

2. DASAR TEORI

2.1 Pedahuluan

Pada Bab 2 akan dibahas mengenai dasar teori mengenai optimasi penggunaan *piggyback* pada kendaraan roda dua. Metode yang digunakan dalam optimalisasi parameter *piggyback* menggunakan *Artificial Neural Network* (NN). Dalam menggunakan NN akan dilakukan pendekatan dengan metode *Genetic Algorithm* (GA), *Particle Swarm Optimization* (PSO), dan *Firefly Algorithm* (FA). Selain itu, akan dijelaskan bagaimana metode tersebut dapat membantu untuk meningkatkan daya dan torsi pada kendaraan Yamaha Aerox 155. Pengujian untuk mengetahui seberapa jauh peningkatan daya dan torsi mesin akan dilakukan menggunakan *dyno test*.

2.2 Kajian Terdahulu

2.2.1 *Velocity Stack*

Jurnal yang ditulis oleh D. Wentworth-Linton dan S. Gao pada tahun 2017 dengan judul "A Computational Study of the Effect of Intake Design on Volumetric Efficiency for Best Performance in Motorsport" membahas mengenai variasi panjang *stack* untuk mengoptimalkan daya dan torsi. Dalam penelitian ini, penulis menggunakan beberapa variasi panjang *stack*, antara lain: 40mm, 70mm, 90mm, dan 120mm. Menurut penelitian yang dilakukan, hasil peningkatan daya dan torsi yang paling optimal terdapat pada panjang *stack* 70mm, dimana dengan panjang tersebut, hasil pengujian yang didapat lebih maksimal.

Jurnal yang ditulis oleh Santoso dan R.A. Prayitno pada tahun 2022 dengan judul "ANALYSIS OF THE EFFECT OF INTAKE MANIFOLD SHAPE ON MOTORCYCLE ENGINE PERFORMANCE" membahas mengenai panjang dari *intake manifold* pada performa mesin. Dalam penelitian ini, penulis menggunakan variasi panjang pada *intake manifold* antara lain: 1cm, 2cm, dan 3cm. Untuk peningkatan daya, panjang yang paling optimal adalah 2cm, sedangkan peningkatan torsi paling maksimal didapatkan pada panjang 3cm.

Jurnal yang ditulis oleh N.A. Wahab, N. Mohd, N. Nasir, M. A. A. Halim, S. B. Mat, dan W.Z. W. Omar pada tahun 2020 dengan judul "The Effect of Velocity Stack Inlet Diameter to the Air Intake Volume Efficiency" membahas mengenai pengaruh lebar lubang *inlet* dari filter udara menuju ke *intake manifold* pada performa mesin. Dalam penelitian ini, penulis menggunakan variasi lebar lubang *inlet*. Hasil yang didapatkan adalah, adanya peningkatan efisiensi volumetrik serendah 1.34% hingga 2.47% untuk konfigurasi dimodifikasi.

Jurnal yang ditulis oleh R. Rawal dan S. M. Thakkar pada tahun 2018 dengan judul "An experimental investigation of inlet manifold modified by CFD Analysis on performance and emission characteristics of direct injection CI engine", membahas mengenai pengaruh perbedaan antara lubang inlet normal dan berulir yang meunuju ke filter udara pada performa mesin. Tujuan penelitian ini adalah untuk melihat apakah ada peningkatan kinerja mesin atau tidak.

Jurnal yang ditulis oleh B. Wahono, A. Setiawan, dan O. Lim pada tahun 2020 dengan judul "Study on the Effect of Intake Flow by Various Intake Port Design on Small Motorcycles Engine", membahas mengenai pengaruh perbedaan antara lubang port intake tangensial dan lubang port helical. Hasil analisis dalam penelitian ini memiliki manfaat dalam memperkirakan aliran dan meningkatkan optimalisasi ruang pembakaran mesin.

Jurnal yang ditulis oleh M. Pieter dan T. Sutrisno pada tahun 2016 dengan judul "PENGARUH VELOCITY STACK INTAKE TERHADAP PERFORMA KENDARAAN", membahas mengenai pengaruh pemasangan velocity stack intake pada peningkatan performa kendaraan dengan melakukan perubahan radius inlet velocity stack intake dengan variasi inlet sebesar 15mm dan 30mm. Hasil dari penelitian yang dilakukan adalah, inlet yang lebih besar menghasilkan peningkatan daya sebesar 3.4% dan kenaikan torsi sebesar 6.6%.

Jurnal yang ditulis oleh G.A. Saputra dan S.Putro pada tahun 2024 dengan judul "Study Experimental of Honda Astrea Grand Performance Modification Intake Manifold Using Velocity Stack Dimple 1mm with Variation Diameter Inlate 60, 70, and 80mm", membahas mengenai pengaruh *dimple* dengan variasi lebar *inlet diameter* 60mm, 70mm, dan 80mm. Hasil penelitian yang telah dilakukan adalah adanya peningkatan daya sebesar 0.7HP dan peningkatan torsi sebesar 0.40Nm.

2.2.2 Piggyback

Jurnal yang ditulis oleh A. N. Faizin, S. Mahendra, dan T. Setiawan pada tahun 2021 dengan judul "PENGARUH PENGGUNAAN FUEL ADJUSTER TERHADAP PERFORMA MESIN SEPEDA MOTOR 4 TAK 110 CC" membahas mengenai penggunaan *piggyback* yang digunakan untuk mengoptimalkan daya dan torsi pada mesin Honda Beat. Dalam penelitian ini, penulis menggunakan *piggyback* agar dapat membuat campuran antara bahan bakar lebih banyak. Hasil yang di dapat ialah peningkatan torsi sebesar 3.33 N.m dan daya sebesar 0.512 HP. Penelitian tersebut dilakukan menggunakan bahan bakar Pertamina. Tinjauan ini menunjukkan bahwa penggunaan *piggyback* akan memengaruhi kinerja dari mesin kendaraan untuk menjadi optimal.

Jurnal yang ditulis oleh A.M. Huda, N. Apriyanto, dan S. Mahendra pada tahun 2022 dengan judul "PENGARUH PENGGUNAAN FUEL ADJUSTER IQUTECHÉ TERHADAP PERFORMA DAN EMISI GAS BUANG SEPEDA MOTOR HONDA VARIO 150 CC TAHUN 2019" membahas mengenai penggunaan *piggyback fuel adjuster* Iquteche terhadap performa dan gas buang pada Honda Vario. Dalam melakukan penelitian ini, penulis menggunakan *piggyback fuel adjuster* Iquteche, yang dimana pada instalasinya hanya *plug and play* atau dapat langsung digunakan tanpa harus di program. Dalam penelitian yang dilakukan, terlihat bahwa penggunaan *piggyback fuel adjuster* Iquteche dapat meningkatkan daya sebesar 30% dan torsi sebesar 37%. Dari penelitian yang telah dilakukan, ditemukan bahwa kelebihan menggunakan *piggyback fuel adjuster* Iquteche ini adalah tenaga dan torsi yang meningkat. Namun perlu diperhatikan bahwa peningkatan tenaga dan torsi yang signifikan tanpa adanya pengaturan campuran yang pas juga mempengaruhi banyaknya bahan bakar yang terpakai.

Jurnal yang ditulis oleh D. Hermawan, A. Nasution, dan I. Hermawan pada tahun 2020 dengan judul "ANALYSIS OF THE EFFECT OF ADDING ECO RACING ON PERTAMAX FUEL ON THE PERFORMANCE OF 2 STOCK MOTOR ENGINES" membahas mengenai pemakaian bahan bakar Pertamina 92 terhadap daya yang lebih meningkat pada sepeda motor. Dalam melakukan penelitian, penulis menggunakan sepeda motor yang telah diisi dengan bahan bakar Pertamina 92 dan melakukan pengujian pada rentang 4000 – 4200 RPM dan 5400 – 5600 RPM. Dari kedua pengujian tersebut, terbukti bahwa penggunaan bahan bakar Pertamina dapat meningkatkan daya hingga 9kW.

2.2.3 Artificial Intelligence (AI) Method

Jurnal yang ditulis oleh A. Saleh dan L. Gáspár pada tahun 2023 dengan judul "Optimizing asphalt foaming using neural network", membahas mengenai pengaruh penggunaan Neural Network (NN) yang dipadukan dengan metode Particle Swarm Optimization (PSO) pada kontrol pembusaan aspal, dimana NN digunakan sebagai pemetaan sebelum masuk kedalam metode PSO. Hasil yang didapatkan pada penelitian ini dapat dimanfaatkan untuk mengoptimalkan desain campuran dan meningkatkan pemahaman tentang faktor-faktor yang berpengaruh dalam aplikasi aspal berbusa, sehingga menghasilkan peningkatan kinerja dan efisiensi di bidang ini.

Jurnal yang ditulis oleh T. He dan K. Zhao pada tahun 2018 dengan judul "Multispectral Remote Sensing Land Use Classification Based on RBF Neural Network with Parameters Optimized by Genetic Algorithm", membahas mengenai pengaruh penggunaan metode Genetic Algorithm pada klasifikasi gambar multispektral. Hasil eksperimen menunjukkan

bahwa akurasi klasifikasi keseluruhan algoritma ini meningkat sebesar 11% dibandingkan menggunakan algoritma lain, yang dimana akan meningkatkan akurasi klasifikasi gambar multispektral secara efektif.

Jurnal yang ditulis oleh F.B Faruque, S. Chowdhury, M. S. Nazim, M. Sowmitra, A. K. M. A. M. Sowmitra pada tahun 2021 dengan judul "Optimized Distributed Generation Planning for Radial Distribution System Using Particle Swarm Optimization Algorithm", membahas mengenai pengaruh penggunaan metode Particle Swarm Optimization (PSO) untuk menentukan posisi terbaik dari Unit Distributed Generation (DG). Hasil eksperimen menunjukkan pengurangan maksimal dari kehilangan daya yang sebenarnya terlihat dalam hasil, sementara lokasi optimal untuk Pembangkitan Terdistribusi juga teridentifikasi.

Jurnal yang ditulis oleh S. Ranjith dan P. J. Jayarin pada tahun 2022 dengan judul "Improved Localization Algorithm Using Hybrid Firefly Genetic Algorithm in Wireless Sensor Network", membahas mengenai pengaruh penggunaan metode Firefly Algorithm (FA) untuk mencari jumlah total node sensor, posisi node jangkak atau node landmark yang diketahui. Hasil penelitian menunjukkan bahwa Algoritma Hybrid Genetic Algorithm-Firefly Localization, setelah menjalani 100 iterasi, memiliki tingkat akurasi estimasi posisi node sensor sebesar 100% untuk vektor populasi kunang-kunang 15, 18, dan 20, dengan tingkat kompleksitas waktu yang lebih efisien

Jurnal yang ditulis oleh N. Chityal dan S. Sapkal pada tahun 2022 dengan judul "Performance analysis of GA, PSO and JA for determining the optimal parameters in friction drilling process", membahas mengenai penggunaan metode *Genetic Algorithm (GA)*, *Particle Swarm Optimization (PSO)*, *Jaya Algorithm (JA)* pada proses meminimalkan kekasaran permukaan lubang bor gesekan yang bergantung pada parameter yaitu kecepatan putaran, sudut gesekan dan ketebalan benda kerja. Untuk hasil yang didapat, PSO memiliki kinerja yang lebih baik daripada metode lainnya.

Jurnal yang ditulis oleh A. T. Hammid dan M. H. Bin Sulaiman pada tahun 2018 dengan judul "Series division method based on PSO and FA to optimize Long-Term Hydro Generation Scheduling", membahas mengenai penggunaan metode Particle Swarm Optimization (PSO) dan Firefly Algorithm (FA) pada proses menganalisis debit air sehari-hari, mempertimbangkan batasan ketinggian reservoir, serta mengoptimalkan pembangkitan listrik yang bergantung pada unit head pembangkit listrik tenaga air yang dipengaruhi oleh fluktuasi daya, proses start-up, dan penghentian unit pembangkit listrik tenaga air. Untuk hasil yang didapat, metode FA memiliki hasil yang kuat dan memiliki efisiensi serta keunggulan yang baik.

2.2.4 Kesimpulan

Dari jurnal yang telah dijadikan sebagai referensi, didapatkan kesimpulan bahwa pada masing-masing referensi mengenai *velocity stack* tidak melakukan banyak perubahan terhadap variasi desain *velocity stack*. Sehingga pada penelitian yang akan dilakukan variasi yang diubah meliputi bagian *inlet diameter*, *outlet height*, dan *lip radius*. Kemudian pada sektor *piggyback*, tidak dicantumkan apakah dengan setelan konfigurasi minus akan meningkatkan daya dan torsi sebanyak berapa persen. Sehingga pada *piggyback* yang akan digunakan, akan dilakukan penyetelan dengan angka minus. Kemudian untuk jenis *piggyback* yang akan digunakan adalah *Piggyback Fuel Adjuster (FADJ)* Iquteche.

Kombinasi antara desain *velocity stack* dan setelan *piggyback* akan dioptimalisasi menggunakan metode *Genetic Algorithm (GA)*, *Particle Swarm Optimization (PSO)*, dan *Firefly Algorithm (FA)*. Jika melihat pada beberapa sumber referensi, dikatakan bahwa metode yang paling optimal adalah PSO dan FA. Namun pada pengujian yang akan dilakukan akan ditinjau ulang mengenai metode mana yang dapat menghasilkan kombinasi paling optimal agar tercapai peningkatan daya dan torsi.

2.3 Dasar Teori

Pada dasar teori akan dibahas mengenai optimasi desain *velocity stack* pada performa mesin Yamaha Aerox 155 dengan *piggyback* menggunakan bantuan teknologi *Artificial Intelligence (AI)*. Dalam melakukan penelitian ini, metode AI yang akan digunakan antara lain: *Genetic Algorithm (GA)*, *Particle Swarm Optimization (PSO)*, dan *Firefly Algorithm (FA)*. Pengetahuan dalam dasar teori ini akan membantu dalam melaksanakan topik penelitian agar di dapat solusi terbaik untuk meningkatkan daya dan torsi pada Yamaha Aerox 155.

2.3.1 Velocity Stack

Velocity stack merupakan komponen sistem pemasukan aliran ke mesin yang bertujuan agar aliran udara ke *throttle body* lebih optimal. Dengan lebih banyaknya volume aliran udara yang masuk ke dalam *intake*, maka pembakaran bahan bakar akan lebih mudah terjadi. Kemudian dengan semakin banyaknya bahan bakar yang dibakar, maka semakin besar nilai daya yang dapat diperoleh. Masuknya udara yang lancar, tanpa hambatan, dan konsisten juga memastikan mesin berfungsi pada efisiensi puncak bahkan pada kecepatan tinggi ketika banyak tekanan yang turun pada mesin. Hal ini terjadi karena rentang RPM diatur dengan baik pada semua kecepatan, yang dimana hal ini baik untuk mengoptimalkan daya dan dapat mengurangi panas berlebih. Secara keseluruhan, aliran udara yang baik, keberhasilan pembakaran bahan bakar yang sempurna, dan timing katup yang tepat memungkinkan mesin

menghasilkan daya yang lebih besar. Dapat dilihat contoh bentuk *velocity stack* pada Gambar 2.1



Gambar 2. 1 *Velocity Stack*

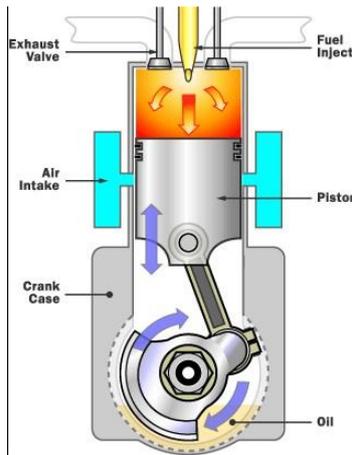
Sumber: Ratmotosport (2017). <https://ratmotosport.wordpress.com/2017/01/04/velocity-stack-dan-efeknya-terhadap-tenaga/>

2.3.2 Internal Combustion Engine (ICE)

Internal Combustion Engine (ICE) adalah jenis mesin pembakaran dalam yang menggunakan pembakaran bahan bakar di dalam ruang bakar untuk menghasilkan tenaga mekanik (Ferguson & Kirkpatrick, 2016, p1-4). Mesin pembakaran dalam ini sangat umum digunakan dalam kendaraan bermotor seperti mobil, truk, motor, dan juga digunakan dalam mesin-mesin industri dan pembangkit listrik.

Prinsip kerja ICE melibatkan campuran bahan bakar seperti bahan bakar atau gas dengan udara di dalam ruang bakar mesin. Proses ini menghasilkan reaksi pembakaran yang menghasilkan gas panas dan tekanan pada piston dalam silinder. Gerakan piston menggerakkan poros engkol, yang kemudian mentransfer tenaga mekanik ke komponen lain, seperti roda kendaraan melalui transmisi. Pada Gambar 2.2 dapat dilihat untuk proses yang terjadi dalam ICE.

Ada beberapa jenis mesin pembakaran dalam yang umum digunakan, seperti mesin bensin (*spark-ignition engine*) dan mesin diesel (*compression-ignition engine*). Mesin bensin menggunakan busi (*spark plug*) untuk membakar campuran udara-bahan bakar yang terkompresi, sedangkan mesin diesel menggunakan kompresi tinggi untuk memicu pembakaran bahan bakar yang terkompresi di dalam silinder (Ferguson & Kirkpatrick, 2016 p198-220).



Gambar 2.2 *Internal Combustion Engine*

Sumber: Mectips (2018). <https://mectips.com/internal-combustion-engines-advantages-disadvantages-applications/>

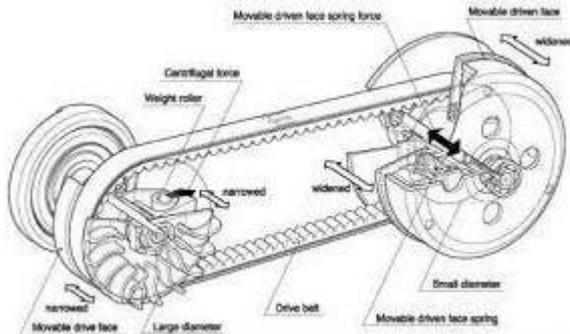
2.3.2.1 *Pertamina Pertamina 92*

Bahan bakar Pertamina yang diproduksi oleh Pertamina, menonjolkan angka oktan minimal 92. Hal ini memastikan pembakaran yang optimal dan bebas residu, sehingga direkomendasikan khusus untuk penggunaan kendaraan sehari-hari. Keunggulan bahan bakar dengan angka oktan yang tinggi tidak hanya meningkatkan efisiensi pembakaran, tetapi juga menghasilkan residu yang minimal sehingga akan menjaga mesin tetap bersih dari penumpukan karbon yang mengganggu kinerja mesin. Selain itu, Pertamina menawarkan manfaat tambahan melalui formula PERTATEC (*Pertamina Technology*), yaitu campuran aditif yang mampu membersihkan endapan kotoran pada mesin, meningkatkan umur mesin, melindungi dari karat, dan membuat penggunaan bahan bakar yang lebih efisien (MyPertamina, 2024).

2.3.3 *Continous Variable Transmission (CVT)*

Transmisi adalah suatu komponen yang berfungsi memindahkan tenaga dari mesin ke roda dengan pengaturan torsi atau momentum yang sesuai. Pada sepeda motor jenis skuter matik, umumnya menggunakan transmisi berjenis CVT (*Countinously Variable Transmission*). Transmisi ini bekerja dengan menggunakan dua buah roda gigi yang memiliki diameter yang bervariasi. Prinsip kerja transmisi CVT adalah dengan menggunakan dua buah roda gigi yang disatukan dengan sebuah belt. Pada CVT, ada roda gigi yang bertindak sebagai gigi pemutar (*driver gear*), juga ada yang bertindak sebagai gigi yang diputar (*driven gear*), dan juga ada V belt yang berguna sebagai sabuk penghubung antara *driver gear* dan *driven gear*. Ketika mesin mati, maka diameter *drive gear* mengecil dan diameter *driven gear* membesar. Sehingga ketika mesin hidup, motor bisa langsung berakselerasi karena perbandingan gigi besar. Namun ketika

RPM mesin naik, *drive gear* akan membesar dan *driven gear* otomatis mengecil sehingga perbandingan gigi semakin berkurang (Herimarianto, 2017). Pada gambar 2.3 dapat dilihat bagian-bagian dalam CVT.



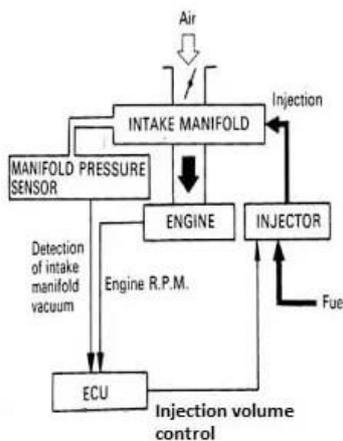
Gambar 2.3 *Countinously Variable Transmission (CVT)*

Sumber: Herimarianto (2018). <https://medium.com/@Marianto454/prinsip-cara-kerja-cvt-pada-motor-matic-349e42f846ef>

2.3.4 **Electronic Fuel Injection (EFI)**

Electronic Fuel Injection (EFI) adalah sistem injeksi bahan bakar elektronik yang digunakan dalam kendaraan bermotor untuk mengatur dan mengirimkan bahan bakar ke mesin secara presisi. Sistem ini menggantikan karburator yang digunakan pada sistem pembakaran tradisional. Dalam EFI, bahan bakar akan disemprotkan ke dalam *intake* atau langsung ke ruang bakar mesin dengan menggunakan injektor. Sistem ini dikontrol oleh *Electronic Control Unit* (ECU) yang menerima data dari berbagai sensor yang terhubung ke mesin, seperti sensor suhu, sensor tekanan udara, sensor *throttle position*, sensor O_2 , dan sensor-sensor lainnya (Suzuki Indonesia, 2022). Pada Gambar 2.4 dapat dilihat siklus kerja dari sistem EFI.

ECU mengambil data dari sensor-sensor tersebut dan menggunakan algoritma pemrograman yang kompleks untuk mengatur waktu dan jumlah bahan bakar yang disemprotkan oleh injektor. Ini memungkinkan pengaturan yang akurat dan optimal sesuai dengan kondisi kerja mesin, seperti kecepatan mesin, beban mesin, suhu udara, dan kebutuhan bahan bakar lainnya.



Gambar 2.4 Sistem EFI

Sumber: Rohman, A (2011). <https://teknisimobil.com/dasar-otomotif/4-jenis-sistem-efi-berdasarkan-teknologi-kontrol-15075/>

2.3.5 *Electronic Control Unit (ECU)*

Electronic Control Unit (ECU) pada sepeda motor merupakan sistem komputer yang bertanggung jawab mengendalikan berbagai aspek elektronik pada motor. ECU berperan penting dalam mengatur dan mengoptimalkan kinerja mesin serta sistem lainnya pada sepeda motor. ECU pada sepeda motor terdiri dari berbagai modul elektronik yang bekerja secara terintegrasi untuk mengatur fungsi elektrik pada sepeda motor. Beberapa fungsi yang dikendalikan oleh ECU, seperti mengendalikan banyaknya bahan bakar yang masuk ke dalam ruang bakar, timing pengapian, sistem pembakaran, dan sistem keamanan.

ECU menggunakan berbagai sensor yang tersebar di sepeda motor untuk mengumpulkan data dan mengirimkan instruksi kepada komponen-komponen elektronik lainnya. Dengan informasi yang dikumpulkan dan analisis yang dilakukan, ECU memastikan bahwa sepeda motor beroperasi secara optimal, efisien, dan aman dalam berbagai situasi. Contoh gambar ECU dapat dilihat pada Gambar 2.5.



Gambar 2.5 *Electronic Control Unit (ECU)*

Sumber: Yamaha Indonesia (2024). <https://www.yamaha-motor.co.id/parts-catalogue/>

2.3.5.1 *Standard Electronic Control Unit (ECU)*

Pabrikan kendaraan biasanya memprogram *Electronic Control Unit (ECU)* sesuai dengan standar dan spesifikasi kendaraan mereka. Ini mencakup konfigurasi untuk memenuhi persyaratan emisi, efisiensi bahan bakar, kinerja, dan keselamatan kendaraan. ECU standar bawaan pabrik sering kali memiliki batasan atau pembatasan tertentu untuk memastikan kendaraan tetap beroperasi dalam parameter yang aman dan memenuhi persyaratan regulasi. Untuk ECU standar dari Yamaha Aerox 155 dapat dilihat pada Gambar 2.6.



Gambar 2.6 ECU Standar Aerox 155

Sumber: Yamaha Indonesia (2024). <https://www.yamaha-motor.co.id/parts-catalogue/>

2.3.5.2 *Piggyback Fuel Adjuster (FADJ) Iquteche*

ECU *piggyback* pada kendaraan roda dua adalah jenis ECU tambahan yang dipasang di sepeda motor untuk meningkatkan kinerja mesin atau mengatur sistem lainnya sesuai dengan kebutuhan. ECU *piggyback* bekerja bersama dengan ECU standar yang ada pada sepeda motor dan memberikan kontrol tambahan atau modifikasi pada beberapa parameter operasional. ECU *piggyback* biasanya terhubung ke ECU standar menggunakan *harness* atau kabel-kabel atau konektor yang sesuai dengan spesifikasi ECU standar pada kendaraan. Ini memungkinkan ECU *piggyback* untuk membaca dan memodifikasi data yang dikirimkan oleh sensor-sensor dan mengirimkan sinyal tambahan ke komponen-komponen mesin.

Pada *Piggyback Fuel Adjuster (FADJ) Iquteche*, parameter yang dapat diubah adalah banyaknya semprotan bahan bakar yang masuk ke dalam ruang bakar melalui injektor. Kemudian untuk variabel yang dapat disetel ada pada rentang RPM, dimana ada 3 variabel antara lain *low*, untuk mengatur penyemprotan bahan bakar pada 0 – 3500 RPM, *médium*, untuk mengatur penyemprotan bahan bakar pada 3500 – 7000 RPM, dan *high*, untuk mengatur penyemprotan bahan bakar pada 7000 – ∞ . Ketiga variabel tersebut dapat disetel dari -20% hingga +50%, namun untuk penggunaan mesin standar pabrik tidak disarankan memakai setelan pada ketiga variabel kurang dari -5% (cicakkreatip, 2017). Dalam melakukan

penyetelan *piggyback* harus sesuai dengan spesifikasi kendaraan agar tidak terjadi *overheat*, merusakkan komponen *internal* pada mesin, dan bahan bakar menjadi boros. Pada Gambar 2.7 merupakan *Piggyback Fuel Adjuster* Iquteche yang akan digunakan.



Gambar 2.7 *Piggyback Fuel Adjuster* Iquteche

Sumber: Sooko (2024). <https://www.sooko.io/fadj/>

2.3.6 Yamaha Aerox 155

Yamaha Aerox merupakan salah satu jenis sepeda motor dengan tipe matik. Kendaraan ini pertama kali di produksi oleh Yamaha pada tahun 1997 dan hanya dikhususkan untuk pasar Eropa. Sepeda motor Yamaha Aerox pertama kali diluncurkan dengan mengusung mesin 2-tak dengan kubikasi mesin 50cc dengan transmisi otomatis. Pada generasi pertama tersebut seringkali digunakan sebagai kendaraan *paddock* untuk kebutuhan tim Yamaha pada ajang MotoGP. Generasi kedua dari model Yamaha Aerox masih mengusung mesin 2-tak namun dengan peningkatan kubikasi pada mesin yaitu menjadi 100cc yang dapat mengeluarkan tenaga sebesar 7,5 daya kuda dan torsi sebesar 5,5 Nm. Kemudian pada generasi ketiga, Yamaha Aerox beralih menjadi 4-tak karena mengikuti regulasi yang berlaku. Pada generasi ketiga tersebut, Yamaha Aerox mulai memasuki pasar Indonesia di tahun 2016. Pada generasi ketiga, Aerox cukup bersaing dengan para kompetitor dikelasnya, seperti sudah menggunakan lampu LED, helm yang dapat masuk kedalam bagasi, dan fitur lainnya. Kemudian pada akhir tahun 2016, Yamaha kembali meluncurkan tipe Aerox dengan kubikasi mesin sebesar 155cc dan memiliki teknologi *Variable Valve Actuation*) dan *Blue Core* (Yamaha Jatim, n.d). Pada generasi keempat tersebut dapat dilihat pada Gambar 2.8, dimana desain tersebut sangat menarik peminat para pembeli karena model yang *sporty* serta memiliki tampilan gahar. Kemudian pada generasi kelima, Yamaha kembali meluncurkan Aerox dengan kubikasi mesin yang tetap 155cc namun dengan pembaharuan fitur. Fitur tersebut ialah *Y-Connect* yang

terhubung dengan *Communication Control Unit (CCU)*, yang dimana pada fitur tersebut memungkinkan pemilik dapat memantau kondisi serta lokasi kendaraan melalui *smartphone* yang mereka miliki.

Pada spesifikasi Yamaha Aerox 155 yang akan digunakan dalam melakukan penelitian ini adalah sebagai berikut:

1. Tipe Mesin : SOHC, 4 Langkah, Silinder Tunggal
2. Diameter x Langkah : 58 x 58,7 mm
3. Kapasitas Mesin : 155,1 cm^3
4. Daya Maksimum : 11,0 kW / 8.000 rpm
5. Torsi Maksimum : 13,8 Nm / 6.250 rpm
6. Sistem Pembakaran : Fuel Injection (FI)
7. Kompresi : 10,5 : 1
8. Transmisi : Sentrifugal Otomatis, Kering
9. Sistem Pendingin : Pendingin Cairan
10. Kapasitas Tanki : 4,6 Liter



Gambar 2.8 Yamaha Aerox 155 2018

2.3.7 Neural Network

Neural Network (NN) merupakan model algoritma yang mencoba meniru otak manusia, dimana NN dapat meniru otak manusia yang dapat memberikan stimulasi atau rangsangan, melakukan proses, dan memberikan *output* untuk menemukan hubungan diantara data yang diberikan. NN merupakan salah satu pemodelan kompleks yang dapat memprediksi bagaimana ekosistem merespon perubahan variabel lingkungan dengan terinspirasi oleh cara kerja sistem saraf biologis, khususnya pada sel otak manusia dalam memproses informasi. Algoritma yang paling populer digunakan oleh NN adalah *supervised*

learning. Namun, hal tersebut tidak menutup kemungkinan bahwa NN dapat digunakan untuk algoritma *semi-supervised* dan *unsupervised learning* (MIT News, 2017).

Secara umum, sebuah NN memiliki tiga jenis lapisan, yaitu lapisan input, lapisan tersembunyi, dan lapisan output. Lapisan input merupakan lapisan terluar yang akan menerima input dari dunia luar atau sistem lainnya. Kemudian pada lapisan tersembunyi terdiri atas sejumlah neuron yang akan memproses input dan mentransfer kembali informasi ke lapisan berikutnya. Pada lapisan output atau lapisan terakhir akan menghasilkan kelas atau nilai yang ingin di prediksi oleh jaringan.

2.3.7.1 Pembuatan Model Neural Network (NN)

Pada tahap ini, akan dibuat model pemetaan neural network yang digunakan untuk dapat diterapkan pada metode GA, PSO, dan FA guna mengoptimalkan desain *velocity stack* dengan *piggyback* pada kendaraan motor Yamaha Aerox 155. Langkah berikutnya adalah pelatihan dan pengujian model NN. Pada tahap ini, data yang telah diambil pada tahap pengumpulan data akan digunakan sebagai data input untuk pelatihan model. Setelah model dilatih, dilakukan pengujian untuk menguji performa model. Pembuatan model NN yang akan digunakan untuk optimalisasi desain *velocity stack* dengan *piggyback* pada kendaraan Yamaha Aerox 155. Model NN akan bertanggung jawab untuk mempelajari hubungan antara *velocity stack* dan *piggyback* dengan performa mesin, serta akan memberikan nilai *Mean Square Error* (MSE) yang paling rendah. MSE merupakan kuadrat dari rata-rata kesalahan. Pembuatan model NN melibatkan beberapa keputusan desain, termasuk jumlah neuron, arsitektur, fungsi aktivasi, dan algoritma pembelajaran yang digunakan. Berikut adalah penjelasan lebih rinci tentang beberapa faktor penting dalam pembuatan model NN:

1. Jumlah Neuron

Jumlah neuron atau unit pada setiap layer NN menjadi salah satu pertimbangan utama. Jumlah neuron dapat bervariasi tergantung pada kompleksitas masalah dan jumlah input serta output yang terlibat. Dalam mengoptimalkan desain *velocity stack* dengan *piggyback* pada kendaraan motor Yamaha Aerox 155, model NN dapat memiliki beberapa layer dengan jumlah neuron yang bervariasi.

2. Arsitektur

Arsitektur neural network merujuk pada tata letak layer dan hubungan antara mereka. Beberapa jenis arsitektur yang umum digunakan adalah *feedforward* neural network. Dalam kasus ini, model NN yang digunakan untuk mengoptimalkan desain *velocity stack*

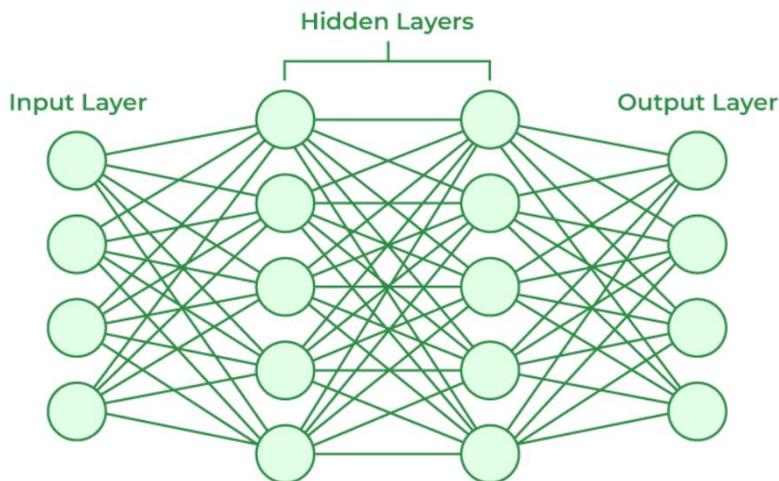
dengan *piggyback* pada kendaraan motor Yamaha Aerox 155 mungkin menggunakan arsitektur *feedforward* dengan beberapa *hidden layer*.

3. Fungsi Aktivasi

Fungsi aktivasi diterapkan pada setiap neuron dalam NN digunakan untuk memperkenalkan non-linearitas ke dalam model. Beberapa fungsi aktivasi yang umum digunakan adalah tansig dan logsig. Tansig merupakan fungsi aktivasi yang menampilkan kurva sigmoid yang serupa dengan fungsi sigmoid umumnya, tetapi dengan rentang nilai yang terdapat antara -1 dan 1, sedangkan pada logsig akan menampilkan rentang nilai antara 0 dan 1. Pilihan fungsi aktivasi akan mempengaruhi kemampuan model untuk mempelajari pola yang kompleks dan mempengaruhi kecepatan konvergensi algoritma pembelajaran. Dalam kasus ini, fungsi aktivasi yang sesuai seperti tansig atau logsig dapat digunakan.

4. Algoritma Pembelajaran

Algoritma pembelajaran digunakan untuk memperbarui bobot dan bias dalam NN selama proses pelatihan. Beberapa algoritma pembelajaran yang populer adalah *Levenberg-Marquardt* (LM), *Bayesian Regularization* (BR), dan *New Scaled Conjugate* (SCG). LM merupakan metode optimasi nonlinear yang digunakan untuk koreksi *error backpropagation* untuk menemukan bobot yang disesuaikan. Kemudian BR merupakan salah satu algoritma yang digunakan untuk mengatasi permasalahan *overfitting* yang sering terjadi dalam pembelajaran jaringan saraf tiruan. Sedangkan SCG merupakan metode pelatihan saraf yang baik dengan efisiensi kesederhanaan dan pemakaian memori yang rendah. Pilihan algoritma pembelajaran dapat mempengaruhi kecepatan konvergensi dan kualitas hasil model. Dalam kasus ini, algoritma seperti LM, BR, dan SCG dengan penyesuaian momentum atau varian dari algoritma tersebut dapat diterapkan.



Gambar 2.9 Model Arsitektur *Neural Network*

Sumber: C. Calin. <https://www.scalablepath.com/machine-learning/chatgpt-architecture-explained>

Selain faktor-faktor di atas, penting juga untuk memperhatikan parameter lain seperti *learning rate*, *batch size*, dan jumlah *epoch* yang digunakan selama pelatihan model. Parameter-parameter ini harus disesuaikan dan diuji secara empiris untuk mencapai hasil yang optimal dalam proses pelatihan dan evaluasi model NN. Setelah semua keputusan desain telah dipertimbangkan, langkah-langkah berikut dapat diikuti dalam pembuatan model NN:

1. Penentuan Input dan Output

Langkah pertama adalah menentukan input dan output dari model neural network. Dalam konteks mengoptimalkan desain *velocity stack* dengan *piggyback* pada kendaraan motor Yamaha Aerox 155, input dapat berupa kombinasi desain *velocity stack* dan setelan *piggyback*. Output yang diinginkan adalah performa mesin yang dioptimalkan, seperti torsi dan daya yang maksimal.

2. Pengumpulan Data

Data yang relevan dan representatif perlu dikumpulkan untuk melatih dan menguji model neural network. Data ini dapat mencakup hasil pengujian mesin menggunakan *velocity stack* dengan *piggyback* dengan berbagai kombinasi parameter, termasuk data daya dan torsi yang dihasilkan. Pengumpulan data ini dapat dilakukan dengan mengujicoba kendaraan Yamaha Aerox 155 dengan berbagai desain *velocity stack* dan setelan *piggyback* yang diinginkan, dan mengambil hasil pengujian menggunakan alat dyno.

3. *Preprocessing Data*

Sebelum melatih model neural network, data yang dikumpulkan perlu melalui tahap *preprocessing*. Hal ini melibatkan normalisasi data, pemisahan data menjadi set pelatihan dan set validasi, dan penyesuaian skala data jika diperlukan. *Preprocessing* data juga dapat melibatkan teknik seperti pengisian *missing value* atau penghapusan *outlier*.

4. Pembuatan Model

Setelah data siap, model *neural network* dapat dibangun menggunakan *library* atau *framework* yang sesuai seperti *TensorFlow* atau Keras. Model dapat dibuat dengan mendefinisikan layer-layer *neural network*, jumlah neuron dalam setiap layer, dan fungsi aktivasi yang sesuai. Selain itu, parameter-parameter seperti learning rate, batch size, dan jumlah epoch juga harus ditentukan.

5. Pelatihan Model

Setelah model neural network dibuat, langkah selanjutnya adalah melatih model menggunakan data pelatihan. Proses pelatihan melibatkan memasukkan data pelatihan ke dalam model, melakukan perhitungan *feedforward* dan *backward propagation* untuk memperbarui bobot dan bias, dan mengulangi langkah ini untuk beberapa epoch. Selama pelatihan, performa model dapat dievaluasi menggunakan metrik seperti loss function atau akurasi.

6. Evaluasi Model

Setelah model dilatih, performanya dievaluasi menggunakan set data validasi yang tidak digunakan selama pelatihan. Evaluasi dilakukan untuk memeriksa apakah model dapat menggeneralisasi dengan baik pada data yang belum pernah dilihat sebelumnya. Metrik evaluasi seperti *mean squared error* (MSE) dapat digunakan untuk menilai kualitas prediksi model.

7. Penyetelan dan Optimisasi

Jika model tidak memberikan hasil yang memuaskan, langkah penyetelan dan optimisasi dapat dilakukan. Hal ini melibatkan penyesuaian hyperparameter seperti learning rate, jumlah neuron, atau arsitektur model. Metode seperti cross-validation atau grid search dapat digunakan untuk menemukan kombinasi parameter yang optimal.

8. Validasi dan Verifikasi

Setelah model dianggap cukup baik, model dapat divalidasi menggunakan set data uji yang tidak pernah digunakan sebelumnya. Validasi dilakukan untuk memastikan bahwa model dapat menghasilkan prediksi yang akurat dan dapat digeneralisasi dengan baik pada data yang belum pernah dilihat sebelumnya. Selain itu, verifikasi model juga

dilakukan dengan membandingkan hasil prediksi model dengan data yang diperoleh dari pengujian aktual pada kendaraan motor Yamaha Aerox 155 dengan setelan *piggyback* yang dioptimalkan.

9. Analisis Hasil

Setelah model berhasil divalidasi dan diverifikasi, hasil prediksi model perlu dianalisis secara mendalam. Hal ini melibatkan pemahaman terhadap hubungan antara parameter *piggyback* yang dioptimalkan dengan performa mesin seperti torsi dan daya. Analisis ini dapat memberikan wawasan tentang bagaimana pengaturan parameter *piggyback* dapat mempengaruhi performa mesin dan memberikan rekomendasi untuk peningkatan lebih lanjut.

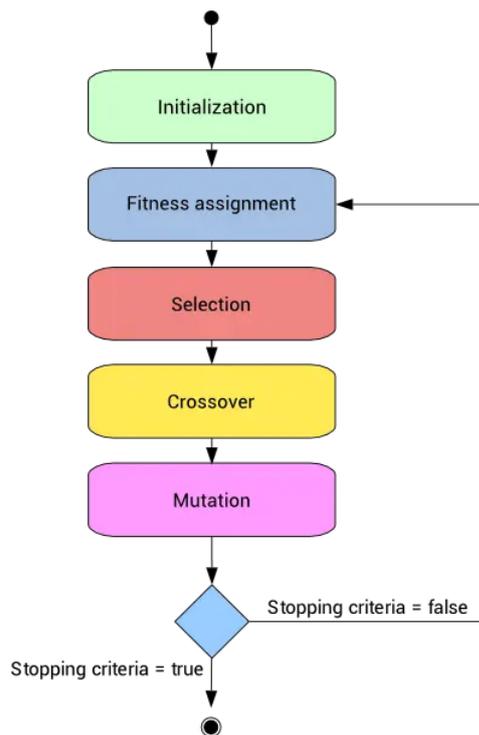
10. Dokumentasi

Tahap pelaksanaan juga melibatkan dokumentasi yang lengkap tentang semua langkah yang telah diambil, parameter yang digunakan, dan hasil yang diperoleh. Dokumentasi ini penting untuk memastikan reproduktibilitas penelitian dan memungkinkan peneliti lain untuk memahami dan menggunakan kembali metode yang telah dilakukan.

Tahap pelaksanaan merupakan inti dari penelitian ini, di mana model NN dibangun, dilatih, dan dievaluasi untuk mengoptimalkan desain *velocity stack* dengan *piggyback* pada kendaraan motor Yamaha Aerox 155. Dalam tahap ini, diperlukan keahlian dalam penggunaan *library* atau *framework* seperti *TensorFlow* sangat penting untuk mencapai hasil yang diharapkan.

2.3.8 Genetic Algorithm (GA)

Genetic Algorithm (GA) merupakan algoritma komputasi yang terinspirasi oleh teori evolusi Darwin. Teori tersebut menyatakan bahwa kelangsungan hidup suatu makhluk dipengaruhi oleh aturan yang kuat yang akan menang, selain itu hidup suatu makhluk dapat bertahan melalui proses reproduksi, *crossover*, dan mutasi. Konsep tersebut kemudian diadopsi menjadi algoritma komputasi agar dapat mencari solusi dari suatu permasalahan agar dapat menghasilkan solusi yang lebih mudah diimplementasikan (mathworks, n,d).



Gambar 2.10 Diagram Proses *Genetic Algorithm* (GA)

Sumber: Gómez, F. (2023).

https://www.neuraldesigner.com/blog/genetic_algorithms_for_feature_selection/

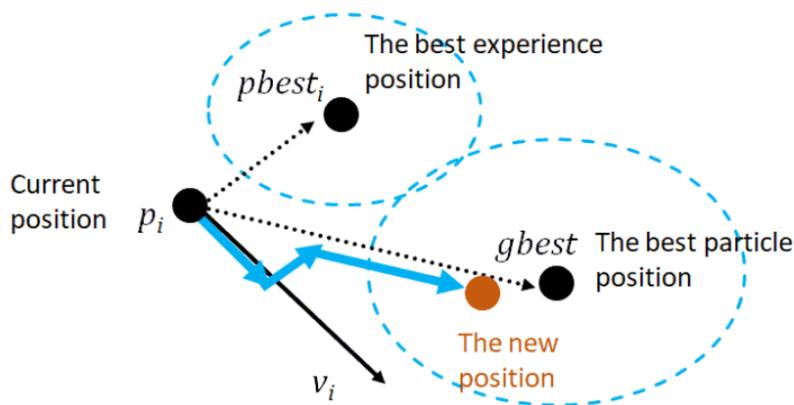
Dalam prosesnya, GA melibatkan komponen utama, antara lain representasi, inialisasi, evaluasi, seleksi, reproduksi dan crossover, mutasi, iterasi, dan konvergensi. Representasi adalah solusi dalam GA yang direpresentasikan dalam bentuk kromosom, yang sering kali berupa rangkaian gen atau string yang mewakili parameter yang diatur. Kemudian inialisasi adalah individu dalam populasi mewakili satu kromosom yang mungkin menghasilkan solusi. Pada evaluasi merupakan individu dalam populasi dinilai atau dievaluasi berdasarkan fungsi tujuan (*fitness function*) yang menentukan kualitas solusi guna mengukur sejauh mana individu memenuhi kriteria yang diinginkan. Seleksi merupakan penyeleksian individu terbaik yang memiliki fitness tertinggi dipilih untuk bertahan dan menjadi "orang tua" dalam generasi berikutnya. Reproduksi dan *crossover* merupakan proses reproduksi yang melibatkan kombinasi genetik antara individu yang terpilih melalui operasi *crossover*. Kemudian *crossover* akan menghasilkan keturunan baru dengan menggabungkan informasi genetik dari orang tua. Mutasi adalah operasi yang dilakukan dengan probabilitas rendah untuk memperkenalkan variasi baru dalam populasi. Mutasi mengacu pada perubahan acak pada kromosom, yang dapat membantu menjelajahi ruang solusi yang lebih luas. Pada iterasi, proses seleksi, reproduksi, crossover, dan mutasi berulang-ulang dilakukan dalam beberapa

generasi. Pada setiap iterasi, populasi akan berevolusi dan solusi yang lebih baik akan muncul seiring berjalanya waktu. Kemudian pada konvergensi GA akan berlanjut hingga mencapai kondisi berhenti, yaitu ketika mencapai solusi yang telah memenuhi kriteria yang diinginkan atau saat mencapai jumlah iterasi maksimum yang ditentukan.

Dalam penerapan GA, proses seleksi memainkan peran penting dalam mengidentifikasi individu yang paling cocok dalam suatu populasi, maka digunakan tiga metode berbeda: *Roulette Wheel* (RW), *Tournament*, dan *Random*. Pada metode RW, pemilihan akan dilakukan dengan menggunakan roda roulette. Yang dibangun dari kebugaran relatif (rasio kebugaran individu dan kebugaran total) masing-masing individu. Kemudian pada *tournament* akan dilakukan metode seleksi individu dari populasi individu dalam algoritma genetika, dimana pemenang dari setiap turnamen dipilih untuk dilakukan crossover. Kemudian pada metode *random*, akan dilakukan seleksi secara acak, dimana dalam strategi ini orang tua akan dipilih secara acak dari populasi yang ada. Mengingat GA memprioritaskan “populasi terbaik”, strategi seleksi ini dipilih secara strategis untuk membedakan kandidat yang paling menjanjikan. Melalui evaluasi yang ketat, individu-individu terbaik dipilih, memastikan mereka menjadi landasan untuk menghasilkan prediksi paling akurat dalam uji coba di dunia nyata. Dalam studi ini, ketiga metode ini akan digunakan untuk menyempurnakan model prediktif. Penerapan kolektif mereka bertujuan untuk mencapai tingkat optimalisasi tertinggi.

2.3.9 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) merupakan metode optimisasi yang terinspirasi oleh perilaku kawanan atau kelompok partikel dalam mencari solusi optimal. Sistem pada PSO akan bekerja dengan menggabungkan konsep dari alam semesta dan interaksi sosial untuk mencari solusi terbaik dalam ruang pencarian yang kompleks. Dalam PSO, solusi akan dicari dengan mengoptimalkan fungsi tujuan (*fitness function*) yang ingin diminimalkan atau dimaksimalkan. Proses optimisasi PSO melibatkan sejumlah partikel yang bergerak dalam ruang pencarian, dan setiap partikel mewakili satu kemungkinan solusi (Tam, A., 2021).



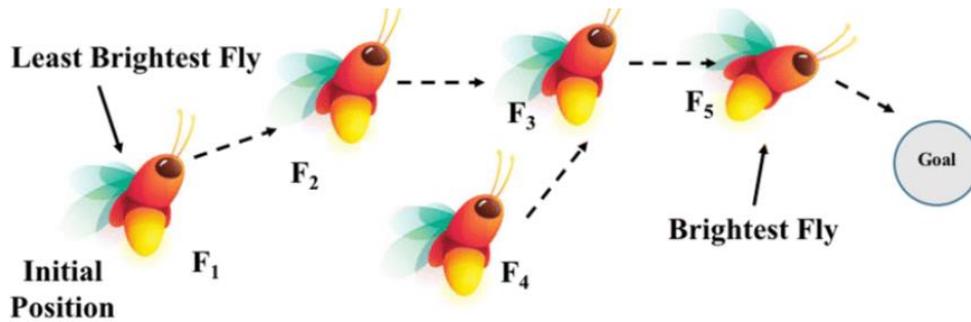
Gambar 2.11 Model Ilustrasi pada Metode *Particle Swarm Optimization (PSO)*

Sumber: Rhim, H. (2024). <https://www.baeldung.com/cs/psa>

Pada PSO, pada tiap partikel memiliki atribut, antara lain posisi, kecepatan, *Personal Best* (PBest), dan *Global Best* (Gbest). Pada atribut posisi akan menentukan representasi solusi yang diusulkan oleh partikel di dalam ruang pencarian. Kemudian pada atribut kecepatan akan menentukan seberapa cepat partikel yang mengarahkan pergerakan partikel di dalam ruang pencarian. *Personal Best* (PBest) merupakan solusi terbaik yang pernah ditemukan oleh partikel tersebut selama perjalanannya berlangsung. Kemudian pada *Global Best* (GBest) adalah solusi terbaik yang pernah ditemukan oleh seluruh partikel dalam populasi.

2.3.10 *Firefly Algorithm (FA)*

Firefly Algorithm (FA) dapat dikategorikan sebagai algoritma cerdas, metaheuristik, dan terinspirasi oleh alam. Secara khusus, kunang-kunang dalam populasi menunjukkan aktivitas kilatan cahaya yang khas, berperan sebagai daya tarik untuk menarik pasangan, sebagai sarana komunikasi, dan sebagai peringatan risiko terhadap predator. Metode ini dirancang dengan asumsi bahwa semua kunang-kunang memiliki satu pasangan dan bahwa daya tarik antara mereka sejalan dengan tingkat kecerahan individu. Dengan kata lain, kunang-kunang yang lebih terang akan menarik kunang-kunang yang kurang terang menuju ke arahnya. Jika tidak ada kunang-kunang yang lebih terang dari suatu kunang-kunang, maka kunang-kunang tersebut akan bergerak secara acak (Yarpiz, 2020).



Gambar 2.12 Diagram Kerja Metode *Firefly Algorithm* (FA).

Sumber: Sharma, K., Doriya, R. (2020). https://www.researchgate.net/figure/Diagram-representing-working-of-Firefly-algorithm_fig3_341248500

Dalam formulasi algoritma kunang-kunang, fungsi tujuan terkait dengan ciri-ciri berkedip cahaya dari populasi kunang-kunang. Prinsip fisika menyatakan bahwa intensitas cahaya berbanding terbalik kuadrat dengan jarak, yang memungkinkan penentuan fungsi kecocokan jarak antara dua kunang-kunang. Untuk mengoptimalkan fungsi penyesuaian, individu dalam populasi dipaksa untuk berpindah secara sistematis atau acak. Tujuannya adalah memastikan bahwa semua kunang-kunang beralih ke kunang-kunang yang lebih menarik dengan kilatan cahaya yang lebih terang, sehingga populasinya berkumpul di sekitar kunang-kunang paling terang.

Proses ini melibatkan tiga parameter utama dalam eksekusi algoritma kunang-kunang, yaitu daya tarik, pengacakan, dan penyerapan. Parameter daya tarik bergantung pada intensitas cahaya antara dua kunang-kunang dan dijelaskan melalui fungsi eksponensial. Ketika parameter ini diatur ke nol, terjadi pergerakan acak sesuai dengan parameter pengacakan yang diatur oleh prinsip distribusi Gaussian, yang menghasilkan bilangan acak dari interval [0,1]. Sebaliknya, parameter penyerapan memengaruhi nilai daya tarik dan dapat bervariasi dari nol hingga tak terbatas. Dalam kasus konvergensi tak terbatas, gerakan kunang-kunang menyerupai perjalanan acak yang diatur oleh parameter-parameter tersebut.

2.3.11 Optimasi Piggyback Menggunakan Metode GA, PSO, dan FA

Dalam melakukan optimasi desain *velocity stack* pada mesin Yamaha Aerox 155 dengan *piggyback* akan menggunakan *Neural Network* (NN). NN akan digunakan sebagai alat untuk memodelkan sistem penghubung antara variabel masukan dan keluaran. Umumnya NN digunakan untuk pengoptimalan *black box*, yang dimana tidak diketahui hubungan matematis antara variabel masukan dan keluaran. Maka dari itu, untuk menggunakan NN akan dilakukan

menggunakan pendekatan seperti *Genetic Algorithm (GA)*, *Particle Swarm Optimization (PSO)*, dan *Firefly Algorithm (FA)* agar mendapat hasil yang akurat.

Penggunaan *Artificial Intelligence (AI)* dalam mengoptimasi desain *velocity stack* pada mesin Yamaha Aerox 155 dengan *piggyback* bertujuan untuk meningkatkan daya dan torsi pada mesin. Pada langkah awal mulai, NN akan digunakan untuk memodelkan hubungan antara variabel masukan dan keluaran pada kendaraan, seperti kecepatan kendaraan dan putaran mesin. Dengan demikian, Neural Network dapat digunakan untuk memprediksi efek dari variasi parameter kendaraan pada performa kendaraan. Berikutnya, hasil hubungan antara variabel masukan dan keluaran akan di proses dengan metode GA, PSO, dan FA. Dimana ketiga metode tersebut akan mengeluarkan hasil desain *velocity stack* dan setelan *piggyback* paling optimal untuk dapat di aplikasikan pada Yamaha Aerox 155.

Untuk optimasi *piggyback* menggunakan metode GA, memiliki langkah-langkah sebagai berikut:

1. Inisialisasi Populasi Awal

Populasi awal akan terdiri dari sejumlah individu yang memiliki kombinasi parameter yang acak.

2. Evaluasi Populasi

Tiap individu dalam populasi akan dilakukan proses evaluasi dengan menghitung nilai *fitness* berdasarkan performa mesin kendaraan dengan parameter yang diberikan.

3. Seleksi Individu

Individu yang memiliki nilai *fitness* tertinggi akan dipilih sebagai orangtua untuk menghasilkan generasi berikutnya.

4. *Genetic Operation*

Genetic operation seperti halnya dengan crossover dan mutase, dimana akan diterapkan pada orangtua untuk menghasilkan individu baru.

5. Evaluasi Populasi Baru

Populasi baru terdiri dari individu-individu orangtua dan anak yang telah dievaluasi menggunakan langkah evaluasi populasi yang sama seperti pada langkah kedua.

6. Iterasi

Pada langkah 3 hingga 5 akan diulang secara iteratif hingga mencapai kriteria berhenti yang ditentukan atau telah mencapai nilai *fitness* yang memadai.

Berikutnya, untuk optimasi *piggyback* menggunakan metode PSO, memiliki langkah-langkah sebagai berikut:

1. Inisialisasi Partikel

Pada tiap partikel dalam populasi awal akan memiliki posisi dan kecepatan awal yang acak, sehingga harus memiliki inisial agar dapat diidentifikasi.

2. Evaluasi Partikel

Partikel akan dievaluasi dengan cara menghitung nilai *fitness* berdasarkan performa mesin kendaraan dengan parameter yang diberikan oleh posisi partikel tersebut.

3. Pembaruan Posisi dan Kecepatan Partikel

Setiap partikel akan memperbarui posisi dan kecepatannya berdasarkan posisi terbaik yang pernah dicapai oleh partikel itu sendiri (PBest) dan posisi terbaik yang pernah dicapai oleh seluruh partikel dalam populasi (GBest).

4. Evaluasi Partikel Baru

Partikel baru yang telah memperbarui posisi dan kecepatannya dievaluasi menggunakan langkah evaluasi partikel yang sama seperti pada langkah kedua.

5. Iterasi

Pada langkah 3 hingga 4 diulang secara iteratif hingga mencapai kriteria berhenti yang ditentukan, misalnya mencapai jumlah iterasi tertentu atau mencapai nilai *fitness* yang memadai.

Metode Firefly Algorithm (FA) adalah algoritma optimasi yang terinspirasi dari perilaku serangga kunang-kunang (firefly). Berikut adalah langkah-langkah umum yang dapat diikuti untuk menggunakan metode Firefly Algorithm:

1. Inisialisasi Populasi

- Menentukan jumlah firefly.
- Tetapkan posisi awal firefly secara acak di dalam ruang pencarian (search space).

2. Evaluasi Objektif

- Menghitung nilai fungsi objektif untuk setiap firefly berdasarkan posisinya dalam ruang pencarian.

3. Pembaruan Posisi

- Menggunakan aturan pembaruan posisi firefly untuk memindahkan firefly ke arah firefly lain yang memiliki nilai fungsi objektif lebih baik.
- Pembaruan posisi dapat menggunakan persamaan berdasarkan intensitas cahaya (nilai objektif) dan jarak antara firefly.

4. Evaluasi Posisi Baru

- Menghitung nilai fungsi objektif untuk firefly yang telah memperbarui posisinya.

5. Seleksi Elitisme

- Memilih firefly terbaik berdasarkan nilai fungsi objektif untuk tetap bertahan ke generasi berikutnya.

6. Iterasi

- Mengulangi langkah-langkah 3 hingga 5 untuk beberapa iterasi atau hingga kriteria konvergensi terpenuhi.

7. Output Solusi Terbaik

- Apabila iterasi selesai, output posisi firefly dengan nilai fungsi objektif terbaik sebagai solusi optimal.

Langkah-langkah di atas adalah panduan umum, dan implementasi khusus dari Firefly Algorithm dapat bervariasi tergantung pada konteks masalah yang sedang diatasi. Pemilihan parameter seperti ukuran populasi, parameter pembaruan posisi, dan kriteria konvergensi dapat memengaruhi kinerja algoritma. Oleh karena itu, eksperimen dan penyesuaian parameter mungkin diperlukan untuk mengoptimalkan kinerja Firefly Algorithm pada masalah tertentu.

Selama proses optimasi, populasi GA, partikel PSO, dan pemilihan *firefly* terbaik akan mengalami iterasi dan pembaruan secara berulang. Pada setiap iterasi, nilai *fitness* dari setiap individu atau partikel akan dihitung berdasarkan hasil prediksi model neural network menggunakan kombinasi parameter yang diwakilinya. Nilai *fitness* akan menjadi ukuran kualitas atau performa dari kombinasi parameter tersebut. Selanjutnya, berdasarkan nilai *fitness*, langkah seleksi, *crossover*, dan mutasi akan dilakukan pada individu-individu atau partikel-partikel yang terlibat dalam proses optimasi. Individu-individu atau partikel-partikel yang memiliki nilai *fitness* lebih baik akan memiliki peluang yang lebih besar untuk dijadikan orangtua (parent) dalam proses reproduksi. Hal ini akan menghasilkan generasi baru individu atau partikel yang diharapkan memiliki kombinasi parameter yang lebih baik.

Proses evaluasi, pembaruan, dan iterasi ini akan terus berlanjut hingga mencapai kondisi berhenti yang telah ditentukan sebelumnya, seperti mencapai jumlah iterasi maksimum atau mencapai nilai *fitness* yang memadai. Dengan demikian, tahap pelaksanaan pada penelitian ini akan melibatkan penggunaan metode GA, PSO, dan FA untuk mencari kombinasi antara desain *velocity stack* dengan setelan konfigurasi pada *piggyback*.