

BAB 2

TINJAUAN PUSTAKA

1.1 Profil Instansi

Akademi Olahraga Prestasi Nasional (AKORNAS) merupakan lembaga pendidikan tinggi dalam lingkungan kementerian riset, teknologi, dan pendidikan tinggi. AKORNAS bertempat di Jakarta yang menyelenggarakan program studi Diploma-III ilmu kepelatihan olahraga dengan gelar Ahli Madya (A.Md). Lulusan AKORNAS dibekali keahlian kecabangan yang siap bersinergi dengan KONI pusat dan induk cabang olahraga.



Gambar 0.1 Logo AKORNAS

2.1.1 Visi dan Misi Instansi

1.1.1.1 Visi

Visi Akademi Olahraga Prestasi Nasional (AKORNAS) adalah:

1. Menjadikan Akademi Olahraga Prestasi Nasional selalu dinamis dan inovatif terhadap Ilmu Pengetahuan dan Teknologi di bidang keolahragaan melalui penelitian, pengembangan dan kerjasama dengan berbagai pihak baik di dalam negeri maupun luar negeri dan menjadi lembaga pendidikan tinggi yang sehat, unggul dan sejahtera dalam menghasilkan tenaga pelatih, wasit, juri dan manajer olahraga yang profesional serta menjadi lembaga

rujukan di bidang keolahragaan yang dilandasi oleh nilai-nilai ilmiah, religiusitas, edukatif, dan profesional.

1.1.1.2 Misi

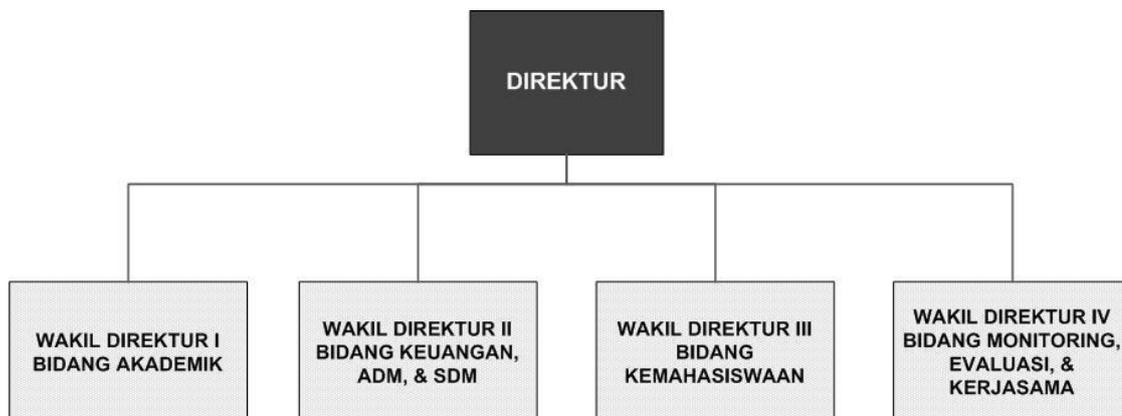
Misi Akademi Olahraga Prestasi Nasional (AKORNAS) adalah:

Menyelenggarakan pendidikan kepelatihan olahraga yang bermutu tinggi dan pembinaan kemahasiswaan yang komprehensif dalam rangka melahirkan para pelatih olahraga profesional.

1. Mengkaji dan mengembangkan Ilmu Pengetahuan dan Teknologi olahraga di bidang kepelatihan sesuai kebutuhan masyarakat olahraga baik di masa kini sampai 10 tahun mendatang.
2. Menyelenggarakan pengabdian kepada masyarakat dalam bidang kepelatihan olahraga dengan memasyarakatkan olahraga dan mengolahragakan masyarakat di tingkat daerah sampai dengan tingkat pusat.
3. Menyelenggarakan pendidikan berbasis praktek untuk menghasilkan lulusan yang berkualitas dan siap pakai untuk mencukupi kebutuhan pelatih olahraga prestasi baik masa kini sampai 10 tahun mendatang.
4. Membangun semangat nasionalisme yang tinggi sebagai kekuatan untuk melahirkan para pelatih olahraga profesional untuk mencapai prestasi dalam rangka meningkatkan harkat dan martabat bangsa.

2.1.2 Struktur Organisasi dan Uraian Tugas

Struktur organisasi Akademi Olahraga Prestasi Nasional (AKORNAS) dapat dilihat pada gambar 2.2 sebagai berikut.



Gambar 0.2 Struktur Organisasi AKORNAS

Di dalam Akademi Olahraga Prestasi Nasional, terdapat beberapa jabatan pada strukturalnya yang dipegang oleh masing-masing individu. Berikut adalah nama-nama tersebut:

- Pembina
 - Prof. Dr. Djaali
 - Dr. Abdul Sukur, S.Pd., M.Si.
 - E.F. Hamidy, MBA.
 - Mayjen TNI (Purn) Suwarno, S.IP., M.Sc.
 - Mayjen TNI Dody Usodo Hargo, S.Sip., M.M.
 - Dr. dr. Ekawahyu Kasih, S.H., M.M., M.H.
- Direktur
 - Mayjen TNI (Purn) Tono Suratman, S.IP.
- Wakil Direktur I – Bidang Akademik
 - Drs. Subroto, M.M.
- Wakil Direktur II – Bidang Keuangan, Adm & SDM
 - Drs. Johnny E. Awuy
- Wakil Direktur III – Bidang Kemahasiswaan
 - Ir. K. Inugroho, MM.
- Wakil Direktur IV – Bidang Monitoring, Evaluasi & Kerjasama
 - Surya Dharma, S.IP.

1.2 Landasan Teori

2.2.1 Sports Science

Ilmu Olahraga atau biasa disebut *sports science* merupakan suatu program ilmiah dari prinsip pengetahuan untuk membantu atlet dalam meningkatkan performanya. Tentunya teknologi pun dibutuhkan dalam pembinaan olahraga untuk membantu mengatasi masalah yang muncul dalam olahraga. Diskusi mendalam kebutuhan atlet secara individual menjadi salah satu hal yang spesifik bagi dasar pemikiran *sport science*. Untuk kebanyakan atlet, informasi didapat dari adanya pengukuran yang dilakukan secara berkala mengikuti program latihan yang telah dibuat [2].

2.2.2 Pencak Silat

Menurut Erwin Setyo Kriswanto dalam kamus Bahasa Indonesia, pencak silat merupakan keahlian dalam mempertahankan diri dengan kepandaian menangkis, menyerang dan membela diri atau tanpa senjata. Pencak silat juga merupakan sebuah metode beladiri warisan leluhur bangsa Indonesia yang diciptakan guna mempertahankan diri dari bahaya-bahaya yang mengancam keselamatan dan kelangsungan hidupnya [8].

Sedangkan menurut Djomali dalam sebuah jurnal yang disusun oleh E Latifah, A Rusdiana, dan S Ugelta mendefinisikan pencak silat adalah gerakan serang bela yang teratur menurut tempat, keadaan dan waktu. Pencak silat dapat didefinisikan bahwa “Pencak Silat adalah budaya manusia Indonesia untuk membela atau mempertahankan eksistensi (kemandirian) terhadap lingkungan hidup guna peningkatan iman dan taqwa terhadap Tuhan Yang Maha Esa”. [9]

Berdasarkan pendapat dan penjelasan beberapa ahli di atas, maka dapat disimpulkan bahwa olahraga pencak silat sebagai suatu keterampilan beladiri dan sebagai sarana dengan materi pendidikan rohani dan jasmani. Sikap jasmani adalah sikap kesiapan fisik tubuh untuk melakukan gerakan-gerakan dengan kemahiran teknik yang baik. Sedangkan sikap rohani adalah kesiapan mental dan pikiran untuk melakukan tujuan dengan waspada. Yang memiliki filosofi hidup yang diberi nilai-

nilai luhur pencak silat dan mempunyai kode etik yang biasa disebut dengan nama prasetya pencak silat.

2.2.3 Latihan Fisik

Latihan fisik termasuk unsur yang sangat penting pada seluruh cabang olahraga. Oleh karena itu latihan fisik perlu menjadi hal yang serius untuk diperhatikan dan direncanakan dengan matang serta sistematis sehingga tingkat kesegaran jasmani dan kemampuan fungsional alat-alat tubuh akan berkembang lebih baik. [10]

Menurut Yunyun Yudiana, Herman Subardjah, dan Tite Juliantine, *“Latihan kondisi fisik adalah proses mengembangkan kemampuan aktivitas gerak jasmani yang dilakukan secara sistematis dan ditingkatkan secara progresif untuk mempertahankan juga meningkatkan derajat kebugaran jasmani agar tercapai kemampuan kerja fisik yang optimal.”*

Unsur penting yang terkandung dalam latihan fisik salah satunya adalah kecepatan. Dalam olahraga pencak silat kecepatan tentu menjadi hal yang sangat dibutuhkan karena dengan kecepatan gerak yang tinggi lawan akan sulit menduga gerakan yang dilakukan.

Kecepatan tidak hanya dilakukan untuk menyerang lawan namun untuk menghindari serangan lawan juga. Dalam meningkatkan kecepatan pun atlet dapat melatih reaksi gerakan, mempercepat kapasitas gerakan, mengatur keseimbangan gerakan yang dapat dipertahankan maupun ditingkatkan.

2.2.4 Kecepatan Reaksi Pukulan

Kecepatan memegang peranan yang sangat penting dalam menunjang prestasi seorang atlet dimana atlet mempunyai kemampuan fisik untuk bergerak dalam waktu sesingkat mungkin. Dalam kegiatan olahraga ada tiga jenis kecepatan menurut Nossek yang diterjemahkan oleh Harsono [11] bahwa:

(1) *Sprinting of speed*, yaitu kemampuan bergerak maju ke depan dalam waktu yang singkat. Berhasilnya *sprinting of speed* tergantung dari kemampuan

untuk melakukan frekuensi pergantian kaki sebanyak mungkin dan setiap pergantian kaki (langkah) akan menghasilkan jarak yang sejauh mungkin.

(2) *Reaction of speed*, yaitu kemampuan reaksi dalam waktu yang sesingkat-singkatnya setelah menerima rangsangan.

(3) *Speed of movement*, adalah kemampuan kecepatan kontraksi otot terhadap suatu gerakan yang tidak terputus.

2.2.5 VO2 Max (Volume Oksigen Maksimal)

VO2 Max atau Volume Oksigen Maksimal mengacu pada kecepatan pemakaian oksigen, bukan sekedar banyaknya oksigen yang diambil (*Oxygen Uptake*) namun selama usaha maksimal yang dilakukan atau disebut juga penggunaan oksigen maksimal yang merupakan tempo tercepat dimana seseorang dapat menggunakan oksigen selama berolahraga [6].

Kondisi fisik sangat erat hubungannya dengan VO2 Max, karena VO2Max itu adalah tempo tercepat dimana seseorang dapat menggunakan oksigen selama berolahraga. Maka seseorang yang mempunyai VO2 Max yang baik dalam penggunaan oksigen akan lebih baik sehingga kesegaran jasmani akan baik pula, jadi VO2 Max juga mempengaruhi kondisi fisik seorang atlet atau kesegaran jasmani seseorang.

Pada penelitian ini VO2 Max menjadi hasil dari tes fisik lari 15 menit menggunakan metode balke test.

2.2.6 Balke Test

Menurut Sukadiyanto, tes ini merupakan cara untuk menghitung prediksi VO2 Max pada olahragawan atau atlet dengan menggunakan jarak tempuh lari selama 15 menit. Setelah berlari selama 15 menit, kemudian dicatat hasil jarak tempuh yang dicapai seorang atlet saat berlari selama waktu 15 menit tersebut [6]. Pelaksanaan tes ini dapat dilakukan seperti berikut:

- a) Tes dilakukan di lapangan atau lintasan lari 400 m yang jaraknya jelas.
- b) Penanda jarak atau bendera kecil untuk menandai jarak lintasan.

- c) *Stopwatch* atau alat pengukur waktu dalam satuan menit.
- d) Adapun protokol pelaksanaan tesnya adalah sebagai berikut; (1) Peserta tes berdiri di garis *start* dan bersikap untuk berlari secepat-cepatnya selama 15 menit. (2) Peserta memulai tes dengan aba-aba “Ya” dengan pencatat waktu mulai menghitung oleh pengetes. (3) Selama waktu 15 menit, pengetes memberi aba-aba berhenti, di mana bersamaan dengan itu *stopwatch* dimatikan. (4) Pengetes mengukur jarak yang ditempuh peserta tes yang telah ditempuh selama 15 menit, dengan meteran. Selanjutnya hasil jarak tempuh lari selama 15 menit dimasukkan ke dalam rumus sebagai berikut:

$$\text{VO2Max} = 33.3 + \text{Jarak tempuh}/15 - 133 \times 0.172$$

2.2.7 Python

Python adalah Bahasa pemrograman yang berfungsi melakukan eksekusi kode sumber dalam suatu Bahasa pemrograman, pada umumnya python digunakan sebagai Bahasa skrip. Tidak seperti Bahasa pemrograman pada umumnya yang harus dikompilasi oleh kompilator. Kompilasi hasil proses dalam file kode mesin, tidak dapat dibaca oleh manusia, bahwa komputer dapat membaca dan mengikuti. Bahasa skrip, di sisi lain, dibaca, ditafsirkan, dan ditindaklanjuti setiap kali *run* program. Bahasa skrip tidak menghasilkan file yang dikompilasi, dan instruksi diikuti persis seperti yang tertulis. Jika menulis kode yang asal, maka akan mendapatkan hasil yang asal juga. Untuk alasan ini, bahasa skrip dapat menghasilkan program yang lebih lambat. Namun python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi [9]. Pada penelitian ini Bahasa pemrograman python digunakan sebagai Bahasa pemrograman pada perangkat Raspberry.

2.2.8 Unified Modelling Language (UML)

Unified Modeling Language (UML) adalah sistem arsitektur yang dikembangkan oleh *Boach*, *Object Modelling Technique* (OMT) dan *Object Oriented Software Engineering* (OOSE). Pada pengembangannya, *Unified*

Modeling Language (UML) menjadi salah satu alat bantu yang sangat handal di dunia pengembangan sistem yang berorientasi obyek. Hal ini disebabkan karena UML menyediakan Bahasa pemodelan visual yang memungkinkan pengembang sistem menjadi lebih mudah dimengerti dan dilengkapi dengan mekanisme yang efektif untuk berkomunikasi rancangan yang dibuat [15].

Dokumentasi UML menyediakan 10 macam diagram untuk membuat model aplikasi berorientasi objek yang 4 di antaranya adalah:

1. *Use Case Diagram*

Use case diagram Mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan dengan digunakan *external actors*, *actor* yang berinteraksi dengan sistem dapat berupa user atau sistem lainnya.

2. *Class Diagram*

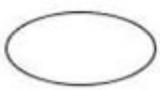
Use case diagram adalah gambaran sistem dari sudut pandang pengguna (*user*) sistem tersebut, sehingga pembuatan use case lebih dititik beratkan pada fungsional yang ada pada sistem.

3. *Activity Diagram*

Activity diagram merupakan bagian penting dari UML yang menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang terbentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti *use case* atau interaksi.

4. *Sequence Diagram*

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan rangkaian pesan yang dikirim antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem dan *message* (pesan) yang diletakkan diantara objek-objek di dalam use case.

SIMBOL	NAMA	KETERANGAN
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Gambar 0.3 Simbol-simbol pada UML

2.2.9 Database Firebase

Firebase adalah *framework* yang berguna untuk membangun aplikasi portabel dan web untuk bisnis yang membutuhkan *real-time database* yang berarti ketika satu pengguna memperbarui catatan dalam *database*, pembaruan harus disampaikan kepada setiap pengguna dengan segera. Ini memberikan *platform* dasar dan terpadu untuk banyak aplikasi bersama dengan sejumlah fitur *Google* lain

yang disertakan layanan [16]. Firebase menangani sebagian besar pekerjaan sisi server ketika melibatkan pengembangan aplikasi. Ada banyak elemen yang menjadikan Firebase alat yang sangat penting dalam pengembangan dari sudut pandang pengembang. Dengan cara ini, itu membantu menjaga kondisi keselarasan antara pengembang dan klien dengan menyebabkan penundaan kerja minimal. Dalam merancang suatu aplikasi penghitung kecepatan pukulan dan latihan fisik lari terdapat beberapa komponen atau layanan yang ditawarkan oleh Firebase yaitu:

- **Real-time Database:** *Real-time Database* adalah database yang di *hosting* ke *cloud*. Data disimpan sebagai JSON (*JavaScript Object Notation*) dan disinkronkan terus menerus untuk setiap klien terkait. Ketika aplikasi lintas platform dikembangkan dengan iOS, Android, dan JavaScript SDK, bagian terbesar dari permintaan pengguna didasarkan pada satu *instance database Real-time* dan *instance* ini diperbarui dengan setiap data baru.
- Penyimpanan: *Firestore* dirancang untuk mengembangkan aplikasi yang perlu menyimpan dan menyediakan konten buatan pengguna, untuk contoh foto atau *file* lainnya. Ini memberikan pemindahan dokumen yang aman dan unduh untuk aplikasi *Firestore*, terlepas dari kualitas jaringan. *Firestore* didukung oleh *Google Cloud Storage*, layanan penyimpanan objek yang mampu, *basic*, dan hemat biaya.

Pada penelitian ini database firebase digunakan sebagai penyimpanan data yang dikirim oleh alat maupun program.

2.2.10 Progressive Web Application (PWA)

Aplikasi Web Progresif pertama kali diperkenalkan pada tahun 2015 dan merupakan situs web yang dioptimalkan untuk berfungsi pada perangkat seluler. PWA tersedia untuk semua orang, apapun perangkatnya, dapat dipasang di layar beranda pengguna seperti aplikasi aregular dan kontennya selalu ditampilkan dengan benar. Mereka dilayani melalui koneksi HTTPS dan tidak bergantung pada konektivitas Internet, karena pekerja layanan akan menangani perilaku aplikasi saat *offline*. Pekerja layanan juga akan memastikan aplikasi selalu diperbarui.

Maka, Aplikasi Web Progresif ini adalah teknologi yang dirancang untuk mengatasi keterbatasan *browser* seluler dan aplikasi *native*. PWA menggunakan kemampuan *web* modern yang menggambarkan kumpulan teknologi, konsep desain, dan API Web yang bekerja bersama untuk menghadirkan pengalaman pengguna seperti aplikasi *native*. PWA dapat dijalankan dengan mengklik ikon di layar utama ponsel sama seperti bagaimana seseorang menggunakan aplikasi *native*. PWA dapat langsung dijalankan walaupun tidak ada jaringan internet serta mendukung *splash screen* dan *push notification* [17]. Pada penelitian ini PWA digunakan sebagai tampilan program yang dibangun.

2.2.11 JavaScript

JavaScript adalah bahasa berorientasi objek dengan sintaks yang mirip Java, tetapi tidak seperti Java, ia menggunakan sistem objek berbasis prototipe. Objek adalah seperangkat properti, peta yang bisa merubah dari string ke nilai. Properti yang mengevaluasi penutupan dan dipanggil menggunakan konteks objek induknya memainkan peran metode di Java. Setiap objek memiliki bidang prototipe yang mengacu pada objek lain. Pencarian properti melibatkan pencarian objek saat ini, lalu induknya, dan induknya hingga properti ditemukan. Sistem objek JavaScript sangat fleksibel. Sebagai akibatnya, sulit untuk membatasi perilaku dari objek yang diberikan. Sebagai contoh, adalah mungkin untuk mengubah isi dari prototipe apa saja kapan saja atau untuk mengganti bidang prototipe sekaligus [18].

Dalam JavaScript, fungsi apa pun bisa menjadi konstruktor untuk "kelas" objek, dan berisi bidang prototipe, awalnya merujuk objek kosong. Kata kunci baru membuat objek berdasarkan bidang prototipe dan menggunakan fungsi sebagai konstruktor. Semantik yang baru itu sederhana, tetapi tidak biasa: pertama, sebuah objek kosong dibuat, dengan induknya disetel ke objek yang dirujuk oleh bidang prototipe fungsi konstruksinya; kedua, konstruktor disebut, dengan ini terikat pada objek baru. Objek yang dirujuk oleh kata kunci ini tidak ditentukan oleh pelingkupan leksikal, tetapi oleh penelepon; akhirnya nilai dari konstruktor (jika ada) dibuang, dan ekspresi baru mengevaluasi hal ini. Pada penelitian ini Bahasa

pemrograman javascript digunakan sebagai Bahasa pemrograman dalam membangun program yang dibangun.

2.2.12 *Raspberry Pi*

Raspberry Pi adalah sebuah *single board computer (mini PC)* pertama kali diluncurkan pada tahun 2012. *Raspberry Pi* dibuat/dikembangkan oleh sebuah perusahaan yang bernama *Raspberry Pi Foundation* dari UK. Perusahaan ini membuat *Raspberry Pi* bertujuan untuk mempromosikan pengajaran ilmu komputer di sekolah dasar. *Raspberry Pi* menggunakan *system on a chip (SoC)* dari Broadcom BCM2837 dan tidak menggunakan *hard disk*, namun menggunakan *SD Card* untuk proses *booting* dan penyimpanan data jangka panjang [12]. Pada penelitian ini perangkat raspberry digunakan sebagai server atau transfer data. Gambar *Raspberry Pi* dapat dilihat pada gambar 2.3.



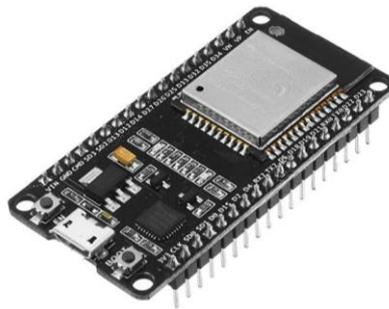
Gambar 0.4 *Raspberry Pi*

2.2.13 Mikrokontroler

Mikrokontroler merupakan suatu *Central Processing Unit (CPU)* yang disertai dengan memori serta terdapat sarana *input* dan *output* yang dibuat dalam bentuk chip IC (*Integrated Circuit*). CPU ini terdiri dari dua bagian yaitu yang pertama adalah unit pengendali yang berfungsi untuk mengambil instruksi-instruksi yang tersimpan dalam memori, member kode instruksi-instruksi tersebut dan melaksanakannya dan yang kedua adalah unit aritmatika dan logika yang berfungsi untuk melakukan proses-proses perhitungan yang diperlukan selama suatu program dijalankan [13].

Mikrokontroler lebih dari sekedar sebuah mikroprosesor karena sudah terdapat ROM (*Read Only Memory*), RAM (*Read Access Memory*), beberapa port masukan maupun keluaran, dan beberapa peripheral seperti pencacah / pewaktu, ADC (*Analog to Digital Converter*), DAC (*Digital to Analog Converter*) dan serial komunikasi. Salah satu mikrokontroler yang digunakan untuk penelitian ini yaitu mikrokontroler ESP32.

ESP32 adalah mikrokontroler dengan sistem biaya rendah, daya rendah pada mikrokontroler chip dengan *Wi-Fi* terintegrasi dan *Bluetooth* mode ganda. Seri ESP32 menggunakan mikroprosesor *Tensilica Xtensa LX6* baik dalam variasi *dual-core* dan *single-core* dan termasuk *switch* antena *built-in*, RF balun, penguat daya, penguat penerima derau rendah, filter, dan modul manajemen daya. ESP32 dibuat dan dikembangkan oleh *Espressif Systems*, sebuah perusahaan Cina yang berbasis di Shanghai, dan diproduksi oleh TSMC menggunakan proses 40 nm mereka. Mikrokontroler ESP32 dapat dilihat pada gambar 2.4.



Gambar 0.5 Mikrokontroler ESP32

2.2.14 Sensor Proximity

Sensor Proximity merupakan sebuah sensor yang digunakan untuk mendeteksi apakah terdapat suatu objek atau tidak, juga untuk mengetahui jarak suatu benda atau objek terhadap sensor. Selain itu sensor proximity juga dapat membedakan 2 buah garis antara garis berwarna hitam dengan garis berwarna putih, seperti sensor yang digunakan pada rangkaian robotika khususnya rangkaian *line tracer robot / line follower robot* [14]. Sensor proximity ini dapat dikalibrasi untuk menyesuaikan pembacaan sensor terhadap objek, Sehingga pembacaan sensor akan

lebih akurat. Dalam penelitian ini menggunakan sensor proximity induktif dimana sensor ini berfungsi untuk mengukur kecepatan tanpa melakukan kontak langsung dengan benda atau objek. Pada penelitian ini sensor proximity digunakan sebagai sensor yang mendeteksi objek pukulan atlet. Sensor Proximity dapat dilihat pada gambar 2.5.

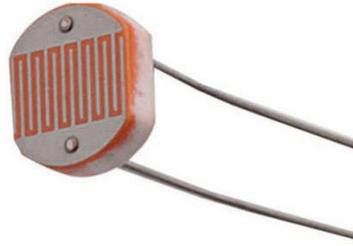


Gambar 0.6 **Sensor Proximity**

2.2.15 Sensor LDR (*Light Dependent Resistor*)

Sensor LDR merupakan sensor yang dapat mendeteksi cahaya. Sensor ini biasa disebut sensor cahaya karena dari singkatan nya pun diartikan yang tergantung pada cahaya. Sensor ini digunakan untuk merubah energi cahaya menjadi cahaya listrik. Sensor LDR merupakan sebuah sensor jenis semikonduktor yang dibuat dari cadmium selenoide dan timah sulfida. Sensor yang terdiri dari sebuah piringan bahan semilkonduktor dengan dua buah elektroda pada permukaannya. LDR tergantung pada cahaya, artinya nilai tahanannya akan berubah-ubah apabila terkena cahaya yang diterima.

Prinsip kerja sensor ini yaitu jika sinar atau cahaya mengenai permukaan yang konduif dari LDR, maka tahanannya menjadi lebih kecil dan arusnya menjadi lebih besar sedangkan bila tidak ada sinar yang mengenai permukaan maka nilai tahanannya akan menjadi besar tergantung dari intensitas cahaya yang masuk pada permukaan konduif dari LDR. Pada penelitian ini perangkat sensor LDR digunakan sebagai penangkap cahaya dari laser dan mendeteksi jika atlet melewati laser tersebut. Sensor LDR dapat dilihat pada gambar 2.6.



Gambar 0.7 Sensor LDR

2.2.16 Laser

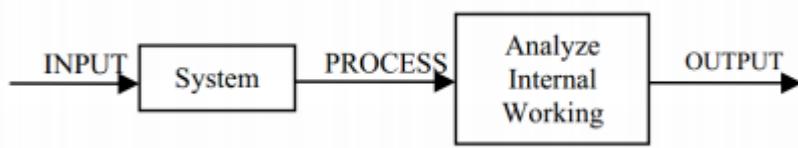
Laser atau singkatan dari *Light Amplification by Stimulated Emission of Radiation* merupakan mekanisme suatu alat yang memancarkan radiasi elektromagnetik melalui proses pancaran terstimulasi. Salah satu keistimewaan dari laser yaitu sifatnya yang tidak menyebar sehingga memiliki fokus yang lebih baik dari sumber cahaya lain. Pada penelitian ini perangkat laser digunakan sebagai pemancar cahaya pada sensor LDR.

2.2.17 Metode Pengujian

Pengujian merupakan hal yang sangat diperlukan dalam sebuah sistem, karena menjadi salah satu cara atau proses eksekusi program untuk menemukan kesalahan pada sistem yang dibuat. Berikut adalah 2 jenis dari metode pengujian yang dipakai untuk menguji perangkat lunak yang dibuat yaitu *white box testing* dan *black box testing* [19].

2.2.17.1 White Box Testing

White Box Testing merupakan pengujian untuk memperlihatkan cara kerja dari produk secara rinci sesuai dengan spesifikasinya. Metode pengujian ini dilakukan dengan menggunakan struktur kontrol program untuk memperoleh kasus uji. Pengujian ini didasarkan pada detail prosedur dan alur logika kode program. Pada kegiatan *white box testing*, tester melihat *source code* program dan menemukan *bugs* dari kode program yang diuji. Intinya *white box testing* adalah pengujian yang dilakukan sampai kepada detail pengecekan kode program.



Gambar 0.8 **Teknik Pengujian White Box**

Berikut adalah kasus yang dapat ditemukan menggunakan pengujian ini:

- Menjamin seluruh jalur independen di dalam modul yang dieksekusi sekurang-kurangnya sekali .
- Menguji semua keputusan logikal.
- Menguji seluruh Loop yang sesuai dengan batasannya.
- Menguji seluruh struktur data internal yang menjamin validitas.
- Basis Path adalah teknik uji coba white box (Tom Mc Cabe). Basis Path digunakan untuk mendapatkan kompleksitas logik dari suatu prosedur dan menggunakan ukuran ini sebagai petunjuk untuk mendefinisikan himpunan jalur yang akan diuji. Basis Path menggunakan notasi graph untuk menggambarkan aliran kontrolnya.

2.2.17.2 Black Box Testing

Black box testing adalah pengujian yang dilakukan hanya mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak, *black box testing* juga merupakan metode pengujian perangkat lunak yang menguji fungsionalitas aplikasi yang bertentangan dengan struktur internal atau kerja (lihat pengujian white-box). Pengetahuan khusus dari kode aplikasi / struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan.



Gambar 0.9 **Teknik Pengujian Black Box**

Berikut adalah kesalahan yang dapat ditemukan menggunakan pengujian ini:

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan interface
3. Kesalahan dalam struktur data atau akses basisdata eksternal
4. Inisialisasi dan kesalahan terminasi
5. validitas fungsional
6. kesensitifan sistem terhadap nilai input tertentu
7. batasan dari suatu data

Ada beberapa tipe dari *Black Box Testing* yang akan dijabarkan dibawah yaitu sebagai berikut:

1. Equivalence class partitioning
 - a. Bagi domain Input ke dalam beberapa kelas yang nantinya akan dijadikan sebagai kasus uji
 - b. kelas yang telah terbentuk disajikan sebagai kondisi input dalam kasus uji
 - c. Kelas tersebut merupakan himpunan nilai-nilai yang valid dan tidak valid
 - d. kondisi input bisa merupakan suatu range, harga khusus, suatu himpunan, atau suatu boolean
 - e. Bila kondisi input berupa suatu range, maka input kasus ujinya satu valid dan dua yang invalid
 - f. Bila kondisi input berupa suatu harga khusus, maka input kasus ujinya satu valid dan dua yang invalid
 - g. Bila kondisi input berupa suatu anggota himpunan, maka input kasus ujinya satu valid dan dua yang invalid.
 - h. Bila kondisi input berupa suatu anggota Boolean, maka input kasus ujinya satu valid dan dua yang invalid

2. Sample testing

- a. Melibatkan sejumlah nilai yang dipilih dari data masukan kelas ekivalensi
- b. Integrasikan nilai tersebut ke dalam kasus uji
- c. Nilai yang dipilih dapat berupa konstanta atau variabel Limit Testing
- d. Kasus uji yang memproses nilai batas (atau titik singular)
- e. Nilai batas disimpulkan dari kelas ekivalensi dengan mengambil nilai yang sama atau mendekati nilai yang membatasi kelas ekivalensi tersebut
- f. Limit test also juga melibatkan data keluaran dari ekivalensi kelas
- g. Pada kasus segi tiga, misalnya limit testing mencoba untuk mendeteksi apakah $a+b \geq c$ dan bukan $a + b > c$
- h. Bila kondisi input menentukan suatu range, maka kasus ujinya harus mencakup pengujian nilai batas dari range dan nilai invalid yang dekat dengan nilai batas. Misal bila rangenya antara $[-1.0, +1.0]$, maka input untuk kasus ujinya adalah $-1.0, 1.0, -1.001, 1.001$
- i. Bila kondisi inputnya berupa harga khusus kasus ujinya harus mencakup nilai minimum dan maksimum. Misal suatu file dapat terdiri dari 1 to 255 record, maka kasus ujinya harus mencakup untuk nilai 0, 1, 255 and 256, atau uji saat keadaan record kosong dan record penuh.

3. Robustness testing

Data dipilih dari luar range yang didefinisikan. Tujuan pengujian ini adalah untuk membuktikan tidak adanya kejadian yang katastrofik yang dihasilkan akibat adanya keabnormalan.

4. Behavior testing

Suatu pengujian yang hasilnya hanya dapat dievaluasi per sub program, tidak bisa dilakukan per modul

5. Requirement testing

- a. Menyusun kasus uji untuk tiap kebutuhan yang berkorelasi dengan modul / CSU
- b. Tiap kasus uji harus dapat dirunut dengan kebutuhan perangkat lunaknya melalui matriks keterunutan.

