuk jutaan impian yang akan dikejar, untuk sebuah pengharapan, agar hidup jauh lebih bermakna, hidup tanpa mimpi ibarat arus

*sungai. Mengalir tanpa tujuan. Teruslah belajar, berusaha, dan berdoa untuk
menggapainya.*

Jatuh berdiri lagi. Kalah mencoba lagi. Gagal Bangkit lagi.

Never give up!

Sampai Allah SWT berkata “waktunya pulang”

Terimakasih kuucapkan..

*Atas segala kekhilafan salah dan kekuranganku,
kurendahkan hati serta diri menjabat tangan meminta beribu-ribu kata maaf tercurah.*

Skripsi ini kupersembahkan.

“NHEJOMY”



KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Segala puji bagi Allah SWT tuhan semesta alam, karena atas segala rahmat dan karunia-Nya kepada penulis sehingga penulis mampu menyelesaikan skripsi dengan judul “*Design Immersive Tools* pada Game Sepeda Elektrik Berbasis Statistik dan Komunikasi Data Asinkron” dengan baik dan lancar. Shalawat dan salam selalu tercurah kepada tauladan terbaik kita Nabi Agung Muhammad SAW yang telah membimbing umatnya dari zaman kegelapan dan kebodohan menuju cahaya islam yang terang *rahmatan lil alamiin* ini.

Dalam penyelesaian skripsi ini, banyak pihak yang telah memberikan bantuan baik secara moril, nasihat dan semangat maupun materiil. Atas segala bantuan yang telah diberikan, penulis ingin menyampaikan doa dan ucapan terimakasih yang sedalam-dalamnya kepada :

1. Prof. DR. H. Mudjia Raharjo, M.Si, selaku rektor UIN Maulana Malik Ibrahim Malang beserta seluruh staf. Dharma Bakti Bapak dan Ibu sekalian terhadap Universitas Islam Negeri Malang turut membesarkan dan mencerdaskan penulis.
2. Dr. Hj. Bayyinatul M., drh., M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang beserta seluruh staf. Bapak dan ibu sekalian sangat berjasa memupuk dan menumbuhkan semangat untuk maju kepada penulis.

3. Bapak Dr. Cahyo Crysdiyan, selaku Ketua Jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang, yang sudah memberi banyak memberi pengetahuan, inspirasi dan pengalaman yang berharga.
4. Bapak Bapak Yunifa Miftachul Arif, M.T, selaku dosen pembimbing I yang senantiasa memberi masukan dan nasihat serta petunjuk dalam penyusunan skripsi ini.
5. Bapak Dr. M Faisal, M.T, selaku dosen pembimbing II yang telah meluangkan waktu untuk membimbing, memotivasi, mengarahkan dan memberi masukan kepada penulis dalam pengerjaan skripsi ini hingga akhir.
6. Segenap Dosen Teknik Informatika yang telah memberikan bimbingan keilmuan kepada penulis selama masa studi.
7. Para peneliti yang telah mengembangkan Game dengan Engine *Unity3d* yang menjadi acuan penulis dalam pembuatan skripsi ini. Serta semua pihak yang telah membantu yang tidak bisa disebutkan satu satu. Terimakasih banyak.

Berbagai kekurangan dan kesalahan mungkin pembaca temukan dalam penulisan skripsi ini, untuk itu penulis menerima segala kritik dan saran yang membangun dari pembaca sekalian. Semoga apa yang menjadi kekurangan bisa disempurnakan oleh peneliti selanjutnya dan semoga karya tulis ini bisa bermanfaat dan menginspirasi bagi kita semua. Amin.

Wassalamualaikum Wr. Wb.

Malang, 13 Mei 2016

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
ABSTRAK	xviii
ABSTRACT	xix
مستخلص البحث	xx
BAB I PENDAHULUAN	1Error! Bookmark not defined.
1.1 Latar Belakang	Error! Bookmark not defined.
1.2 Identifikasi Masalah	Error! Bookmark not defined.
1.3 Tujuan Penelitian	Error! Bookmark not defined.
1.4 Batasan Masalah	4
1.5 Manfaat Penelitian	4
1.6 Metode Penelitian	4
1.5 Sistematika Penelitian	7
BAB II KAJIAN PUSTAKA	9
2.1 Immersive Tools	9
2.2 Statisstik dan Multisensor	10
2.3 Komunikasi data serial yang menggunakan Asinkron	14
2.4 Mikrokontroller Arduino Duemilanove	16

2.4.1 <i>Arduino Development Environment</i>	17
2.5 ADC	18
2.6 Sensor	19
2.6.1 <i>LED infrared</i> dan <i>photodiode</i>	19
2.6.2 Sensor Potensiometer	20
BAB III ANALISIS DAN PERANCANGAN	Error! Bookmark not defined.
3.1 Kebutuhan Sistem	23
3.1.1 Kebutuhan <i>Platform</i>	23
3.2.1 Deskripsi Sistem.....	24
3.2.1.1 Komunikasi data serial menggunakan USB A?B Asinkron.....	24
3.2.1.2 Komunikasi Asinkron.....	27
3.2.1.2 ADC.....	34
3.2.1.3 Prosedur pengisian program mikrokontroler menggunakan Arduino...35	
3.2.1.4 Sensor Kecepatan menggunakan LED inframerah dan photodiode.....37	
3.2.1.5 Sensor Kemudi.....	39
3.2.1.6 Statistika Data.....	42
3.3.1 Kalibrasi Sensor.....	44
3.2 Perancangan alur pproses	55
3.2.1 Perancangan alur proses pengiriman data multisensory.....	55
BAB IV HASIL DAN PEMBAHASAN	Error! Bookmark not defined.
4.1 Hasil uji coba.....	63
4.1.1 Hasil Kalibrasi.....	63
4.1.2 Hasil Uji komunikasi serial Asinkron	67
4.1.3 Hasil Uji Coba Statistik data	69
4.2 Koneksi data multisensor ke game	75
4.3 Integrasi Dalam Islam	78
BAB V KESIMPULAN DAN SARAN	80
5.1 Kesimpulan	80
5.2 Saran.....	81

DAFTAR PUSTAKA	82
----------------------	----

DAFTAR GAMBAR

Gambar 2.1 Immersive Tools	9
Gambar 2.2 Immersive Tools Cycle.....	10
Gambar 2.3 Potensiometer Belokan Kanan dan Kiri	11
Gambar 2.4 Ilustrasi	12
Gambar 2.5 Tanda Sensor Pada Roda	13
Gambar 2.6 Arduino Duemilanove	18
Gambar 2.7 IDE Arduino	19
Gambar 2.8 Inframerah dan Photodiode.....	22
Gambar 2.9 Aplikasi Sensor Garis dengan <i>infrared</i> dan <i>photodiode</i>	22
Gambar 2.10 Ilustrasi Rangkaian Sensor	23
Gambar 2.11 Sensor Potensiometer.....	24
Gambar 3.1 Komunikasi Arduino Dengan Game	28
Gambar 3.2 Kabel USB A/B	29
Gambar 3.3 Kaki Soket USB.....	30
Gambar 3.4 Proses Pengiriman Data Serial Asinkron.....	33
Gambar 3.5 Proses Penerima Data Serial Asinkron.....	34
Gambar 3.6 Paket UART : 1 mulai bit, 8 bit data, 1 bit paritas dan 1 bit berhenti	35
Gambar 3.7 Data Titik Pengambilan Sampel oleh Penerima	36
Gambar 3.8 Konfigurasi kaki-kaki Arduino.....	37
Gambar 3.9 <i>Integrated Development Environment</i>	38
Gambar 3.10 Bentuk LED Infra Merah.....	42
Gambar 3.11 Kurva Tanggapan Frekuensi Sensor Phodiode.....	43
Gambar 3.12 Peletakan Sensor Saat Mendeteksi Warna Hitam dan Warna Putih Pada Roda Simulasi	44
Gambar 3.13 Potensiometer Pada Simulasi Sepeda Untuk Sensor Kemudi Dan Pengereman	45
Gambar 3.14 Potensiometer dengan belokan kanan dan kiri	46
Gambar 3.15 Tanda Sensor pada roda.....	48
Gambar 3.16 Rancangan Alur Proses sensor data	59

Gambar 3.17 Proses pengolahan data multisensor	60
Gambar 3.18 Diagram Blok.....	61
Gambar 4.1 Hasil Uji Sensor pada game.....	Error! Bookmark not defined.
Gambar 4.2 Hasil Uji Sensor pada game.....	Error! Bookmark not defined.
Gambar 4.3 Hasil Uji Sensor pada game.....	72



DAFTAR TABEL

Tabel 4.1 Nilai Hasil Uji Kecepatan.....	69
Tabel 4.2 Nilai Hasil Uji Kemudi.....	70
Tabel 4.3 Nilai Hasil Uji Rem.....	Error! Bookmark not defined.
Tabel 4.4 Hasil Uji Percobaan dan Pengamatan.....	72



ABSTRAK

Maysaroh, Maya. Siti. 2016. **Design Immersive Tool Pada Game Sepeda Elektrik Berbasis Statistik Dan Komunikasi Data Asinkron**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Pembimbing : (I) Yunifa Miftachul Arif, MT, (II) Dr. M. Faisal, M.T

Kata Kunci: Immersive Tool, Multisensor

Game merupakan media hiburan yang sangat diminati hampir semua lapisan masyarakat. Pada era sekarang tak sedikit *Game* yang kurang mendidik anak-anak untuk mengembangkan bakat mereka, hal ini tak luput dari arahan orang tua, maka dari itu penulis mencoba untuk menciptakan Simulasi *Game Sepeda Elektrik* dengan menggunakan *Immersive Tools* ini di mainkan oleh anak – anak dan dewasa.

Tools Immersive Tools adalah sebuah alat *control* yang dipergunakan untuk mengendalikan objek simulasi didalam dunia virtual. Sehingga *Immersive Tools* membantu berjalannya *game* simulasi sepeda elektrik ini sehingga player bisa membelokkan , mempercepat dan mengerem sepeda.

Game sebagai sarana hiburan yang ingin menerapkan *immersive Tools* Multisensor pada game sepeda sehingga menarik untuk di mainkan oleh pemain. pada game memakai komunikasi data serial Asinkron,yaitu komunikasi yang mengantarkan data digital secara bit per bit secara bergantian melalui media *interface* serial.

ABSTRACT

Panji, Galih. 2015. **Raden Fatah Historical Game Using Fuzzy As NPC Control Behavior**. Thesis. Informatics Engineering Department of Science and Technology Faculty Islamic State University Maulana Malik Ibrahim Malang.

Adviser : (I) Dr. M. Faisal, M.T, (II) Fachrul Kurniawan, M, M.T

Keyword: Fuzzy Sugeno, NPC, Game, Educational Game.

Game is an entertainment media that has much enthusiasts from all segments of society. The quality of games are determined by several aspect, start from artificial intelligence, content presented, and others. Artificial Intelligence is considerably needed by a game in order to create an action and reaction to achieve realistic level expected, especially for Non Playable Player (NPC).

Interesting educational game is one breakthrough game that provides an educational content as an interesting learning media. Raden Fatah historical game is educational game with first player shooter (FPS) genre desktop base using Unity3d Engine. Player guided to complete a mission that represents the history how Raden Fatah build the first Islamic kingdom in Java.

NPC are controlled by artificial intelligence to do an action and reaction response to player. In this research, Fuzzy sugeno algorithm is used on artificial intelligence method to obtain suitable behavior. The testing did in desktop.

مستخلص البحث

Panji, Galih. 2015. **Raden Fatah Historical Game Using Fuzzy As NPC Control Behavior**. Thesis. Informatics Engineering Department of Science and Technology Faculty Islamic State University Maulana Malik Ibrahim Malang.

Adviser : (I) Dr. M. Faisal, M.T, (II) Fachrul Kurniawan, M, M.T

اللعبة هي وسائل الاعلام وسائل الترفيه التي لديها الكثير من المتحمسين من جميع شرائح المجتمع. يتم تحديد نوعية الألعاب التي كتبها عدة جوانب، بداية من الذكاء الاصطناعي، والمحتوى المقدم، وغيرها. هناك حاجة إلى الذكاء الاصطناعي إلى حد كبير من قبل لعبة من أجل خلق الفعل ورد الفعل المتوقع لتحقيق مستوى واقعي، وخاصة بالنسبة لل (NPC) Non Playable Character لعبة تعليمية للاهتمام هو لعبة واحدة انفراجة التي توفر المحتوى التعليمي باعتبارها وسائل الإعلام التعلم مثيرة للاهتمام. اللعبة التاريخية رادين فتح هي لعبة تعليمية مع First Player Shooter (FPS) سطح المكتب نوع قاعدة باستخدام Unity3d Game Engine . لاعب الموجهة لإكمال المهمة التي تمثل التاريخ كيف رادين فتح بناء أول مملكة إسلامية في جاوة. NPC يتم التحكم فيها بواسطة الذكاء الاصطناعي للقيام الفعل ورد الفعل استجابة لاعب. في هذا البحث، Fuzzy Sugeno يستخدم الخوارزمية على طريقة الذكاء الاصطناعي للحصول على السلوك المناسب. اختبار فعل في سطح المكتب

اللعبة هي وسائل الاعلام





BAB I

PENDAHULUAN

1.1. LATAR BELAKANG

Simulasi merupakan merupakan salah satu perkembangan teknologi akhir-akhir ini yang sangat pesat perkembangannya, terutama di bidang computer (Pogal, 2014). Ralph Schroeder (2008) dalam penelitian dunia Virtual menjelaskan Game virtual merupakan dunia berbasis computer yang memungkinkan atau memaksa pengguna untuk memiliki rasa yang hadir dalam lingkungan lain dari satu mereka sebenarnya dalam, dan berinteraksi dengan lingkungan. Dengan hadirnya game virtual yang diciptakan dengan kejadian nyata mempermudah dalam memainkannya selain itu juga kondisi ini memperhatikan kesehatan pemain karena menggunakan ayuhan kaki untuk digerakkan.

Immersive Tools adalah sebuah alat control yang dipergunakan untuk mengendalikan suatu objek didalam dunia virtual. Pemakaian *Immersive Tools* pada suatu game ditujukan untuk memvisualisasikan keadaan seperti sesungguhnya, hal ini dapat menjadikan suatu game lebih menantang dan menarik untuk dimainkan (Pogal 2014). Selain itu juga terdapat simulasi yang bisa dijadikan sebuah permainan untuk memberikan suasana berbeda dalam belajar dalam proses simulasi *game* ini masukan yang dibutuhkan

meliputi berbelok dan kecepatan, kecepatan putaran roda dan belokan sepeda tiruan diambil kemudian divisualisasikan ke dalam permainan.

Game menjadi salah satu pilihan utama mengisi waktu senggang setelah beraktifitas. Citra game dimasyarakat masih dipandang sebagai media yang menghibur dibanding sebagai media pembelajaran. Sifat dasar game yang menantang, membuat ketagihan dan menyenangkan bagi mereka yang menyukai permainan *modern*.

Dalam hal ini peneliti membuat game elektrik dengan menggunakan *Immersive Tools* sehingga pemain game ini bisa bermain bersepeda mengemudi dengan membelokkan kanan dan kiri, dan mengerem. Game elektrik ini simulasi game yang lebih sederhana dari penelitian sebelumnya sehingga peneliti lebih mudah dalam mengembangkannya dan diharapkan bisa dikembangkan pada penelitian selanjutnya.

1.2. IDENTIFIKASI MASALAH

Berdasarkan Latar Belakang yang telah dikemukakan di atas dapat diidentifikasi masalahnya sebagai berikut:

1. Bagaimana model lingkup *Immersive Tools* untuk menyajikan game elektrik seperti berbelok, mengerem, mempercepat ayuhan ?
2. Apa komunikasi data yang digunakan dalam mentransmisikan data dari *Immersive Tools* ?

1.3. TUJUAN PENELITIAN

Tujuan penelitian ini adalah untuk membuat control Immersive Tools Game sepeda yang diharapkan bisa memberikan suasana yang berbeda dalam bermain game pada umumnya, dan juga permainan bisa menjadi lebih menarik untuk di mainkan.

1.4. BATASAN MASALAH

Untuk menjaga fokus dari penelitian ini, maka beberapa batasan yang diberikan adalah sebagai berikut :

- a. Game ini dimainkan oleh Single Player
- b. Platform yang di gunakan adalah windows 7.
- c. Game engine yang di gunakan adalah Unity 3D Free.
- d. Game sepeda elektrik menggunakan Immersive Tools diharapkan pemain bisa memberikan suasana yang berbeda baik dari segi mengayuh dan berbelok.
- e. Game dihubungkan ke PC dengan menggunakan Arduino Uno.

1.5. MANFAAT PENELITIAN

Manfaat yang di harapkan dari penelitian diantaranya adalah :

- a. Game sebagai sarana bermain (menghibur) pemain
- b. Game sebagai simulasi untuk dijadikan game sepeda asli pada umumnya.
- c. Game ini juga bisa dijadikan sebagai pembelajaran

1.6. METODE PENELITIAN

Tahapan metode penelitian yang dilakukan dalam penelitian ini adalah dengan langkah-langkah berikut ini :

a. Analisis

Pada tahap ini yaitu proses menganalisa setiap permasalahan yang akan muncul dalam pembuatan system ini, diantaranya:

1. Study Literatur

- a. Pengumpulan informasi tentang *immersive tools*.
- b. Pengumpulan software yang diperlukan.
- c. Pengumpulan informasi untuk spesifikasi hardware yang dibutuhkan.

2. Perumusan Masalah

Menganalisa hardware dan software yang digunakan user, Kebutuhan User, dan waktu pembuatan system.

3. Analisis literature

1. Identifikasi dan Desain Sistem
2. Menganalisa *Immersive Tools*, metode asinkron dan bahasa pemrograman C pada mikrokontroler.
3. Menganalisa system *Immersive Tools* pada Game sepeda.

4. Perancangan System

Pada tahap ini membahas tentang desain system pada game, meliputi:

1. Pembuatan desain fungsi yang digunakan.

Pembuatan desain fungsi yang digunakan dalam *immersive tools*.

2. Pembuatan desain *input*.

3. Pembuatan desain *output*.
4. Pembuatan desain proses.
5. Pembuatan desain sepeda dan penempatan sensor yang dibutuhkan.
6. Pembuatan Perangkat.
 1. Merancang desain *Immersive Tools*.
 2. Menyiapkan peralatan yang diperlukan seperti *Mikrokontroller*, *Sensor Photodiode*, *Potensiometer*, *Project board*, *Cable*, *Rotary DC Fan*.
7. Pembuatan Software.
 1. Pembuatan game berupa dunia virtual sebagai medan.
 2. Penggabungan control *Immersive Tools* dengan game melalui mikrokontroller.
 3. Finishing visualisasi.
 4. Implementasi program ke laptop/PC.
8. Tahap uji coba dan evaluasi.

Uji coba dan evaluasi ini di lakukan sampai sistem benar siap digunakan, dan kekurangan yang terjadi akan diperbaiki dalam lingkup batasan masalah. Evaluasinya penelitian ini dilakukan untuk mengetahui system yang digunakan apa masih diperbaiki lagi. Dengan seperti itu bisa di kembangkan untuk penelitian selanjutnya.

9. Dokumentasi.

Penulisan laporan skripsi merupakan dokumentasi keseluruhan pelaksanaan penelitian. dokumentasi penelitian bermanfaat untuk penelitian dan pengembangan lebih lanjut.

1.7. SISTEMATIKA PENULISAN

➤ BAB 1 PENDAHULUAN

Bab ini berisi pembahasan tentang latar belakang masalah, identifikasi masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

➤ BAB II KAJIAN PUSTAKA

Pada Bab ini menjelaskan tentang teori-teori yang terkait dengan permasalahan yang akan diambil, di ambil dari penelitian yang sebelumnya untuk dikembangkan lebih lanjut sehingga ada keterikatan dengan penelitian ini.

➤ BAB III ANALISIS DAN PERANCANGAN

Pada Bab ini menjelaskan tentang perancangan sepeda yang akan dibuat yang meliputi perancangan perangkat, yaitu penempatan sensor kecepatan, kemudi dan juga pengereman. Sedangkan IDE yang digunakan sampai pada pembuatan game.

➤ BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi pembahasan tentang pengiriman data dari sensor sampai diolah oleh mikrokontroller yang kemudian dikirim ke computer menggunakan komunikasi data serial asinkron menggunakan USB serta

akan dilakukan pengujian pada game yang dimana untuk mengetahui apakah sensor-sensor bisa berjalan dengan baik dan juga kalibrasi sensor untuk memulai awal permainan.

➤ **BAB V PENUTUP**

Bab ini berisi kesimpulan dan saran sehingga diharapkan game ini bisa dikembangkan pada penelitian selanjutnya

➤ **DAFTAR PUSTAKA**

Di dalam daftar pustaka adalah seluruh materi referensi dalam penelitian skripsi ini, yang dimana tercantumkan dalam bab ini.

➤ **LAMPIRAN**

.Di dalam lampiran ini terdapat data pendukung yang bertujuan untuk melengkapi uraian yang disajikan dalam bagian utama yang akan ditempatkan dibagian ini.

BAB II

KAJIAN PUSTAKA

2.1 *Immersive Tools*

Immersive Tools adalah sebuah alat *control* yang dipergunakan untuk mengendalikan objek simulasi didalam dunia virtual (Pogal, 2014). Sedangkan menurut Triadmadya (2014) mengatakan *Immersive tools* merupakan sebuah alat mengacu pada teknologi yang mengaburkan batas antara dunia fisik dan dunia digital atau simulasi (Triadmadya,Oksali 2014). Dalam membuat *Immersive Tools* pada *game* sepeda ini akan memakai sepeda umumnya, hal ini di tujukan untuk membuat pemain untuk bisa merasa bermain game seperti kejadian bersepeda pada biasanya.



Gambar 2.1 *Immersive Tools*

Pada gambar diatas merupakan gambaran dari immersive Tools pada game simulasi sepeda, seseorang yang memainkan sepeda seperti pada sepeda umumnya, untuk simulationis sendiri didefinisikan sebagai penggunaan alat-alat teknik dan perangkat lunak untuk mereproduksi lingkungan kehidupan nyata atau skenario untuk tujuan pembelajaran.



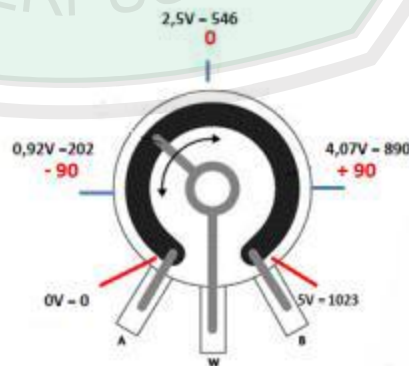
Gambar 2.2 Immersive Tools Cycle

Pada gambar diatas Simulasi Sepeda *Immersive Tools* pemakaian yang mengayuh untuk meningkatkan kecepatan dan juga kemudi untuk berbelok, selain itu pengereman untuk berhenti jika mau berbelok dan jika ada halangan.

2.2 Statistik dan MultiSensor

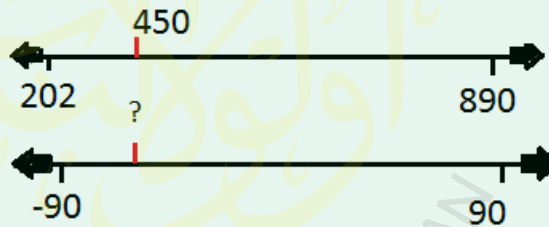
Statistika adalah ilmu yang mempelajari bagaimana merencanakan, mengumpulkan, menganalisis, menginterpretasi, dan mempresentasikan data. Singkatnya, Statistika adalah ilmu yang berkenaan dengan data, sedangkan statistik adalah data, informasi, atau hasil penerapan algoritma statistika pada suatu data. Atau dapat juga sebagai alat pengolah data angka atau metode guna mengumpulkan, mengolah, menyajikan, menganalisis dan menginterpretasikan data statistic.

Dalam *game* ini kegunaan statistika untuk kalibrasi data dari sensor untuk ditampilkan pada monitor di *game*, hal ini ditujukan untuk menampilkan hasil data *real* dari sensor. Maka dari itu untuk tampilan sensor kemudi akan tampil dalam monitor pada sudut belokan di *game*. Berikut ini gambaran sudut dan tegangan dari *potensiometer* yang digunakan untuk sensor kemudi.



Gambar 2.3 Potensiometer dengan belokan kanan dan kiri

Pada gambar ditunjukkan bahwa pada posisi 0° (2,5V-546) maka posisi kemudi tepat lurus, sedangkan pada posisi -90° (0,92V-202) maka kemudi belok kiri dan 90° (4,7V-890) akan belok kanan. Maka dari itu untuk belok kiri nanti dari posisi 0 tegangan akan semakin rendah, begitu juga sebaliknya jika belok kanan maka tegangan akan tambah besar dari posisi 0 dan daya yang digunakan adalah 5V dan 10bit (0-1023) (Ashaulo, 2012). Misal pemain berbelok kekiri yang dimana diketahui bahwa nilai 10bit adalah 0-1023, dan disini pemain berbelok pada titik 450, maka dapat dilakukan dengan cara berikut.



Gambar 2.4 Ilustrasi

Maka dapat dihitung,

$$\text{titikOut} = \text{LowOut} + (\text{valueIn} - \text{LowIn}) + x \frac{\text{highOut} - \text{lowOut}}{\text{highin} - \text{lowin}}$$

dimana :

titikOut = titik hasil

lowOut = titik batas bawah hasil konversi yaitu -90

highOut = titik batas atas hasil konversi yaitu 90

lowIn = titik batas bawah titik asal dalam contoh ini adalah 202

highIn = titik batas atas titik asal dalam contoh ini adalah 890

valueIn = titik asal yang akan dikonversi

Sedangkan penghitungan kecepatan akan diperoleh dari berapa kali sensor membaca titik hitam (0) dan titik putih (1) dalam tiap detik. Maka dari itu akan diketahui kecepatan tiap detik, maka data yang ditampilkan di *serial monitor* merupakan data kecepatan *rotasi* roda, yang diperoleh dari:



Gambar 2.5 tanda sensor pada roda

Pada Gambar diatas dijelaskan bahwa tanda putih (1) berjumlah 7 dan keliling 1,5 meter, sedangkan untuk menghitung kecepatan terdapat pada program dimikrokontroller, yaitu :

$$Speed = \frac{count\ A}{jumlah\ tanda} * 1000 * waktu\ sampling$$

Dimana :

Count A: jumlah baca tanda tiap detik

Jumlah tanda : jumlah data pada roda

1000 : waktu sampling menggunakan detik

Setelah data dalam satuan detik muncul maka nanti pada program mikrokontroler dimasukkan rumus:

$$AVG = x = \frac{Kecepatan\ 1 + \dots + kecepatan\ N}{N}$$

Dimana kecepatan 1- kecepatan N merupakan hasil dari hitung kecepatan tiap detik, dan N merupakan jumlah dari data.

Dimana rps (Rotation persecond)

Dari data diatas dapat disimpulkan kecepatan minimal = 0, sedangkan untuk mencari kecepatan maksimal dapat diinisialisasikan dengan.

```

Max = 0
For(int i=1; i<5; i++)
{
    if (V(i)> max)
    {
        Max=V(i)
    }
}

```

Potongan kode di atas adalah rumus mencari kecepatan maksimal. Jika sudah didapat nilai per detik dari sensor maka akan mencari kecepatan jarak, menggunakan rumus

$$\text{Kecepatan} = \text{rps} \times \text{keliling roda}$$

Maka hasil dari kecepatan jarak ini yang akan diperoleh berapa jarak yang telah dilalui player dalam bermain game menggunakan Immersive Tool. Dan hasil dari data kecepatan nantinya akan ditampilkan pada *Serial Monitor* di dalam mikrokontroller.

2.3 Komunikasi data serial yang menggunakan USB A/B Asinkron

Komunikasi serial adalah komunikasi yang mengantarkan data digital secara bit per bit secara bergantian melalui media *interface* serial. Dalam game sepeda yang menggunakan *Immersive Tool* ini menggunakan pengiriman data serial menggunakan USB A/B. Didalam komunikasi serial terdapat 2 macam cara komunikasi data serial yaitu **Sinkron** dan **Asinkron**. Pada komunikasi **data serial Sinkron**, data dikirim dalam bentuk berkelompok (blok) dalam kecepatan yang tetep tanpa bit awal dan bit akhir. Awalan blok (*start block*) dan akhiran blok (*stop block*) diidentifikasi dalam bentuk *bytes* dengan susunan yang spesifik.

Clock pada penerima dioperasikan secara terus menerus dan dikunci agar sama dengan *clock* yang diterima pengirim. Sedangkan pada komunikasi

data serial Asinkron tidak memiliki bit awalan dan akhiran, maka transmisi tak sinkron memiliki kedua bit tersebut. Pada transmisi ini, informasi akan diuraikan menjadi karakter dan masing-masing karakter tersebut memiliki bit yang diidentifikasi sebagai awalan blok (*start blok*) dan bit akhiran blok(*stop blok*).

Pada game sepeda yang menggunakan *Immesrsive Tools* ini memakai data serial *Asinkron*, Pengiriman data Asinkron ini lebih sederhana dibandingkan dengan pengiriman data *sinkron* karena hanya isyarat data saja yang dikirimkan. Clock penerima dibangkitkan secara local didalam penerima dan tetap dijaga agar sesuai dengan *clock* pengirim. Bit awal dan bit akhir yang dikirimkan tidak membawa informasi tetapi hanya menunjukkan awal dan akhir setiap karakter.

Transmisi *Asinkron* digunakan apabila pengiriman data dilakukan 1 karakter setiap kali pengiriman. Transmisi ini dilakukan dengan cara memberikan bit awal (*Start bit*) pada setiap awal pengiriman karakter dan diakhiri dengan bit akhir (*Stop bit*).

Definisi metode Asinkron sebagai berikut:

- Pengiriman data dilakukan 1 karakter setiap kali, sehingga penerima harus melakukan sinkronisasi agar bit data yang dikirim dapat di terima dengan benar.
- Transmisi kecepatan tinggi

- 1 karakter dengan yang lainnya tidak ada waktu antara yang tetap bila terjadi kesalahan maka 1 blok data akan hilang
- Membutuhkan *start pulse / start bit* (tanda mulai menerima bit data)
- *Idle transmitter* = '1' terus menerus, sebaliknya '0'.
- Tiap karakter diakhiri dengan *stop pulse / stop bit*.
- Dikenal sebagai *start-stop transmission*.

Pengiriman data Asinkron berupa Clock penerima dibangkitkan secara local didalam penerima dan tetap dijaga agar sesuai dengan clock pengirim. Bit awal dan bit akhir yang dikirim tidak membawa informasi tetapi hanya menunjukkan awal dan akhir setiap karakter.

2.4 Mikrokontroller Arduino Duemilanove

Arduino adalah sebuah platform prototyping open source berdasarkan mudah digunakan hardware dan software. papan arduino dapat membaca maskkan cahaya pada sensor, jari pada tombol, dan mengubahnya menjadi output, mengaktifkan motor, menyalakan LED. (Arduino.cc).

Duemilanove adalah mikrokontroller CMOS 8-bit berarsitektur AVR RISC yang memiliki 8K Bytes In-System *Programmable Flash*. Mikorokontroller dengan konsumsi daya rendah ini mampu mengeksekusi

instruksi dengan kecepatan maksimum 16 MIPS Pada frekuensi 16MHz
Berikut adalah fitur selengkapnya dari AVR Duemilanove.



Gambar 2.6 Arduino Duemilanove

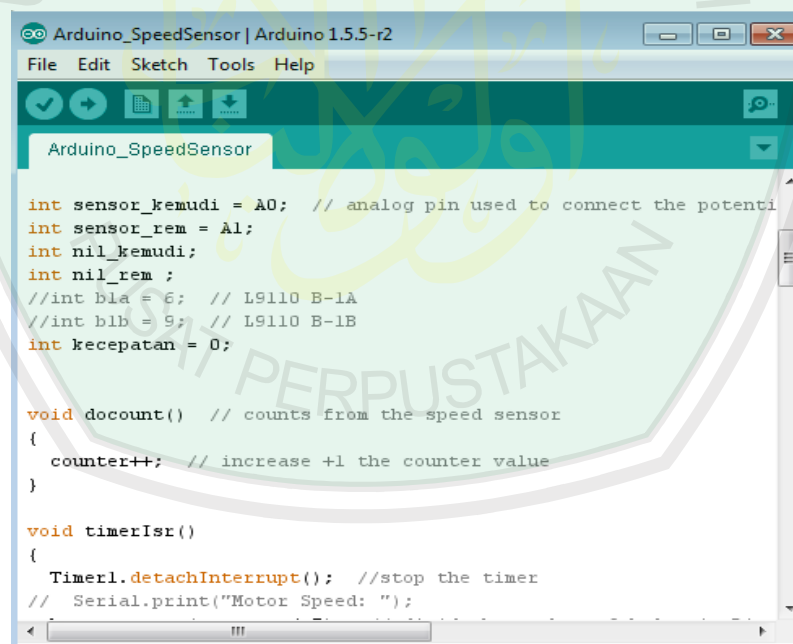
Kelebihan dari arduino Duemilanove:

- ❖ Tidak perlu perangkat chip programmer karena didalamnya sudah ada bootloader yang akan menangani upload program dari komputer.
- ❖ Sudah memiliki sarana komunikasi USB, sehingga pengguna Laptop yang tidak memiliki port serial/RS323 bisa menggunakannya. Sambungan dari komputer ke board arduino menggunakan USB, bukan serial atau parallel port. Sehingga akan mudah menghubungkan Arduino ke PC (Personal Computer) atau laptop yang tidak memiliki serial atau *parallel port*.

- ❖ Bahasa Pemrograman relative mudah karena software Arduino dilengkapi dengan kumpulan *library* yang cukup lengkap.
- ❖ Memiliki modul siap pakai (*shield*) yang bisa ditancapkan pada *board* Arduino. Misalnya *Shield GPS, Ethernet, SD Card*

2.4.1 Arduino Development Environment

Arduino *Integrated Development Environment* berisi editor teks untuk menulis kode, area pesan, konsol teks, toolbar dengan tombol untuk fungsi-fungsi umum dan serangkaian menu. Menghubungkan ke perangkat keras arduino dan genuine untuk mengupload program dan berkomunikasi (Arduinno.cc)



Gambar 2.7 IDE Arduino

Gambar diatas adalah lingkungan pengembangan *Arduino*, menjelaskan sebuah lingkungan pengembangan mencakup *toolbar*, serangkai menu, teks

editor dan baris perintah konsol. Kode ditulis dalam Arduino disebut *Sketsa*, dan memiliki ekstensi *file .ino*. *Arduino IDE* menggunakan alat baris perintah untuk mengkompilasi dan mengupload kode sumber ke mikrokontroler. Selama proses kompilasi kode sumber, IDE memeriksa ukuran kode sumber terhadap memori yang tersedia dalam mikrokontroler. Jika kode sumber lebih besar dari memori yang tersedia, akan muncul bendera pesan kesalahan di konsol.

2.5 ADC

ADC adalah singkatan dari *Analog to digital Converter*. Yang berfungsi untuk mengubah data analog menjadi digital, sehingga dapat diproses oleh mikrokontroler. ADC mengubah suatu tegangan analog ke nomer biner (rangkaiannya 1 dan 0), dan kemudian akhirnya ke nomor digital (basis 10) untuk membaca pada meteran, monitor, atau grafik. Nomor digit biner (bit) yang mewakili jumlah digital menentukan resolusi ADC. Namun, digital jumlah ini hanya perkiraan nilai sebenarnya dari tegangan analog pada suatu saat tertentu karena tegangan hanya dapat diwakili (digital) di diskritkan. Seberapa dekat jumlah digital mendekati nilai analog juga tergantung pada resolusi ADC. Mikroprosesor hanya dapat melakukan kompleks pengolahan sinyal digital.

Hal-hal yang juga perlu diperhatikan dalam penggunaan ADC adalah tegangan maksimum yang dapat dikonversikan oleh ADC dari

rangkaian pengkondisi sinyal, resolusi, pewaktu eksternal ADC, tipe keluaran, ketetapan dan waktu konversinya.

Beberapa Karakteristik penting ADC:

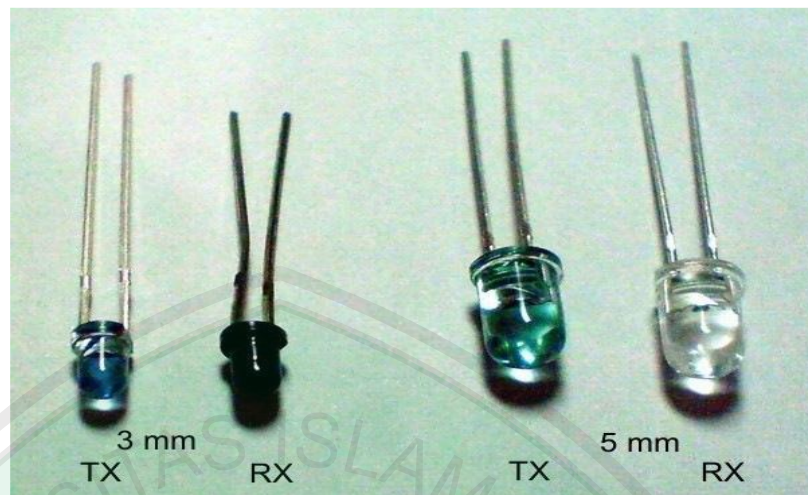
1. Waktu konversi
2. Resolusi
3. Ketidak linearan
4. Akurasi.

Ada banyak cara yang dapat digunakan untuk mengubah sinyal analog menjadi sinyal digital yang nilainya proporsional. Jenis ADC yang biasa digunakan dalam perancangan adalah jenis *successive approximation conversion* atau pendekatan bertingkat yang memiliki waktu konversi jauh lebih singkat dan tidak tergantung pada nilai masukan analognya atau sinyal yang akan diubah. (Mussuga, 2014).

2.6 Sensor

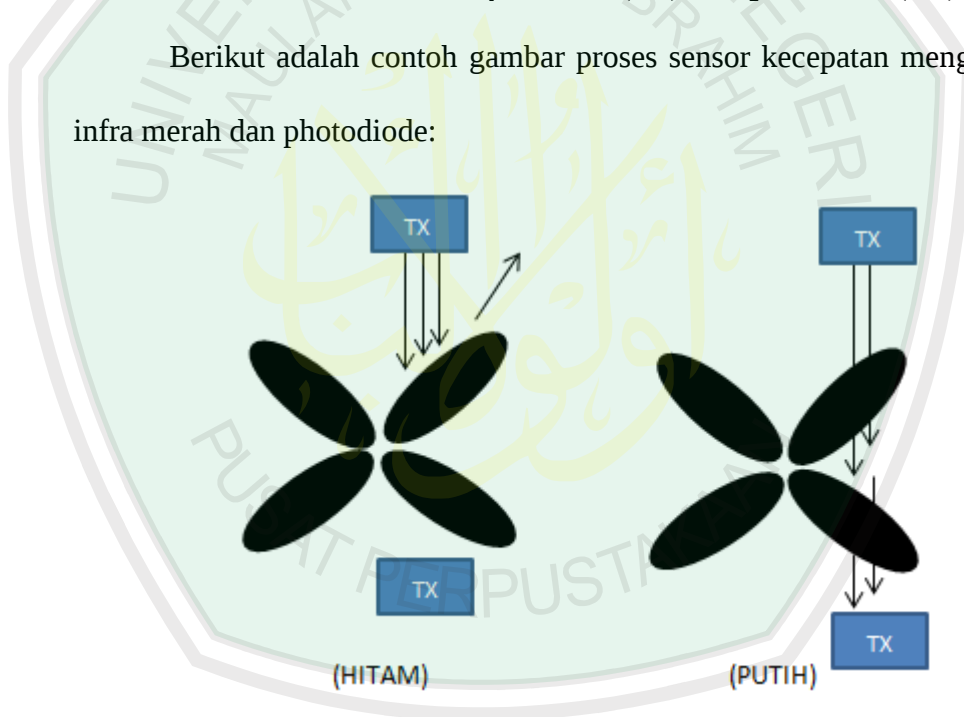
2.6.1 Sensor Kecepatan menggunakan LED Infra Merah dan Photodiode

Infra Red adalah komponen elektronik yang dapat memancarkan sinar inframerah dengan jarak yang pendek dan tidak terlihat oleh mata. Contoh cahaya infra merah yaitu cahaya api. Sedangkan photodiode adalah komponen elektronika yang dapat menangkap sinar infra merah dan mengubahnya kedalam sinyal listrik berupa arus listrik (biasanya penggunaannya sebagai saklar) (Pogal, 2014).



Gambar 2.8 Inframerah (TX) dan photodiode(RX)

Berikut adalah contoh gambar proses sensor kecepatan menggunakan infra merah dan photodiode:

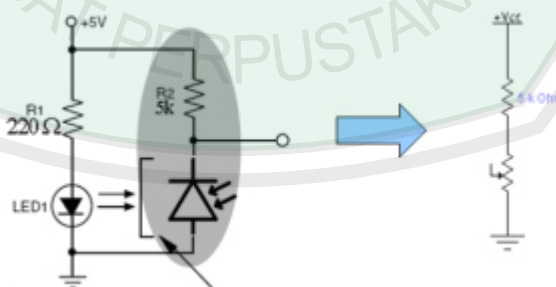


Gambar 2.9 Aplikasi sensor garis dengan inframerah dan photodiode

Dari gambar diatas bisa dilihat bahwa simulasi roda inframerah akan memancarkan sinar dan akan ditangkap oleh *photodiode*, dan penggunaan dalam sensor kecepatan maka nantinya ketika inframerah melewati warna

putih maka sensor *photo dioda* akan menangkap dengan nilai 1 dan begitu juga sebaliknya ketika sensor infra merah melewati garis warna hitam maka sensor *photo dioda* tidak akan bisa mendeteksi (0). *Photo dioda* akan aktif apabila tidak terkena cahaya dari led infra merah. Antara led dan *photo dioda* dipisahkan oleh jarak. Jauh dekatnya jarak memengaruhi besar intensitas cahaya yang diterima oleh *photo dioda*. Pada bagian sensor ini terdapat sebuah LED infra merah (IR LED) yang berfungsi untuk mengirimkan sinyal kepada *receiver*.

Infrared memancarkan cahaya ke bidang berwarna putih, cahaya akan dipantulkan hampir semuanya oleh bidang berwarna putih tersebut, sebaliknya ketika transmitter memancarkan cahaya ke bidang berwarna gelap atau hitam, maka cahaya akan banyak diserap oleh bidang gelap tersebut, Perbedaan cahaya yang diterima oleh receiver akan menyebabkan hambatan yang berbeda-beda di dalam receiver (*photodiode*) tersebut.



Gambar 2.10 Ilustrasi rangkaian sensor

Gambar di atas merupakan ilustrasi rangkaian pada sensor photodiode, Receiver bisadi analogikan dengan resistor variable, yaitu resistor yang nilai hambatannya bisa berubah. Sehingga nilai tegangan di output rangkaian berubah, baca putih akan mengeluarkan output dengan tegangan rendah (0 volt), dan baca hitam akan mengeluarkan output dengan tegangan tinggi (Mendekati $V_{cc} = 5$ Volt).

LED infra merah. Jika dibandingkan dengan menggunakan LED biasa, LED infra merah memiliki ketahanan yang lebih baik terhadap sinyal tampak. Cahaya yang dipancarkan oleh LED infra merah tidak terlihat oleh mata telanjang.

2.6.2 Sensor Kemudi



Gambar 2.11 Sensor *potensiometer*

Sensor yang digunakan ialah *potensiometer*. sensor potensiometer merupakan sensor analog yang paling sederhana namun sangat berguna untuk mendeteksi posisi putaran (Muslimin, 2015), Selain itu juga *Potensiometer* adalah *resistor* tiga terminal dengan sambungan geser yang

membentuk pembagi tegangan dapat disetel. (Pogal, 2014), Jika hanya dua terminal yang digunakan (salah satu terminal tetap dan terminal geser), *potensiometer* berperan sebagai *resistor variabel* atau *Rheostat*. *Potensiometer* biasanya digunakan untuk mengendalikan peranti elektronik seperti pengendali suara pada penguat. *Potensiometer* yang dioperasikan oleh suatu mekanisme dapat digunakan sebagai *transduser*, misalnya sebagai sensor *joystick*.



BAB III

ANALISIS DAN PERANCANGAN

3.1 Kebutuhan Sistem

Kebutuhan *Immersive Tools* yang diperlukan sebagai tahapan dasar analisis system yang akan dibangun, yaitu meliputi kebutuhan platform, deskripsi system, perancangan proses.

3.1.1 Kebutuhan Platform

Analisis kebutuhan merupakan analisis terhadap komponen-komponen yang digunakan untuk membuat system. Analisis kebutuhan ini terbagi menjadi dua macam yaitu komponen perangkat lunak dan perangkat keras sebagai berikut.

1. Perangkat Lunak

Perangkat Lunak adalah istilah umum untuk data yang diformat dan disimpan secara digital, termasuk program computer, di dokumentasinya, dan berbagai informasi yang bisa dibaca dan ditulis oleh computer. Perangkat lunak yang digunakan dalam penelitian ini adalah:

- a. *CadSoft Eagle.*
- b. *Arduino Development Environment*
- c. *Google Sketchup v8.0*
- d. *Unity 4.6.1*
- e. *Windows 7 intel Pentium edition x32.*

2. Perangkat Keras

Perangkat keras yang dibutuhkan dalam penelitian ini adalah:

- a. Mikrokontroler *Arduiono Duemilanove Uno*
- b. Sensor LED
- c. Sensor *Potensiometer* 100 Ohm
- d. Kabel USB
- e. *Rotary DC Fan* 12 Volt
- f. PC/ Laptop

Untuk *Spesifikasi hardware* dan *software* pada PC ataupun laptop, sudah dicoba secara langsung untuk menjalankan game simulasi, hal yang berpengaruh dan menjadi prioritas adalah pada spesifikasi hardware, yang harus setara atau lebih pada spesifikasi diatas, karena hal ini berpengaruh pada pengolahan grafis yang menggunakan animasi 3D.

3.1.2 Deskripsi Sistem

- a. Komunikasi data serial menggunakan USB A/B Asinron

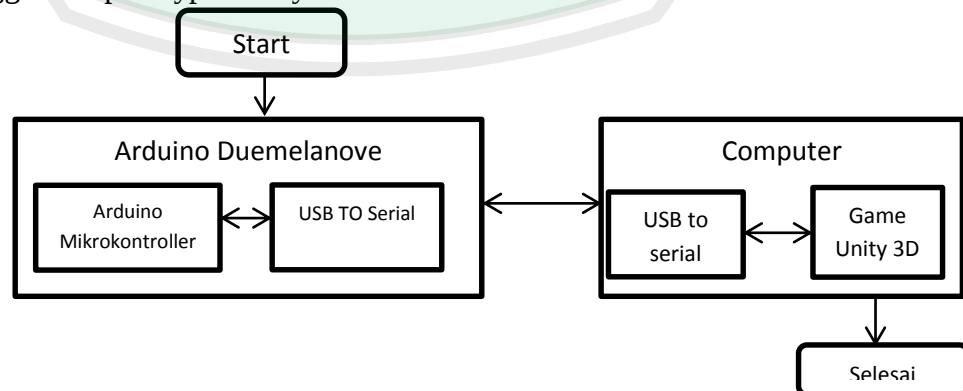
Komunikasi serial adalah komunikasi yang pengiriman datanya per bit secara berurutan dan bergantian. Komunikasi ini mempunyai suatu kelebihan yaitu hanya membutuhkan satu jalur dan kabel yang sedikit dibandingkan dengan komunikasi parallel.

Pada prinsipnya komunikasi serial merupakan komunikasi dimana pengiriman data dilakukan per bit sehingga lebih lambat dibandingkan

komunikasi parallel, atau dengan kata lain komunikasi serial merupakan salah satu metode komunikasi data dimana hanya satu bit data yang dikirimkan melalui seuntai kabel pada suatu waktu tertentu. Pada dasarnya komunikasi serial adalah kasus khusus komunikasi parallel dengan nilai $n=1$, atau dengan kata lain adalah suatu bentuk komunikasi parallel dengan jumlah kabel hanya satu dan hanya mengirimkan satu bit data secara simultan.

Pada komunikasi data serial dikirimkan dalam bentuk kode biner, untuk nilai logika 1 adalah tegangan -3 s/d -25 volt dan untuk logika '0' sebagai tegangan +3 s/d +25 volt, dengan demikian tegangan dalam komunikasi serial memiliki ayunan tegangan maksimum 50 volt. Dengan besarnya range tegangan ini maka dalam komunikasi secara seri memungkinkan untuk menggunakan kabel yang lebih panjang karena gangguan-gangguan mudah diatasi. Sementara untuk tegangan -2 s/d +3 nilai logikanya terdefinisi.

Penggunaan serial ini akan menggunakan USB yaitu Universal serial Bus, transfer data berlangsung jauh lebih cepat dibandingkan dengan penggunaan port type lainnya



Gambar 3.1 Komunikasi Arduino Dengan Game Menggunakan USB

Dalam pengiriman data menggunakan pengiriman data serial dengan USB (*Universal Serial Bus*), Kabel USB adalah sebuah kabel serabut yang digunakan untuk menyambung perangkat ke dalam CPU atau computer. Kabel USB memiliki dua ujung yang ujung satu berupa soket USB yang nantinya akan dicolokkan ke *port* USB pada computer dan yang di ujung lain memiliki adapter yang akan dicolokkan pada mikrokontroller. Fungsi kabel USB yaitu sesuai namanya, gunanya untuk mentransmisikan data dari mikrokontroller ke dalam CPU dan kemudian data informasi tersebut akan diproses lebih lanjut oleh computer.

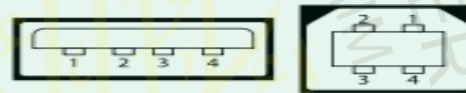
Konektor pada USB ada dua konektor Seri A dan B. konektor seri A dan seri B seperti pada gambar di bawah ini.



Gambar 3.2 Kabel USB A/B

Gambar di atas merupakan model kabel USB, Desain USB ditujukan untuk menghilangkan perlunya penambahan *expansion Card* ke ISA computer atau bus PCI, dan memperbaiki kemampuan plug and play (pasang dan mainkan) dengan memperbolehkan peralatan-peralatan ditukar atau ditambah ke system tanpa perlu me-reboot computer. Ketika USB dipasang, langsung dikenal system computer dan memproses driver yang diperlukan untuk menjalankannya.

Nomor kaki pada soket kabel USB A/B



Penetapan kaki

Kaki	Fungsi
1	V _{BUS} (4.75–5.25 V)
2	D ⁻
3	D ⁺
4	GND
Shell	Shield

Gambar 3.3 Kaki Soket USB

Sedangkan untuk koneksi pas Immersive Tools ini memakai USB A/B, hal ini bertujuan untuk mempermudah. Berikut ini spesifikasi dari koneksi USB A/B.

- ♦ Modus transfer: Asinkron
- ♦ Jumlah system yang ditemukan 2-5
- ♦ Jumlah perangkat maksimal: 127

- ◆ IRQ:11
- ◆ Panjang maksimal: 3-5 meter
- ◆ Maksimal data: 12 bit/sec(1.5 MB/sec)
- ◆ Power: 2.5 w

Transmisi Asinkron memiliki kecepatan data yang bervariasi, oleh karena itu membutuhkan protokol/parameter-parameter yang harus dimengerti oleh peralatan yang ingin berkomunikasi. Sedangkan pada *Arduino Duemelanove* sudah terdapat protokol yang tersedia, jadi hasil data dari mikrokontroler yang berupa String akan diubah menjadi bit.

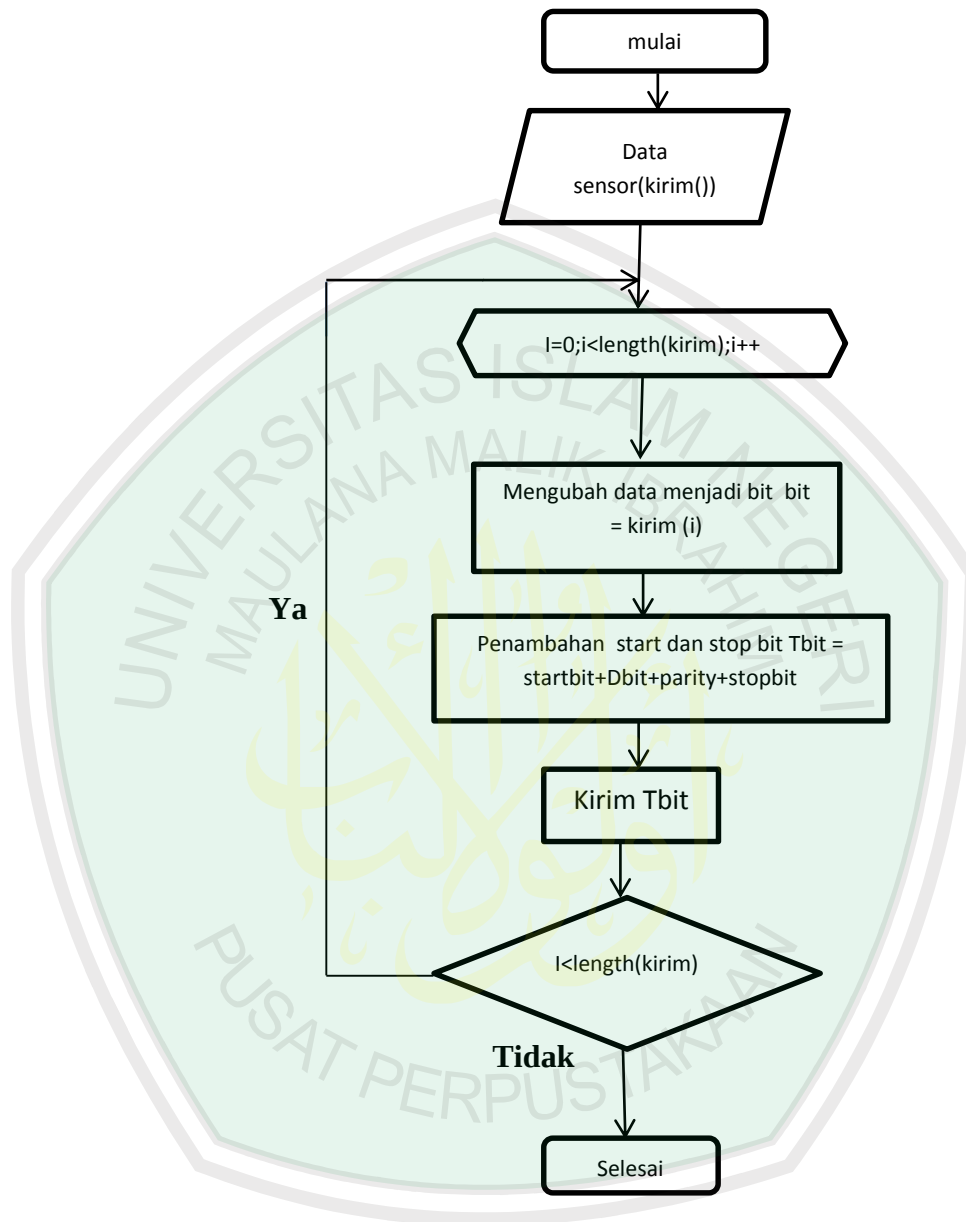
Protokol komunikasi data antara lain :

- Start Bit Selalu bernilai 0 : Ketika komunikasi UART (serial asinkron) akan diberikan, terlebih dahulu dimulai dengan pemberian Start Bit. Fungsinya sebagai pemicu (tanda) kepada penerima (Rx) bahwa akan ada data yang diberikan oleh pemancar (Tx) dan juga akan memicu *clock* pada *receiver* sehingga disinkronkan dengan *clock* pada *transmitter*. *Clock* penerima dan pemancar haruslah akurat dengan toleransi 10% sehingga tidak terjadi kesalahan data.
- Data Bits : adalah data yang akan dikirimkan secara UART dimulai dari LSB (bit ke 0) hingga MSB (bit terakhir). Jangan lupa menentukan banyaknya bit tersebut haruslah sama antara pemancar dengan penerima. Banyaknya data bits pada AVR bisa bernilai 7,8 atau 9 data bit.

- *Parity* (keseimbangan) : berfungsi sebagai pengecekan *error* data yang ditransfer. *Parity* bisa bernilai ODD (ganjil), *EVEN* Genap), dan *NONE* selain itu pemancar dan penerima harus menggunakan *parity* yang sama. Jika ODD *parity* maka jumlah total nilai 1 pada data *bit* + *parity* berjumlah ganjil, contoh ODD, jika data bits 00110101 maka *parity* bernilai 1. Sedangkan jika *EVEN parity* maka jumlah total nilai 1 pada data bit + *parity* berjumlah genap. Contoh *even*, jika data bits 00110101 maka *parity* bernilai 0.
- Stop Bit Selalu bernilai 1 : berfungsi sebagai akhir dari komunikasi data dan kemudian masuk pada *IDLE state*. Pengiriman data selanjutnya dapat dilakukan setelah *stop* bit diberikan.
- *IDLE state* : adalah kondisi tidak terjadinya komunikasi data dan jalur data berlogika 1 secara terus menerus (*marking*).

Sedangkan format pengiriman data untuk arduino memakai **9600-8-N-1**, dimana 9600 merupakan *baud rate*, 8 merupakan jumlah data, N *pariti* dan 1 merupakan *stop bit*. Jadi data dari mikrokontroler yang berupa *string* akan dikonversikan berupa data bit, dan data bit ini yang akan dikirim melalui transmisi asinkron. Dalam pengiriman data terdapat 1 karakter yang merupakan 1 *bit* dan menjadi 8 *bit*. Dalam pengiriman data pasti ada pengirim (TX) dan penerima (RX), yang digambarkan pada flowchart dibawah.

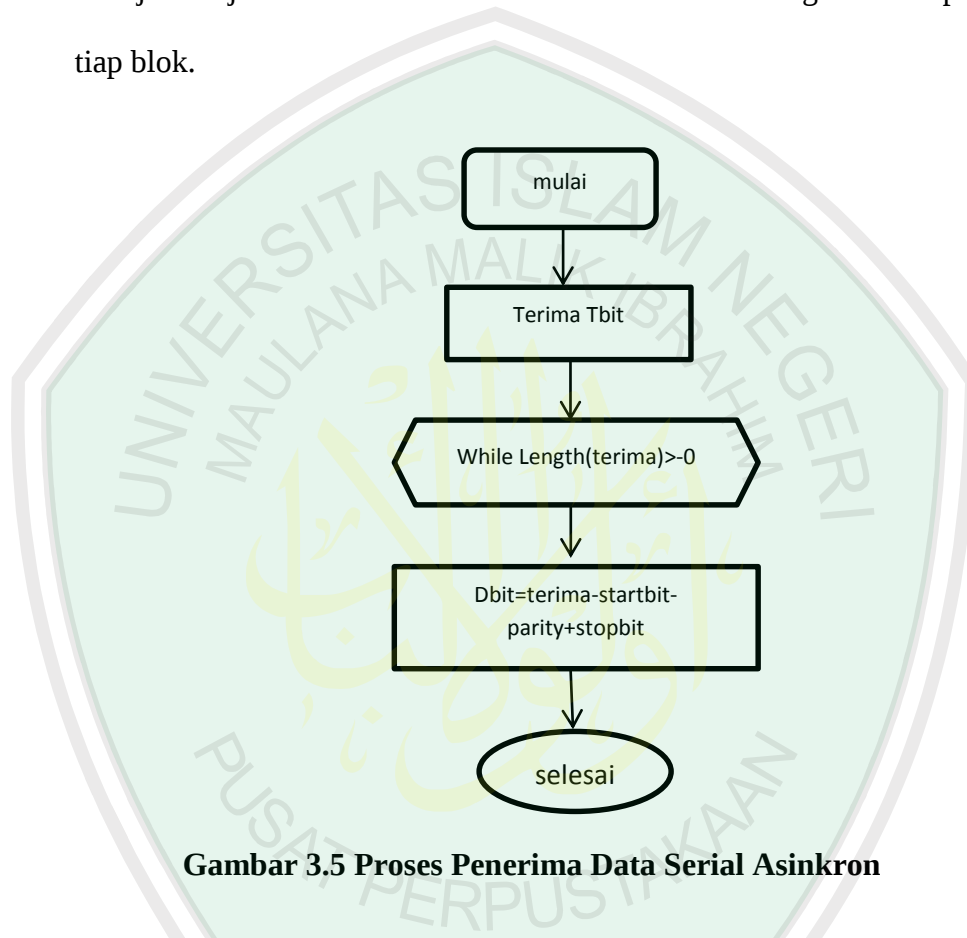
Proses pengiriman data serial asinkron



Gambar 3.4 Proses Pengiriman Data Serial Asinkron

Dari flowchart di atas dapat diterangkan bahwa data dari string dari setiap hasil sensor dikirim ke PC melalui USB, dengan mengubah data dari hasil mikrokontroler menjadi bit dengan protokol dan data tersebut dikirim dari jumlah data pertama sampai akhir dalam satu blok

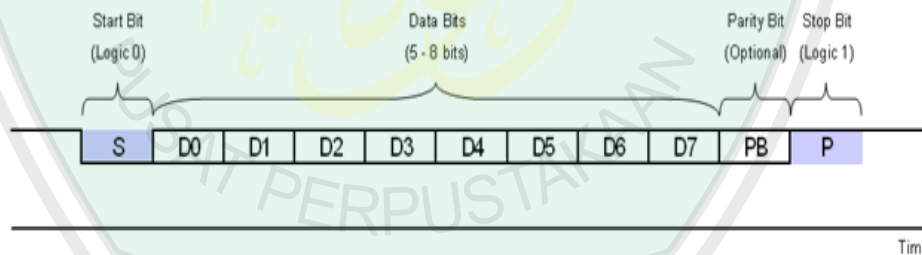
($i=0; \text{length}(\text{ kirim}); i++$). Jika data tidak dikirim maka cancel, data dalam satu blok sudah menjadi bit maka format pengiriman *asinkron* dengan menambahi *start bit*(0)+*data bit*+*pariti*+*stop bit*(1), dan kemudian data *bit* akan dikirim. Dan jika terjadi kesalahan maka data bit akan dikirim lagi untuk diproses lagi tiap blok.



Gambar 3.5 Proses Penerima Data Serial Asinkron

Setelah data bit diterima maka diproses dengan perulangan, untuk menerima dari kiriman data dari yang pertama ($\text{length}(\text{terima}) > 0$), ketika data sudah diterima dalam satu blok maka data yang dikirim akan dikurangi *start bit*, *parity* dan *stop bit*. Maka akan muncul hasil dari data bit dari awal data bit kirim dari mikrokontroler.

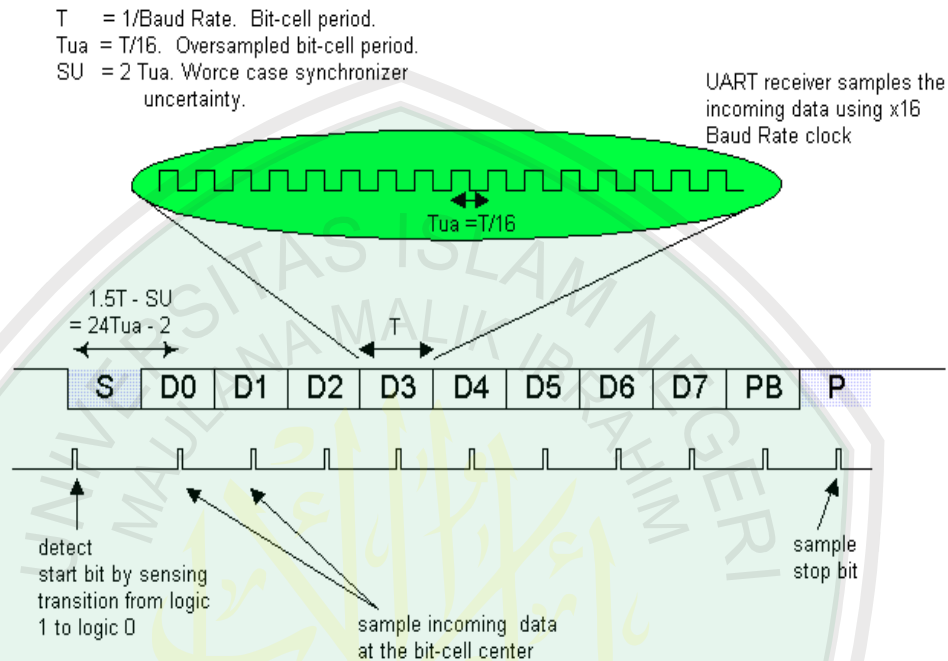
The Universal Asynchronous Receiver / Transmitter (UART) controller adalah komponen kunci dari komunikasi serial subsistem dari sebuah komputer dan pada mikrokontroler *Arduino Duemilanove* memakai UART. UART juga fitur terintegrasi umum di kebanyakan mikrokontroler. Transmisi Asinkron memungkinkan data yang akan dikirimkan tanpa pengirim harus mengirim sinyal *clock* ke penerima. Dalam hal ini, pengirim dan penerima harus setuju pada parameter waktu (*Band Rate*) transmisi sebelum dan bit khusus ditambahkan untuk setiap kata untuk menyinkronkan unit pengirim dan penerima. Dalam transmisi *asinkron*, pengirim mengirimkan *bit start*, 5 sampai 8 bit data (LSB pertama), bit paritas opsional, dan kemudian 1, 1,5 atau 2 berhenti bit (kong,2010).



Gambar 3.6 Paket UART: 1 mulai bit, 8 bit data,1 bit paritas dan 1 bit berhenti

Setiap operasi perangkat keras UART dikendalikan oleh sinyal *clock* yang berjalan pada tingkat yang jauh lebih lebih cepat dari pada tingkat band rate. Untuk mencari frekuensi sampling dari pengiriman serial menggunakan

UART ini dengan rumus baud rate dikalikan 16, dimana baud rate $9600 \times 16 = 153600$. Jadi frekuensi dari pengiriman data serial ini 153600.



Gambar 3.7 Data Titik Pengambilan Sampel Oleh Penerima UART

Transmisi data menerima *UART* harus ditetapkan pada tingkat yang sama baud, panjang karakter, *paritas*, dan stop bit untuk operasi yang tepat. Format khas untuk port serial digunakan dengan PC yang terhubung ke mikrokontroler adalah 1 Mulai bit, 8 bit data, tidak ada paritas dan 1 berhenti bit. *UART* adalah bentuk sederhana dari komunikasi antara mikrokontroler dan PC. Namun karena pertumbuhan yang menjamur teknologi, port serial secara perlahan digantikan dengan cara lain port komunikasi. Namun demikian, komunikasi serial masih mungkin bahkan tanpa port serial fisik pada PC anda yang menggunakan *USB (Universal serial bus)* (Teguh.2003).

Definisi metode *Asinkron* sebagai berikut:

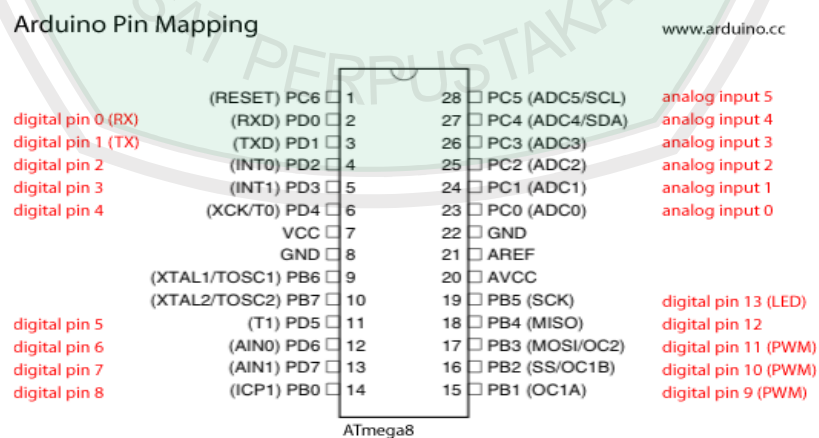
- Jumlah bit tiap karakter adalah 5 sampai 8 bit.
- *Parity* bit yang digunakan untuk mendeteksi kesalahan (*error*) yang berbentuk *odd* (ganjil), *even* (genap) atau tanpa *parity* (*no parity*).
- Jumlah *stop bit* (1 bit, 1,5 bit, atau 2 bit) dan 1 *start bit*.

Baud rate atau kecepatan data (bps).

b. Mikrokontroler Arduino Duemillanove

Duemilanove adalah mikrokontroler CMOS 8-bit berarsitektur AVR RISC yang memiliki 8K Bytes *In-System Programmable Flash*. Mikrokontroler dengan konsumsi daya rendah ini mampu mengeksekusi instruksi dengan kecepatan maksimum 16 MIPS pada frekuensi 16MHz.

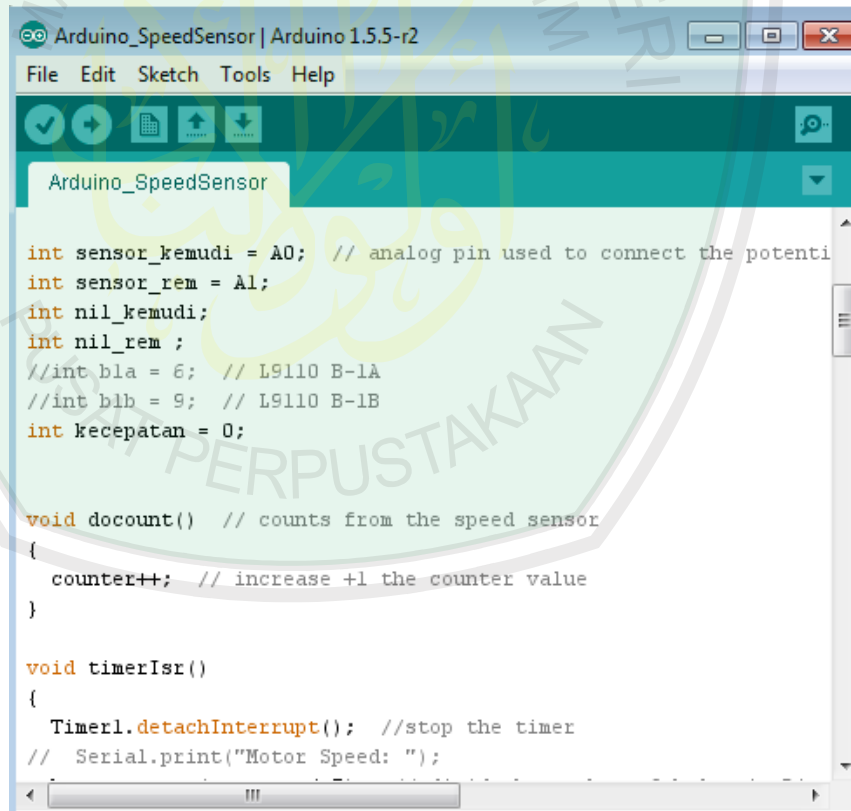
Konfigurasi kaki-kaki mikrokontroler *Duemillanove*



Gambar 3.8 Konfigurasi Kaki-Kaki Arduino

Mikrokontroler Duemilanove mempunyai dua timer yaitu TIMER 0 dan TIMER 1. Kedua timer saling berbagi dua macam SFR (Special Function Register) yang mengontrol timer masing – masing timer memiliki dua macam SFR yang spesifik, yaitu TH0/TL0, untuk timer 0 dan TH1/TL1 untuk timer 1

Bahasa pemrograman Arduino adalah bahasa C. Tetapi bahasa ini sudah dipermudah menggunakan fungsi-fungsi yang sederhana sehingga pemula pun bisa mempelajarinya dengan cukup mudah. Untuk membuat program Arduino dan mengupload ke dalam board Arduino, anda membutuhkan software arduino IDE (*Integrated Development Environment*).



Gambar 3.9 Integrated Development Environment

Berikut ini adalah tombol-tombol toolbar serta fungsinya:



Verify untuk mengecek *error* pada *code* program.



Upload untuk meng *compile* dan meng *upload* program ke *Arduino board*



New untuk membuat *sketch* baru.



Open untuk menampilkan sebuah menu dari seluruh *sketch* yang berada di dalam.



Serial Monitor untuk membuka serial monitor.

c. ADC

ADC adalah singkatan dari *Analog to digital Converter*. Fungsinya merubah data analog menjadi digital sehingga dapat diproses oleh *mikrokontroler*.

Hal-hal yang juga perlu diperhatikan dalam penggunaan ADC Tegangan maksimum yang dapat dikonversikan oleh ADC dari rangkaian pengkondisi sinyal, resolusi, pewaktu eksternal ADC, tipe keluaran, ketetapan, dan waktu konversinya.

Beberapa karakteristik penting ADC :

1. Waktu konversi
2. Resolusi

3. Ketidak Linearan

4. Akurasi

Ada banyak cara yang dapat digunakan untuk mengubah sinyal analog menjadi sinyal digital yang nilainya proposional. Jenis ADC yang biasa digunakan dalam perancangan adalah jenis Successive approximation conversion atau pendekatan bertingkat yang memiliki waktu konversi jauh lebih singkat dan tidak tergantung pada nilai masukan analognya atau sinyal yang akan diubah.

d. Prosedur pengisian program mikrokontroller menggunakan arduino development environment.

Adapun prosedur atau tata cara pengisian program ke mikrokontroller Duemillanove adalah sebagai berikut :

- Program dibuat dalam bahasa C yang ditulis di Code Vision AVR, namun bisa juga diketik dengan menggunakan editor yang mendukung dalam pengisian program pada mikrokontroller Arduino Duemilanove.
- Pada saat penyimpanan program yang dibuat sebaiknya disimpan pada satu folder, tujuannya agar pada saat membutuhkan program yang telah dibuat bisa di download.
- Program yang telah disimpan harus dikompilasi (compiler), tujuannya agar program yang kita simpan menjadi bahasa mesin

sehingga dimengerti oleh mikrokontroler, dengan menekan tombol F9

- Sebelum melakukan kompilasi terlebih dahulu melakukan pengecekan terhadap error pada program yang dibuat.
- Untuk mulai mengisi atau mendownload program mikrokontroller langsung dengan code Vision AVR, dengan cara menekan Ctrl+F9 maka tampil dilayar proses pengisian programming pada mikrokontroller tunggu sampai pengisian 100 %.

Bahasa Pemrograman Arduino adalah bahasa C. tetapi bahasa ini sudah di permudah menggunakan fungsi-fungsi yang sederhana sehingga pemula pun bisa mempelajarinya dengan cukup mudah.

e. Sensor Kecepatan menggunakan LED Infra merah dan photodiode.

LED Infra Merah merupakan salah satu jenis LED (*Light Emitting Diode*) yang dapat memancarkan cahaya infra merah yang tidak kasat mata. Cahaya infra merah merupakan gelombang cahaya yang berupa pada *spektrum* cahaya tak kasat mata. LED infra merah dapat memancarkan cahaya infra merah pada saat dioda LED ini diberikan tegangan bias maju pada anoda dan katodanya. LED infra merah ini dapat memancarkan gelombang cahaya infra merah karena dibuat dengan bahan khusus untuk memancarkan cahaya infra merah. Secara teoritis LED infra merah mempunyai panjang gelombang 7800 Å dan mempunyai daerah frekuensi 3.104 sampai 4.104 Hz. Dilihat dari jangkah frekuensi yang begitu lebar, infra merah sangat fleksibel dalam penggunaannya. Sensor kemudi dan pengereman.



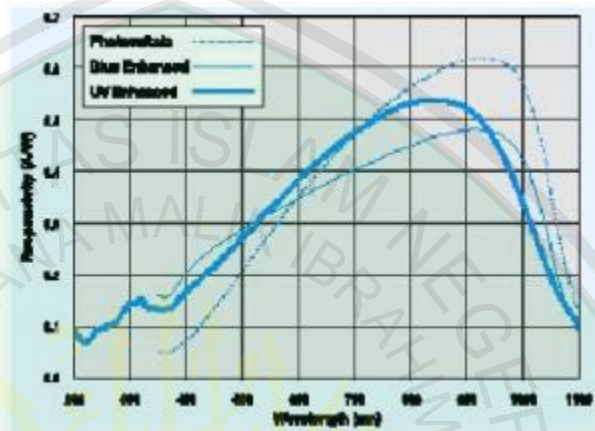
Gambar 3.10 Bentuk LED Infra Merah

Cahaya infra merah tidak mudah terkontaminasi dengan cahaya lain, sehingga dapat digunakan baik siang maupun malam. Aplikasi cahaya infra merah sendiri dapat digunakan sebagai *link* pada jaringan telekomunikasi atau dapat juga dipancarkan pada *fiber optic*. Sebagai receiver cahaya infra merah dapat digunakan photo dioda, maupun modul *receiver* infra merah.

LED Photo dioda merupakan dioda yang peka terhadap cahaya, sensor photo dioda akan mengalami perubahan resistansi pada saat menerima intensitas cahaya dan akan mengalirkan arus listrik secara forward sebagaimana dioda pada umumnya. Sensor *photo diode* adalah salah satu jenis sensor peka cahaya (*photo detector*). Jenis sensor peka cahaya lain yang sering digunakan adalah *phototransistor*.

Tanggapan *frekuensi* sensor *photo dioda* tidak luas. Dari rentang tanggapan itu, sensor *photo dioda* memiliki tanggapan paling baik terhadap 40

cahaya infra merah, tepatnya pada cahaya dengan panjang gelombang sekitar $0,9 \mu\text{m}$. Kurva tanggapan sensor *photo dioda* ditunjukkan pada gambar berikut:



Gambar 3.11 Kurva Tanggapan Frekuensi Sensor Photodioda

Hubungan antara keluaran sensor photodioda dengan intensitas cahaya yang diterimanya ketika dipanjar mundur adalah membentuk suatu fungsi yang *linier*. Prinsip kerja photodioda ketika sebuah *photon* (satu satuan energi dalam cahaya) dari sumber cahaya diserap, hal tersebut membangkitkan suatu *elektron* dan menghasilkan sepasang pembawa muatan tunggal, sebuah *elektron* dan sebuah *hole*, di mana suatu *hole* adalah bagian dari kisi-kisi *semi konduktor* yang kehilangan *elektron*.

Cara kerja Infra merah dan PhotoDioda

- Prinsip kerja dari rangkaian ini adalah photodioda akan ditembak secara terus menerus oleh cahaya infra merah, apabila nanti akan

mendeteksi warna hitam dan putih. Untuk nilai hitam akan muncul 0 sedangkan warna putih akan mempunyai nilai 1.

- Proses penghitungannya dilakukan dengan mendeteksi adanya perpotongan pada jalur infra merah.
- Setiap perpotongan akan memberikan perubahan kondisi logika dari 0 ke 1 selama selang waktu tertentu.
- Perubahan kondisi logika ini yang digunakan sebagai acuan perhitungan.
- Sensor ini diletakkan berhadapan antara sensor infra red dan phototdioda pada jalur yang akan dilewati garis hitam maupun garis putih.



**Gambar 3.12 Peletakkan Sensor Saat Mendeteksi Warna Hitam
Dan Putih Pada Roda Simulasi**

f. Sensor Kemudi dan Pengereman

Sensor kemudi dan pengereman ini menggunakan Potensiometer. Potensiometer adalah resistor tiga terminal dengan sambungan geser yang membentuk pembagi tegangan dapat disetel. Jika hanya dua terminal yang digunakan (salah satu terminal tetap dan terminal geser), Potensiometer berperan sebagai *resistor variabel* atau *Rheostat*. Potensiometer biasanya digunakan untuk mengendalikan peranti elektronik seperti pengendali suara pada penguat.

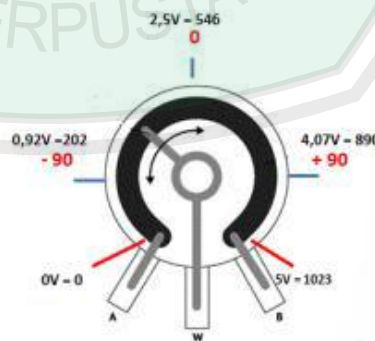


Gambar 3.13 Potensiometer Pada Simulasi Sepeda Untuk Sensor Kemudi Dan Pengereman

Dari Segi mekanik, potensiometer dapat diletakkan pada posisi yang kita inginkan karena dilihat dari bentuknya yang *simple* dan juga mendukung *mekanik*. Selain itu juga ada banyak pilihan bentuk potensiometer yang tersedia di pasaran. Potensiometer yang tersedia di pasaran terdiri dari beberapa jenis, yaitu: potensiometer karbon, *potensiometer wire wound* dan *potensiometer metal film*.

Dari Segi elektrik, penggunaan potensiometer sebagai sensor posisi cukup praktis karena hanya membutuhkan satu tegangan eksitasi dan biasanya tidak membutuhkan pengolahan sinyal yang rumit. Dari segi *programming*, perubahan posisi dapat diukur dari perubahan resistansi yang dimiliki potensiometer yang sebelumnya telah dikonversi menjadi sinyal inputan yang sesuai dengan kontroler baik tegangan maupun arus.

Berikut ini adalah gambaran dari sensor potensiometer yang digunakan untuk sensor kemudi.



Gambar 3.14 Potensiometer Dengan Belokan Kanan Dan Kiri

Pada gambar ditunjukkan bahwa pada posisi 0° (2,5V-546) maka posisi kemudi tepat lurus, sedangkan pada posisi -90° (0,92V-202) maka kemudi belok kiri dan 90° (4,7V-890) akan belok kanan. Maka dari itu untuk belok kiri nanti dari posisi 0 tegangan akan semakin rendah, begitu juga sebaliknya jika belok kanan maka tegangan akan tambah besar dari posisi 0 dan daya yang digunakan adalah 5V dan 10bit(0-1023).

Setiap hasil data dari sensor nantinya akan diolah pada mikrokontroler dan data akan dihitung dan dijadikan sudut belokan dari paling kiri (-90) dan paling kanan (90) sedangkan nilai tengah (0). Yang nantinya data hasil yang akan dikirim ke PC dan akan dimasukkan pada *game*.

Dalam sensor pengereman jika kita mengerem nantinya pasti akan mempengaruhi tentang kecepatan, pengereman dilakukan ketika *player* ingin berbelok maupun terlalu kencang dalam mengayuh. Selain pengereman dengan sensor potensio sendiri dalam sepeda ini juga masih terdapat rem yang masih berfungsi.

g. Statistik Data

Statistik data digunakan untuk mencari nilai kecepatan rata, kecepatan maksimal pada *game* yang menggunakan *Immersive Tool*. Data kecepatan diperoleh dari berapa kali sensor membaca titik putih (1) dan titik hitam (0) pada roda simulasi dalam hal ini menggunakan *Rotary DC Fan* dan data akan dikirimkan ke mikrokontroler untuk diproses. Dalam mikrokontroler

nantinya diberikan rumus menghitung kecepatan, kecepatan rata-rata, kecepatan maksimal.



Gambar 3.15 Tanda Sensor Pada Roda

Dari gambar tanda roda di atas dapat diketahui bahwa untuk keliling roda 180 cm, sedangkan tanda putih berjumlah 7 dan tanda hitam berjumlah 7.

Pemberian tanda ini akan mendeteksi adanya titik hitam dan putih yang akan di baca oleh photodiode sehingga bisa di program pada arduino

Hasil perhitungan data dapat ditampilkan pada *serial monitor* sedangkan untuk mencari data kecepatan dapat menggunakan rumus.

$$Speed = \frac{count\ A}{jumlah\ tanda} \times 1000 \times waktu\ sampling$$

Di mana :

Count A : jumlah baca tanda tiap detik

Jumlah tanda: jumlah data pada roda

1000 : waktu sampling menggunakan detik

Misal $\frac{7}{7} \times 1000 \times 1000 = 1 \text{ rps}$

Hasil data dari perhitungan di atas akan berupa dalam satuan detik. Jadi berapa kali sensor membaca jumlah warna putih tiap detik, maka hasil ini yang akan dihitung. Sedangkan untuk menghitung kecepatan rata-rata dengan rumus perhitungan.

$$AVG = \frac{Kecepatan\ 1 + \dots + kecepatan\ N}{N}$$

Dimana :

AVG (*Average*) kecepatan rata

Kecepatan 1- kecepatan N ialah hasil kecepatan perdetik

N merupakan jumlah dari data

Misal: Misal: $AVG = 1 + \frac{1+2+5+4+3}{5} = \frac{15}{5} = 3 \text{ rps}$

Sedangkan untuk menghitung nilai rata-rata per detik akan menjumlahkan semua data hasil per detik, berikut ini permisalan untuk gambaran mencari kecepatan rata-rata.

$$\begin{aligned}
 &= \frac{1}{1} = 1 \\
 &= \frac{1+2}{2} = 1,5 \\
 &= \frac{1+2+5}{3} = 2,6 \\
 &= \frac{1+2+5+4}{4} = 3 \\
 &= \frac{1+2+5+4+3}{5} = 3
 \end{aligned}$$

} nilai rata-rata

3.1.3 Kalibrasi Sensor

Kalibrasi adalah kegiatan untuk menentukan kebenaran konvensional nilai penunjukkan alat ukur dan bahan ukur dengan cara membandingkan terhadap standar ukur yang mampu telusur (*traceable*) ke standar nasional maupun internasional untuk satuan ukuran dan bahan-bahan acuan tersertifikasi.

Dalam setiap sensor yang dipakai pasti ada penyusutan nilai keakuratan deteksi. Banyak cara untuk kalibrasi, salah satunya juga bisa dengan *Arduino Development Environment*. Nantinya kita bisa melihat nilai yang terdapat pada program, misal pada sensor potensiometer sangatlah sulit

untuk mencari nilai 0, maka dari itu dicari nilai yang paling mendekati nilai 0. Selain memakai *Arduino Development Environment* juga bisa memakai *Voltmeter*, cara ini sangat manual sehingga dicari nilai tengah dari seluruh nilainya.

Berikut ini adalah garis besar dari kalibrasi:

➤ **Tujuan Kalibrasi**

- Mencapai ketertelusuran pengukuran. Hasil pengukuran dapat dikaitkan/ditelusur sampai ke *standar* yang lebih tinggi/teliti (*standar primer nasional* dan / *internasional*), melalui rangkaian perbandingan yang tak terputus.
- Menentukan *deviasi* (penyimpangan) kebenaran nilai *konvensional* penunjukan suatu *instrument* ukur.
- Menjamin hasil-hasil pengukuran sesuai dengan standar nasional maupun internasional.

➤ **Manfaat Kalibrasi**

- Menjaga kondisi instrumen ukur dan bahan ukur agar tetap sesuai dengan spesifikasinya.

- Untuk mendukung sistem mutu yang diterapkan di berbagai industri pada peralatan laboratorium dan produksi yang dimiliki.
- Bisa mengetahui perbedaan (penyimpangan) antara harga benar dengan harga yang ditunjukkan oleh alat ukur.

➤ **Prinsip Dasar Kalibrasi**

- Obyek Ukur (*Unit Under Test*).
- Standar Ukur (Alat standar kalibrasi, *Prosedur/Metrode* standar (Mengacu ke standar kalibrasi internasional atau prosedur yg dikembangkan sendiri oleh laboratorium yg sudah teruji (*diverifikasi*)).
- Operator/Teknisi (dipersyaratkan operator/teknisi yg mempunyai kemampuan teknis kalibrasi (bersertifikat)).
- Lingkungan yg dikondisikan (Suhu dan kelembaban selalu dikontrol, gangguan faktor lingkungan luar selalu diminimalkan & sumber ketidak pastian pengukuran).

➤ **Kalibrasi Diperlukan Untuk**

- Perangkat baru.

- Suatu perangkat setiap waktu tertentu.
- Suatu perangkat setiap waktu penggunaan tertentu (jam operasi).
- Ketika suatu perangkat mengalami tumbukan atau getaran yang berpotensi mengubah kalibrasi.
- Ketika hasil pengamatan dipertanyakan.

Kalibrasi semua sensor yang digunakan menggunakan *Arduino Development Environment*, sebelum kalibrasi dijalankan nantinya pada mikrokontroler sudah diberikan script yang dimana bisa melihat data yang akan dimunculkan pada tampilan *Arduino Development Environment*. Jika sudah terkoneksi antara sensor, mikrokontroler dan PC nantinya data akan dicari titik yang diinginkan untuk memulai dan ukur.

- Kalibrasi Sensor Kecepatan

Kalibrasi sensor kecepatan menggunakan *Arduino Development Environment*. Rangkaian sensor kecepatan yang berupa LED infra merah dan photodiode akan dihubungkan pada mikrokontroler, dalam mikrokontroler akan diberikan *Script* tentang kalibrasi kecepatan. Pada *Arduino Development Environment* akan diupload *Script* yang sudah jadi, selanjutnya rangkaian sensor kecepatan akan dihubungkan pada mikrokontroler dan yang

menghubungkan mikrokontroler pada PC menggunakan USB. Berikut script yang ter-upload pada mikrokontroler untuk kalibrasi sensor kecepatan.

```
int port_kemudi = A0;
int port_kecepatan = A1;
int port_rem = A2;
int kemudi = 0;
int kecepatan = 0;
int rem = 0;

void setup() {
  pinMode(port_kemudi, INPUT);
  pinMode(port_kecepatan, INPUT);
  pinMode(port_rem, INPUT);

  Serial.begin(9600);
}

void loop() {
  kemudi = analogRead(port_kemudi);
  kecepatan = analogRead(port_kecepatan);
```

```
rem = analogRead(port_rem);

kemudi = map(kemudi, 0, 1023, 0, 180);
kecepatan = map(kecepatan, 0, 1023, 0, 180);
rem = map(rem, 0, 1023, 0, 180);

Serial.print(kemudi);
Serial.print(",");
Serial.print(kecepatan);
Serial.print(",");
Serial.println(rem);

delay(100);
}
```

Potongan kode diatas adalah kalibrasi kecepatan

Keterangan program diatas adalah script sensor kecepatan yang sederhana di dalam mikrokontroller sehingga bisa dihubungkan ke PC, rangkaian sensor ke mikrokontroller dan langsung di sambungkan ke PC, maka akan diketahui kalibrasi data dari sensor tersebut.

- Kalibrasi Sensor Kemudi

Kalibrasi sensor kemudi dengan cara memberikan *script* pada mikrokontroller di dalamnya sudah ada library ini adalah library nya: `#include <Servo.h>` dan juga *script* jenis sensornya, sehingga bisa memilih jenis apa sensor yang akan digunakan, dalam hal ini peneliti menggunakan potensiometer untuk sensor kemudi pada *Arduino Development* yang sudah ter-*instal* pada computer. Di bawah ini adalah script nya:

```
#include <Servo.h>
unsigned int counter=0;
Servo myservo; // create servo object to control a servo

int sensor_kemudi = 0; // analog pin used to connect the potentiometer
int nil_kemudi;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  nil_kemudi = analogRead(sensor_kemudi); // reads the value of the
  potentiometer (value between 0 and 1023)
  nil_kemudi = map(nil_kemudi, 0,1023, 0,180); // scale it to use it with
```

the servo (value between 0 and 180)

```
Serial.print(nik_kemudi);
```

```
Serial.print(",");
```

```
delay(50);
```

```
}
```

Potongan kode diatas adalah kalibrasi sensor kemudi

- Kalibrasi Sensor Rem

Untuk kalibrasi sensor Rem hampir sama dengan sensor kemudi namun ada perbedaan dalam perhitungannya, berikut script kalibrasi sensor Rem :

```
Int sensor_rem = A1;
```

```
Int nil_rem;
```

```
Void loop()
```

```
{
```

```
nil_rem = analogRead (sensor_rem);
```

```
nil_rem = map (nil_rem,0, 1023, 0, 180);
```

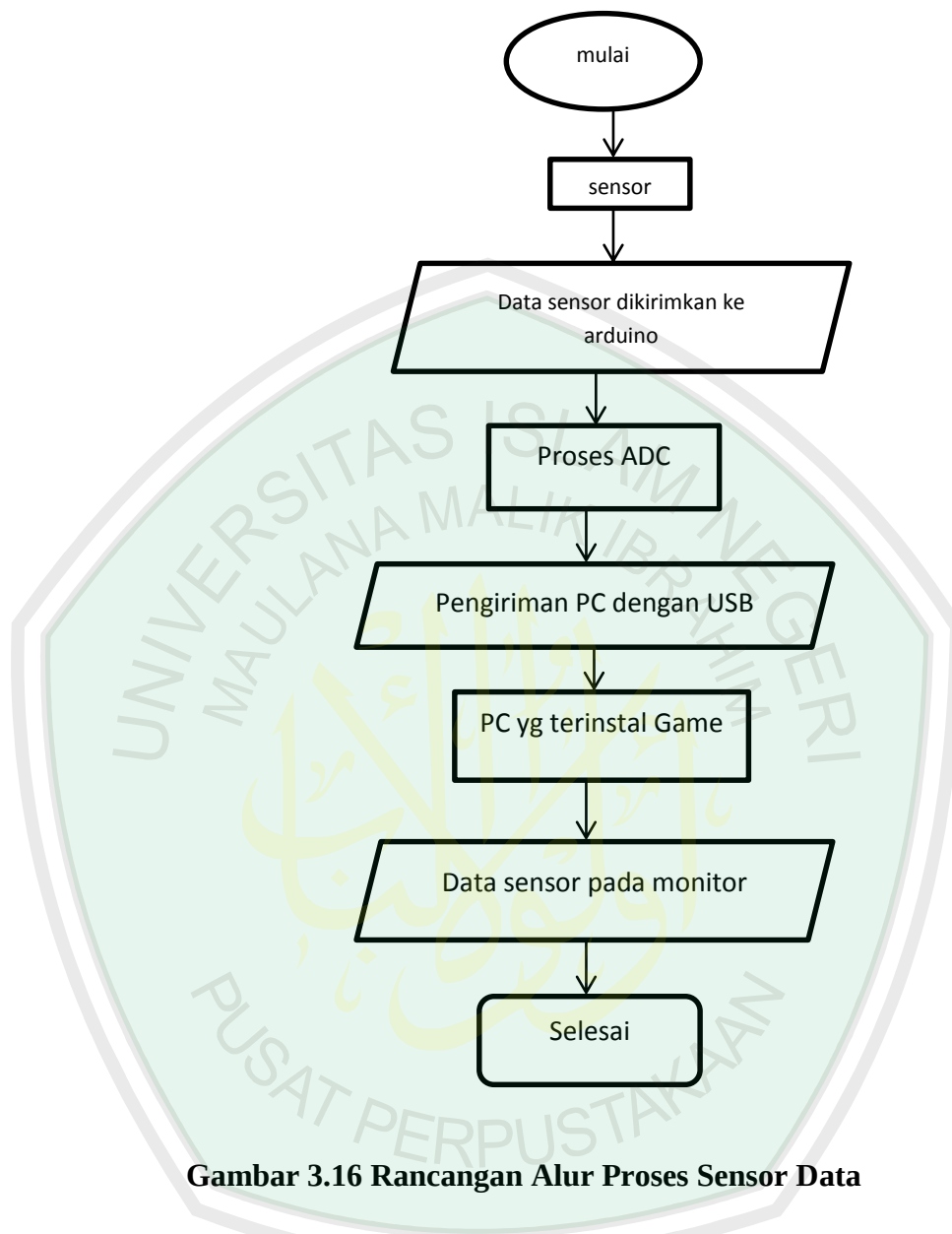
```
Serial.print(",");  
Serial.println (nil_rem);  
  
}
```

Potongan kode diatas adalah kalibrasi sensor Rem

3.2 Perancangan Alur Proses

3.2.1. Perancangan alur proses pengiriman data multisensory

Dalam perancangan alur Proses data Multisensor ini adalah alur pengiriman data sensor ke PC dimana sensor yang sudah disimulasikan untuk simulasi sepeda yang sudah dirangkai dengan baik untuk simulasi sepedanya, berikut alur proses pengiriman sensor ke PC dengan menggunakan USB

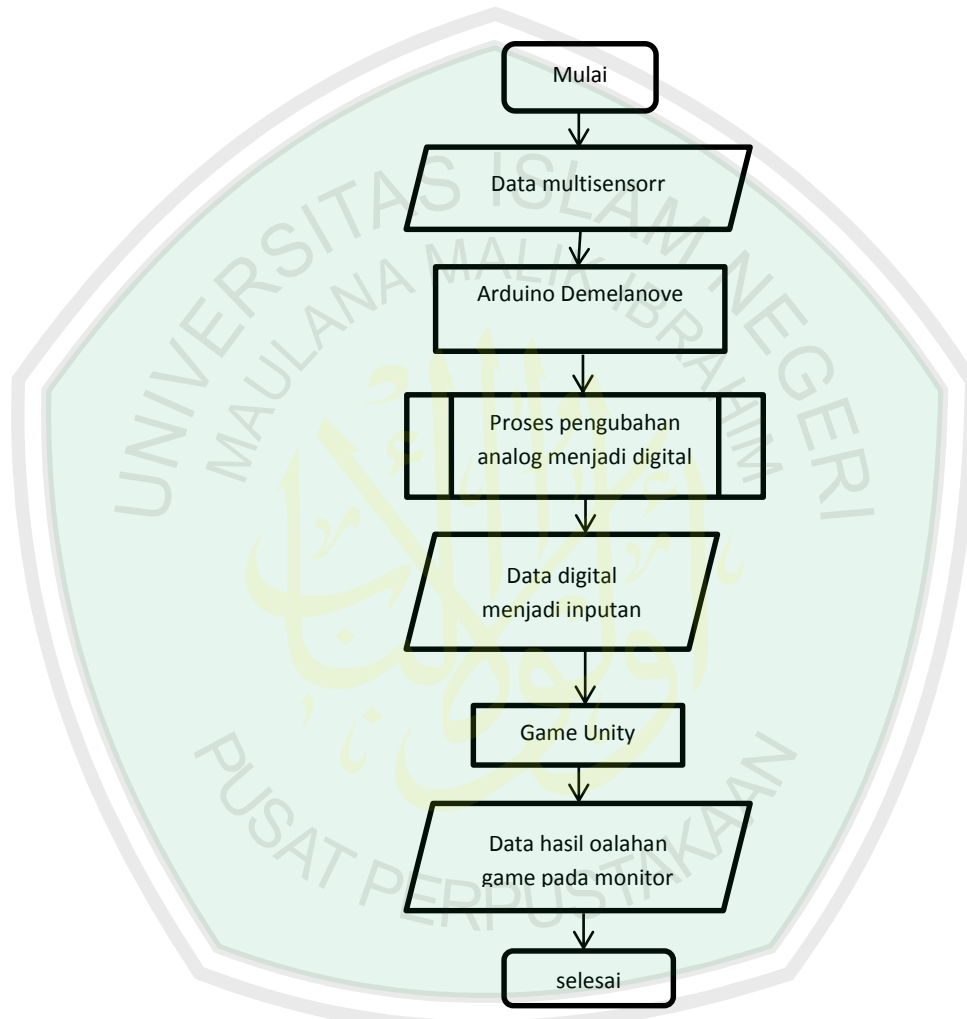


Gambar 3.16 Rancangan Alur Proses Sensor Data

Dari gambar di atas dapat dijelaskan semua sensor akan mengirimkan data ke mikrokontroler, pada mikrokontroler data tersebut akan diolah menjadi data digital dan dikirim menggunakan pengiriman data serial asinkron menggunakan USB. Setelah data terdeteksi sebagai inputan *game* maka data akan diproses

pada *game* dan pada *game* akan ditampilkan pada layar monitor pada permainan sepeda elektrik.

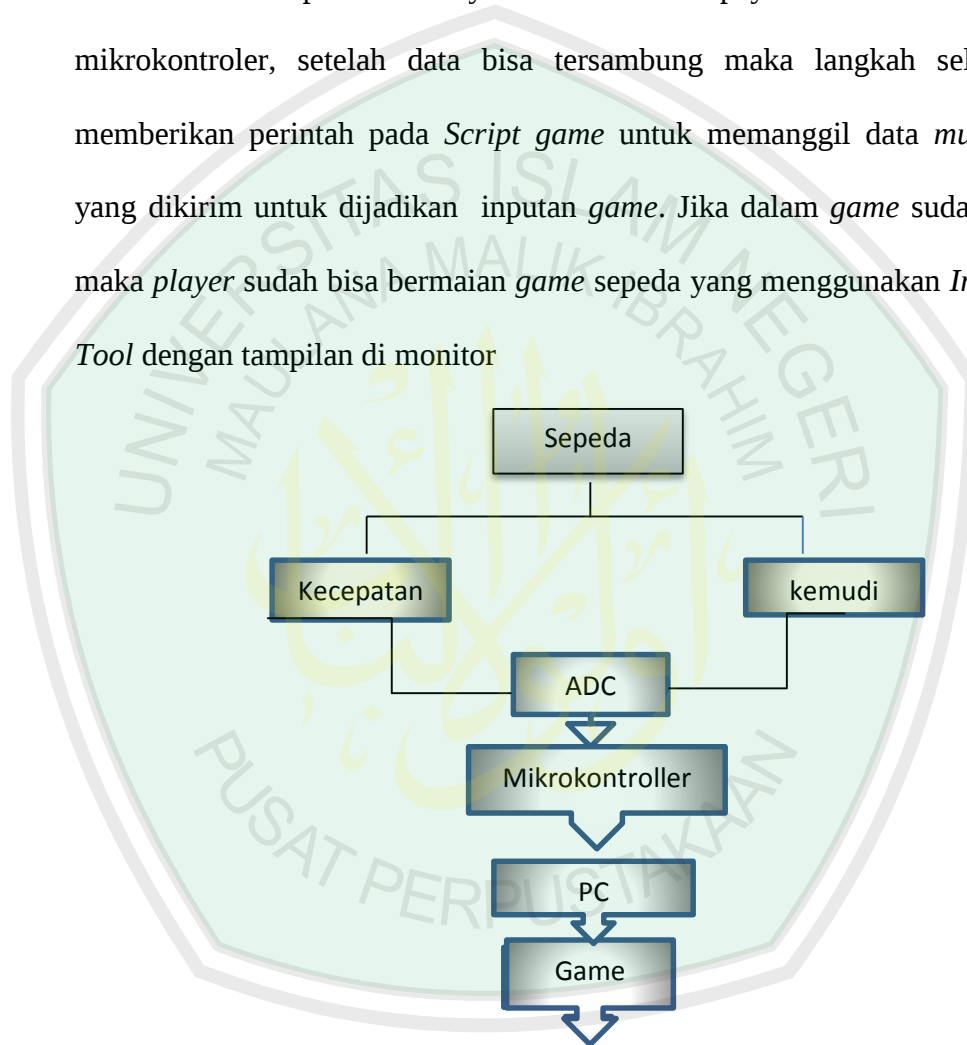
Proses pengolahan data *multisensor*



Gambar 3.17 Proses Pengolahan Data Multisensory

Dari gambar proses di atas dapat diketahui bahwa data *multisensor* yang masih analog akan diubah ADC untuk dijadikan data digital sebelum diolah pada mikrokontroler. Setelah data dari *multisensor* menjadi digital maka akan diolah pada mikrokontroler yang dimana dalam mikrokontroler

sudah di *upload script* menggunakan *Arduino Development Environment*,. Data yang sudah diproses oleh mikrokontroler akan dikirim ke PC menggunakan komunikasi serial *asinkron* yang menghubungkan dengan USB. Pada komputer nantinya akan diatur supaya bisa koneksi dengan mikrokontroler, setelah data bisa tersambung maka langkah selanjutnya memberikan perintah pada *Script game* untuk memanggil data *multisensor* yang dikirim untuk dijadikan inputan *game*. Jika dalam *game* sudah selesai maka *player* sudah bisa bermain *game* sepeda yang menggunakan *Immersive Tool* dengan tampilan di monitor



Gambar 3.18 Diagram Blok

Dari diagram blok diatas dapat dijelaskan bahwa *immersive tool* ini nantinya sebagai alat kontrol simulasi, yang mana nantinya terdapat sensor kemudi, dan sensor

kecepatan. Data hasil sensor ini akan diolah pada mikrokontroler, yang mana sebelum diolah mikrokontroler nantinya data hasil sensor yang masih analog akan diubah menjadi digital oleh ADC. Hasil data dari mikrokontroler ini nantinya yang berupa *String* akan dikirim ke PC yang sudah di *install* game, yang nantinya permainan ini akan ditampilkan pada layar monitor.



BAB IV

HASIL DAN PEMBAHASAN

Secara umum, pengujian ini bertujuan untuk mengetahui apakah alat yang dapat dibuat bekerja sesuai dengan spesifikasi perencanaan yang telah ditentukan. Pengujian dilakukan untuk mengetahui kinerja alat yang dipakai, karena alat juga pasti akan mengalami penurunan kualitas.

4.1 Hasil Uji Coba

4.2.1 Hasil kalibrasi

Hasil Kalibrasi sensor tentunya akan menggunakan *Arduino Development Environment*, berikut hasil kalibrasi dari tiap sensor.

1. Hasil Kalibrasi sensor kemudi

Data yang akan diperoleh ialah value kalibrasi -90 (belok kiri) dan 90 (belok kanan), kalibrasi kemudi dengan tahap setir/setang kemudi dibelokkan ke kiri 90^0 , di belokkan kanan -90^0 dan Lurus berada tepat $= 90$ kemudian tampil di layar serial monitor

```

Void loop () {

    Nil_kemudi = analogRead(sensor_kemudi);

    Nil_kemudi = map(val, 0, 1023, 0, 180);

    Myservo.write(val);

    Serial.print(val);
    Serial.print(",");
    Serial.println(rotation);
}

```

Potongan kode diatas adalah hasil kalibrasi kemudi di arduino

Dari potongan kode diatas terdapat sebuah fungsi string yang dapat memisahkan bagian-bagian dari string tersebut berdasarkan delimiter tertentu.yaitu fungsi split merupakan untuk membagi string tersebut menjadi bagian-bagian tertentu berdasarkan karakter pembatas (delimiter). Fungsi ini mengmbalikan sebuah *array of string* dari sebuah string yang berisi substring yang dibatasi oleh elemen dari array karakter Unicode tertentu.

```

data = value.Split(',')

if (int.TryParse (data [0], out kemudi) == false) {

    kemudi = 0;

}

Serial.print(nil_kemudi);

Serial.print(",");

Serial.print(kecepatan);

```

```
Serial.print(",");  
  
Serial.println(nil_rem);
```

Potongan kode diatas adalah data split kemudi

Kemudian diatur di monodevelop sesuai dengan data yang sudah di kirim dari arduino dengan mengurangi putaran steer dari titik (0) sampai titik (180) menjadi (-90) agar posisi (90) pada titik lurus, (0-90) untuk belok kiri, dan untuk belok kanan (90-180) bertujuan untuk memudahkan proses kalibrasi, dengan memberikan script pada program di unity seperti :

```
dataArduino = this.GetComponent("DataArduino");  
  
RodaDepan.steerAngle = dataArduino.kemudi -90;
```

Potongan kode di atas pengaturan kemudi di unity

Sedangkan pada kalibrasi Rem prosesnya hampir sama dengan sensor kemudi namun perhitungan berbeda, pada nilai 0 sepeda berhenti

```
Void loop ()  
  
{
```

```

nil_rem = analogRead (sensor_rem);

nil_rem = map (nil_rem, 0, 1023, 0, 180);

Serial.print(",");

Serial.println(nil_rem);

}

```

Potongan kode diatas adalah kalibrasi Rem

```

Function FixedUpdate() {
    dataArduino = this.GetComponent ("DataArduino");
    rem = dataArduino.rem/10;
    if (rem > 0){
        rem = max_rem;
    }
    If(rem > 0){
        Kecepatan = kecepatan - rem;
        If(kecepatan < 0){
            Kecepatan = 0; }
    }
    RodaBelakang.brakeTorque = rem;
}

```

Potongan kode diatas adalah script di unity

2. Hasil Kalibrasi Sensor Kecepatan

Untuk kalibrasi kecepatan dengan posisi sensor tanda putih (1) berjumlah 7 dalam satu kali putaran pada roda simulasi, dengan script di *mikrokontroller* sebagai berikut :

```
void timerIsr()
{
  Timer1.detachInterrupt(); //stop the timer
  int rotation = (counter / 7); // divide by number of holes in Disc
  counter=0; // reset counter to zero
  Timer1.attachInterrupt( timerIsr ); //enable the timer
}
```

Potongan kode diatas adalah hasil kalibrasi kecepatan di arduino

Pada kecepatan ini akan didapat dari pembacaan sensor yang sudah dikalibrasi dari *mikrokontroller*.

```
Function fixedUpdate (){
  dataArduino = this.GetComponent ("DataArduino");
  kecepatan = dataArduino.kecepatan/10;

  RodaBelakang.motorTorque = kecepatan;
}
```

Potongan kode diatas adalah script sensor kecepatan di unity

```

data = value.Split(',')
if (int.TryParse (data [1], out kecepatan) == false) {
    kecepatan = 0;
}

```

Potongan kode diatas adalah data split kecepatan

4.2.2 Hasil uji komunikasi serial Asinkron

Data hasil Perhitungan di *mikrokontroller* untuk mengirim data dengan mendeskripsikan serial port di unity disesuaikan dengan *port* di *mikrokontroller* arduino sehingga bisa terhubung satu sama lain.

```

SerialPort stream = new SerialPort ("COM1", 9600);
String [] data ;

Public int kemudi = 0 ;
Public double kecepatan = 0;
Public double rem = 0;

Void start () {
    Stream.open()
}

Void update(){
    String [] data;

    Public value = stream.ReadLine();

```

```

        if (int.TryParse (data [1], out kecepatan) == false) {
            kecepatan = 0 ;}

```

Potongan kode diatas adalah pengiriman data dari arduino ke unity

```

data = value.Split(',');
        if (int.TryParse (data [0], out kemudi) == false) {
            kemudi = 0;
        }
        if (int.TryParse (data [1], out kecepatan) == false) {
            kecepatan = 0;
        }

```

Potongan kode diatas adalah data split

- Pada *transmisi Asinkron*, sebelum terjadi komunikasi, tidak diadakan sinkronisasi *clock* antara pengirim dan penerima
- Dalam pengiriman data, memanggil data sesuai urutan indeks nya
- Data dikirim per karakter dan masing-masing karakter memiliki bit start (0) dan bit stop (1)

- Maka diketahui pengiriman pertama yaitu 0, dan di jadikan 0000 0100 menjadi bit. Untuk mengubah menjadi data bit adalah *protocol* di *mikrokontroller*
- Pada proses ini *transmisi asinkron* dijalankan untuk memberikan startbit(0)+stopbit(1) maka menjadi 0000001001.

Start bit berfungsi untuk menandakan adanya rangkaian bit karakter yang siap dicuplik dan *Stop bit* berfungsi untuk melakukan proses menunggu karakter berikutnya. dan setiap karakter terdiri dari 10 bit dengan rincian 1 bit start bit, 1 bit stop bit, dan 7 bit data.

- Maka menjadi 0000001001. Data bit yang dikirim pada *transmisi asinkron*

Proses Penerima

- Data bit yang diterima dari *transmisi asinkron* 0000001001 masih data bit.
- Maka data akan diubah menjadi data bit semula, dengan Dbit = terima-startbit-stopbit. Data akan kembali menjadi 0000 0100
- Maka data bit 0000 0100 akan di ubah kembali pada *protocol* menjadi 4 dalam satuan decimal.

Proses ini akan terus menerus dikirim data dari mikrokontroller masih ada dengan kecepatan datanya 9600.

Dalam proses mengubah bilangan decimal menjadi bit dengan *protocol* yang sudah tersedia pada mikrokontroller Arduino Duemilanove dan juga pada PC. Sehingga data akhir dari pengiriman inilah yang akan diproses pada game menjadi inputan.

4.2.3 Hasil uji coba statistic data

Pada uji coba ini akan membahas tentang hasil data statistic dari sensor kecepatan kemudi dan rem, dimana memakai sensor kecepatan infra merah dan photodiode dan sensor potensiometer. Pada Proses ini akan dilakukan pembentukan terhadap nilai dari nilai kemudi, nilai kecepatan, dan nilai Rem.

a. Nilai Kecepatan

Nilai kecepatan di ambil dari data nilai pengamat yang sesuai , pada data yang sesuai dengan kategori pelan memiliki nilai 1 untuk nilai dari 0 – 60 , nilai 2 kategori sedang memiliki nilai 50 – 100 , dan nilai 3 kategori cepat memiliki nilai 100 – 170.

Tabel 4.1 Nilai Hasil Uji Kecepatan

NO	Kecepatan (Km/s)				
	Display	Pengamatan	Nilai	kategori	Kesesuaian
1.	165	165	3	cepat	Sesuai
2.	25	70	2	Sedang	Tdk Sesuai
3.	50	150	3	Cepat	Tdk Sesuai
4.	170	170	3	Cepat	Sesuai
5.	170	20	1	Pelan	Tdk Sesuai

b. Nilai Kemudi

Nilai Kemudi di ambil dari data kemudi pada game dan data pada sensor dalam derajat , berikut table hasil uji kemudi :

Tabel 4.2 Nilai Hasil Uji Kemudi

No	KEMUDI (dalam drajat)		
	Display	Pengamatan	Kesesuaian
1.	170	170	Sesuai
2.	159	159	Sesuai
3.	135	135	Sesuai
4.	90	90	Sesuai
5.	59	59	Sesuai

c. Nilai Rem

Nilai Rem di ambil dari data di display dengan nilai pengamat yang sesuai , pada data yang sesuia di ambil dari nilai >100 dengan kategori berhenti memiliki nilai 1, nilai 2 kategori tidak behenti memiliki nilai 1 – 100.

Tabel 4.3 Nilai Hasil Uji Rem

NO	Rem (Km/s)				
	Display	Pengamatan	Nilai	kategori	Kesesuaian
1.	110	110	1	Berhenti	Sesuai

2.	22	22	2	Tidak berhenti	Sesuai
3.	130	130	1	Berhenti	Sesuai
4.	90	90	1	Berhenti	Sesuai
5.	45	45	1	Tidak berhenti	Sesuai

1. Hasil gambar uji pertama

Pengujian pertama yang masing-masing memiliki nilai kemudi dengan nilai 94, kecepatan dengan nilai 53, dan nilai rem 51.



Gambar 4.1 Hasil Uji Sensor Pada Game

1. Hasil gambar Uji kedua

Pengujian kedua yang masing-masing memiliki nilai kemudi 98, nilai kecepatan 59, dan nilai rem 65.



Gambar 4.2 Hasil Uji Sensor Pada Game

2. Hasil gambar Uji ketiga

Pengujian ketiga dengan masing-masing nilai, untuk nilai kemudi 75, nilai kecepatan 66, dan nilai rem 77



Gambar 4.3 Hasil Uji Sensor Pada Game

Gambar hasil uji diatas merupakan hasil data sensor dengan data tampilan di game elektrik,

Dalam Pengamatan dari hasil kemudi berikut table uji percobaannya

Tabel 4.4 Hasil Uji Percobaan dan Pengamatan

No.	INPUT (Rps)						Kesimpulan
	DISPLAY			PENGAMATAN			
	Kem (drajat)	Rem(km/s)	Kec (Km/s)	Kem (drajat)	Rem(km/s)	Kec(Km/s)	
1.	170	110	10	170	110	170	Tidak berhasil
2.	159	22	57	159	22	170	Tidak berhasil
3.	135	130	89	135	130	170	Tidak berhasil
4.	90	90	140	90	90	170	Tidak berhasil
5.	59	45	162	59	45	162	Tidak berhasil
6.	10	35	170	10	35	30	Tidak berhasil
7.	12	30	145	12	19	22	Tidak berhasil

8.	90	42	130	90	19	19	Tidak berhasil
9.	90	120	125	90	19	19	Tidak berhasil
10.	70	8	160	70	32	32	Tidak berhasil
11.	70	70	35	70	35	35	Berhasil
12.	70	50	40	70	40	40	Berhasil
13.	65	60	42	65	20	42	Berhasil
14.	65	25	45	65	20	45	Berhasil
15.	65	105	50	65	30	50	Berhasil
16.	84	38	54	84	30	54	Berhasil
17.	84	44	56	84	30	56	Berhasil
18.	84	27	58	84	30	58	Berhasil
19.	78	32	60	78	30	60	Berhasil
20.	110	62	63	110	62	63	Berhasil
21.	110	65	67	110	65	67	Berhasil
22.	110	71	73	110	71	73	Berhasil

23.	110	173	177	110	173	177	Berhasil
24.	50	167	170	50	167	170	Berhasil
25.	50	171	175	50	171	175	Berhasil
26.	50	159	177	50	159	177	Berhasil
27.	35	132	167	35	132	167	Berhasil
28.	35	141	165	35	141	165	Berhasil
29.	35	120	165	35	120	165	Berhasil
30.	35	114	170	35	114	170	Berhasil
31.	100	104	164	100	104	164	Berhasil
32.	100	93	166	100	93	166	Berhasil
33.	100	87	168	100	87	168	Berhasil
34.	100	78	177	100	78	177	Berhasil
35.	15	79	174	15	79	174	Berhasil
36.	15	80	174	15	80	174	Berhasil
37.	15	65	175	15	65	175	Berhasil
38.	69	68	176	69	68	176	Berhasil
39.	69	55	170	69	55	170	Berhasil

40.	69	38	177	69	38	177	Berhasil
41.	69	25	167	69	25	167	Berhasil
42.	92	20	165	92	20	165	Berhasil
43.	92	18	165	92	18	165	Berhasil
44.	92	12	170	92	12	170	Berhasil
45.	92	124	164	92	124	124	Berhasil
46.	135	170	166	135	170	166	Berhasil
47.	135	159	168	135	159	168	Berhasil
48.	135	135	158	135	135	158	Berhasil
49.	135	90	175	135	90	175	Berhasil
50.	150	59	177	150	59	177	Berhasil

Dari hasil Tabel uji percobaan di atas dapat disimpulkan keberhasilan dan kegagalan project, untuk keberhasilan nya jumlah keberhasilan di bagi dengan total data percobaan di bagi 100 %

$$\frac{\text{Jumlah Keberhasilan}}{\text{Total data percobaan}} \times 100 \% = \frac{40}{50} \times 100 \% = 80 \%$$

Kemudian untuk kegagalan juga sama di hitung dari jumlah kegagalan di bagi dengan total data percobaan dikalikan 100 %, berikut hasil nya:

$$\frac{\text{jumlah kegagalan}}{\text{Total data percobaan}} \times 100 \% = \frac{10}{50} \times 100 \% = 20 \%$$

Peneliti Mengamati dalam proses kegagalan pada simulasi roda, yang dalam hal ini simulasi roda menggunakan Kipas ADC 12 Volt pada ADC ini putarannya konstan. Untuk nilai kecepatan melaju sepeda , dalam kecepatan hanya bisa menggunakan Potensiometer yang dalam hal ini untuk di karenakan terjadi adanya system yang kurang stabil, karena alat yang dijadikan simulasi seperti kipas yang memiliki 12 volt ini karena putaran kipas nya yang konstan.

4.2 Koneksi data multisensory ke game

Koneksi data dari multisensory yang merupakan inputan dari game diperoleh dari data digital dari mikrokontroler yang akan dikirim melalui USB ke PC. Pengiriman data serial menggunakan USB dari mikrokontroller menuju ke game dengan memanggil port serial yang terdapat pada script game, berikut script yang terdapat pada game untuk koneksi dengan mikrokontroller.

```
using UnityEngine;
using System.Collections;
using System.IO.Ports;
```

```

public class DataArduino : MonoBehaviour
{

    SerialPort stream = new SerialPort("COM1", 9600); //Set the port (com4) and the
    baud rate (9600, is standard on most devices)

    string [] data;

    public int kemudi = 0;
    public double kecepatan = 0;
    public double rem = 0;

    void Start ()
    {
        stream.Open();
        //sp.ReadTimeout = 1;
    }
    void Update ()
    {
        string value = stream.ReadLine(); //Read the information
        data = value.Split(',');

        if (int.TryParse (data [0], out kemudi) == false) {
            kemudi = 0;
        }

        if (double.TryParse (data [1], out kecepatan) == false) {
            kecepatan = 0;
        }
    }
}

```

```

        if (double.TryParse (data [2], out rem) == false) {

            rem = 0;

        }

    }

    void OnGUI()

    {

        GUI.Label(new Rect(10,50,300,100), " Kemudi:" + kemudi + ",
kalibrasi:"+ (kemudi-90));

        GUI.Label(new Rect(10,100,300,100), " Kecepatan: " + kecepatan + ",
kalibrasi:"+ (kecepatan/10));

        GUI.Label(new Rect(10,150,300,100), " Rem: " + rem + ", kalibrasi:"+
(rem/10));

    }

}

```

Potongan kode diatas adalah memanggil *port* serial

Pada potongan kode diatas merupakan script untuk memanggil dan membaca data serial yang dikirim dari mikrokontroller, data yang merupakan data digital data sensor yang sudah di proses melalui ADC dan program pada mikrokontroller hanya pengaturan saja untuk mengatur dan mendefinisikan sensor – sensor yang sudah di rangkai di board dengan menggunakan arduino. program yang memakai sensor potensiometer dan photodiode ini akan menjadi inputan pada game.

Dalam pengaturan penyambungan dari mikrokontroller ke PC dengan menggunakan kabel USB hanya cukup mengatur serial port di mikrokontroller dan menyamakan pada game unity nya.

Misal “SerialPort (“COM1”, 9600)”.

Sedangkan untuk melihat tampilan kecepatan, kemudi, dan rem sepeda dari sensor yang sudah dikalibrasi hanya melihat dari layar monitor game nya.

```
void OnGUI()
{
    GUI.Label(new Rect(10,50,300,100), “ Kemudi:” + kemudi +
    “, kalibrasi:”+ (kemudi-90));

    GUI.Label(new Rect(10,100,300,100), “ Kecepatan: “ +
    kecepatan + “, kalibrasi:”+ (kecepatan/10));

    GUI.Label(new Rect(10,150,300,100), “ Rem: “ + rem + “,
    kalibrasi:”+ (rem/10));
}
```

Potongan kode label untuk tampilan kecepatan.kemudi, dan rem

4.3 Pembahasan

Dari Hasil pengujian penyederhanaan yang sudah di kembangkan oleh penelitian sebelumnya di dapat beberapa pembahasan yaitu:

1. Data dari sensor yang sudah di kalibrasi pada mikrokontroller akan dikirim ke PC dan merupakan inputan dari game sepeda, dalam pengiriman data serial ini menggunakan USB dan komonikasi serial Asinkron.
2. Hasil Data statistic nya berupa tampilan di layar monitor pada game sepeda, yang dapat mengetahui data kemudi dan data rem dan data kecepatan *player* dalam mengayuh sepeda.

4.4 Integrasi Dalam Islam

Suatu game pada dasarnya sebagai sarana dan prasarana hiburan, untuk menghilangkan kejenuhan dalam berbagai aktifitas yang padat, dan juga sebagai pembelajaran agar pemain bisa mengerti sedikit dari ilmu tajwid yang dapat diterapkan ketika dalam membaca ayat suci Al- Quran.

berikut ini integrasi game sepeda elektrik menggunakan Immersive Tools dengan Islam, pada ayat al quran menjelaskan tentang berolahraga.

وَأَعِدُّوا لَهُمْ مَا اسْتَطَعْتُمْ مِنْ قُوَّةٍ وَمِنْ رِبَاطِ الْخَيْلِ تُرْهِمُونَ بِهِ عَدُوَّ اللَّهِ وَعَدُوَّكُمْ وَآخَرِينَ مِنْ دُونِهِمْ لَا

تَعْلَمُونَهُمْ اللَّهُ يَعْلَمُهُمْ وَمَا تُنْفِقُوا مِنْ شَيْءٍ فِي سَبِيلِ اللَّهِ يُوَفَّ إِلَيْكُمْ وَأَنْتُمْ لَا تُظْلَمُونَ

“dan persiapkanlah dengan segala kemampuan untuk menghadapi mereka dengan kekuatan apa saja yang kamu miliki dan dari pasukan berkuda yang dapat menggentarkan musuh Allah, musuhmu dan orang-orang selain

mereka yang kamu tidak mengetahuinya. Apa saja yang kamu nafkahkan di jalan Allah niscaya akan dibalas dengan cukup kepadamu dan kamu tidak akan didzalimi (dirugikan)". {Q.S Al-Anfal : 60}

Islam mengajarkan kita untuk menjaga kondisi badan agar tetap prima, karena muslim yang kuat lebih baik dan dicintai dari pada muslim yang lemah baik dari segi fisik maupun non fisik, untuk melatih kekuatan fisik ada hadist yang mengajarkan berenang agar terkoordinasi gerakannya. seperti pada gerakan kaki pada game sepeda bukan hanya gerakan jari jemari saja yang di gerakkan akan tetapi gerakan kaki juga. Game sepeda elektrik ini adalah simulasi sepeda yang nantinya bisa untuk pengembangan penelitian lanjut untuk game sepeda seperti sepeda pada umumnya. Berikut penjelasan hadist tentang ini :

حَدَّثَنَا أَبُو بَكْرِ الطَّلْحِي , حَدَّثَنَا أَحْمَدُ ابْنُ حَمَادٍ ابْنُ سُفْيَانَ , حَدَّثَنَا عَمْرُو بْنُ
عُثْمَانَ الْحَمَصِيُّ , حَدَّثَنَا ابْنُ عِيَّاشٍ , عَنْ سَلِيمِ بْنِ عَمْرٍو الْأَنْصَارِيِّ , عَنْ
عَمِّ أَبِيهِ , عَنْ بَكْرِ بْنِ عَبْدِ اللَّهِ بْنِ رَبِيعِ الْأَنْصَارِيِّ , قَالَ : قَالَ رَسُولُ اللَّهِ صَلَّى
اللَّهُ عَلَيْهِ وَسَلَّمَ : " علموا أبناءكم السباحة و الرماية , و نعم لهُوَ الْمُؤْمِنَةُ فِي
بَيْتِهَا الْمَغْزَلِ , و إِذَا دَعَاكَ أَبْوَاكَ فَأَجِبْ أَمَّاكَ "

Menceritakan kepada kami Abu Bakar Atthalahi dari Ahmad bin Hamad bin Sofyan , dari amru bin usman alhimsi dari ibnu i'yasy dari sulaiman bin amru al-anshari dari paman ayahnya dari Bakar bin Abdillah bin Rabi' al-anshari

berkata :berkata Rasulullah SAW. “ajarilah anak anakmu berenang dan melempar lembing, termasuk juga perempuan perempuan di rumahnya menenun, dan apabila kedua orangtuamu memanggil maka utamakan ibumu. **(HR. Ath-Thahawi).**

Dalam hadits di atas, Rasulullah Saw memerintahkan kepada umat Islam untuk latihan berenang, karena latihan berenang adalah sebuah olahraga yang dilaksanakan untuk melatih pernafasan dan melatih kekuatan kaki dan tangan. Sama dengan game sepeda elektrik ini dapat melatih fisik agar lebih kuat.



BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil implementasi pengujian yang dilakukan peneliti, maka dapat ditarik kesimpulan yang diperoleh dari tugas akhir ini sebagai berikut :

1. Dalam Pengiriman data mikrokontroller ke PC dengan komunikasi data asinkron yang lebih sederhana yaitu menggunakan USB, sehingga data berlangsung jauh lebih cepat dibandingkan dengan penggunaan port type lainnya.
2. Pada simulasi game sepeda elektrik menggunakan sensor *Potensiometer* sebagai sensor posisi cukup praktis karena hanya membutuhkan satu tegangan eksitasi dan biasanya tidak membutuhkan pengolahan sinyal yang rumit
3. Dengan Berbasis Statistik pada game simulasi sepeda ini dapat diketahui kecepatan tiap detik dengan rumus menghitung kecepatan.
4. Dalam game sepeda, menggunakan *immersive Tools* dapat berjalan sesuai dengan kecepatan pemain membelokkan kemudi ke kanan dan membelokkan kemudi ke kiri, dan mengerem jika diperlukan.

5.2 Saran

Dalam Penelitian pengembangan ini sangat sederhana yang masih terdapat beberapa kekurangan dan untuk pengembangan lebih lanjut terdapat saran-saran sebagai berikut ini

1. Dalam satu *game* bisa diberikan dua pemain atau lebih, jadi bisa *multiplayer*.
Sehingga *player* bisa merasakan seakan-akan bersepeda bersama atau juga bisa mengadu kecepatan.
2. Pada Game Sepeda, di dalam Terrain game bisa ditambahkan jalan menanjak agar pemain bersepeda lebih menantang.

DAFTAR PUSTAKA

Triadmadya, O. 2014. *Simulation Games Bike Speed Settings Using Fuzzy Methods*. Department of Informatics, Faculty of Science and Technology of the State Islamic University of Maulana Malik Ibrahim Malang.

Erwanto, H. 2014. *Navigasi player untuk pencarian obstacle Pada game sepeda menggunakan Metode pathfinding A**. Department of Infomatics. Faculty of Science and Technology of the State Islamic University of Maulana Malik Ibrahim Malang.

Mussuga, P. 2014. *Design immersive tool multisensor pada game sepeda berbasis statistik dan komunikasi Data asinkron*. Department of Informatics, Faculty of Science and Technology of the State Islamic University of Maulana Malik Ibrahim Malang.

Rachman, A. Suhartono & Vincent. Purwanto, Yuliman. 2010 *Agen Cerdas Animasi Wajah Untuk Game Tebak Kata*. Pascasarjana Teknik Informatika, Universitas Dian Nuswantoro.

Arduino. (n.d.). *atmega 1280 from microcontroller* (Online).

(<http://arduino.cc>) di akses 16 april 2015.

Unitygems. *Pengenalan Unity from unitygems* (Online).

(<http://unitygems.com>) diakses 27 Mei 2015.

Oxford, "game engine: definition of game engine in Oxford Dictionary (British & World English)," 28 November 2015. [Online]. Available:

- Ghazali. Muh imam, Samopa. Feb, Sani. Nisfu.A, 2015. *Pengembangan Peta Interaktif Tiga Dimensi Gedung Rektorat Institut Teknologi Sepuluh Nopember Menggunakan Unity 3D Engine*. Fakultas Teknologi Informasi, Institut Teknologi Sepuluh November.
- Report and Recommendations to the Northcarolina State Board of Education. 2009. *Immersive Technology Gaming, Simulations, & Virtual,Environments in education*. Nc Department of Public Instruction June St. Clair Atkinson, Ed.D., State Superintendent.
- Zakiri, A. 2009. *Pengenalan Simulasi computer dalam arsitektur menggunakan The SimsTM 3*. Program studi ArsitekturDepok.
- Soegihardjo, O. 2001. *Simulasi Komputer untk Analisis Karakteristik Model Sistem Pegas-peredam Kejut-Massa*. Fakultas Teknologi Industri Jurusan Teknik Mesin Universitas Kristem Petra.
- Schroede, R. 2008. *Virtual Worlds Research : Past, Present, & Future*. *Journal Virtual Worlds Research Vol, 1. No. 1* (Online), diakses 27 desember 2015.
- Stallings, W. 2001. *Komunikasi data dan computer*. Jakarta: Salemba Teknika.