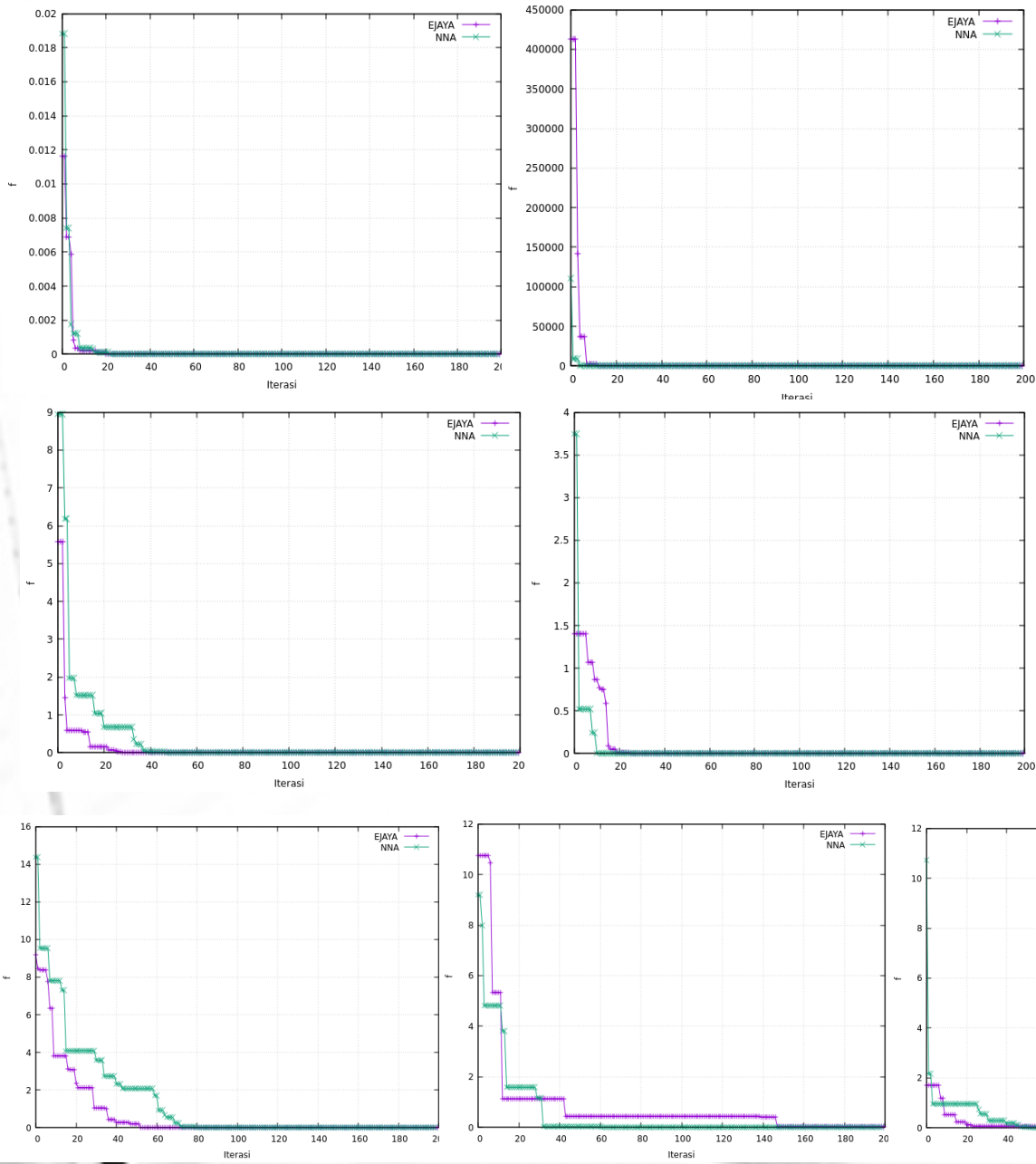


# Optimum Design

Optimizing the design of  
engineering systems

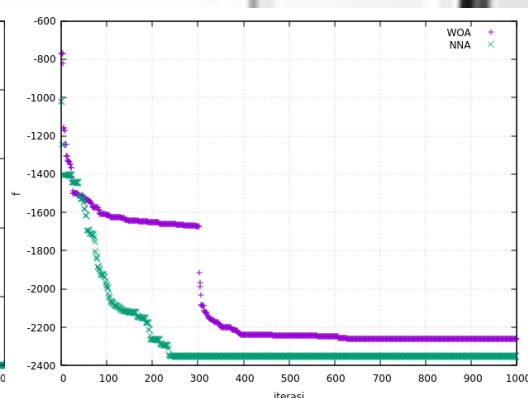
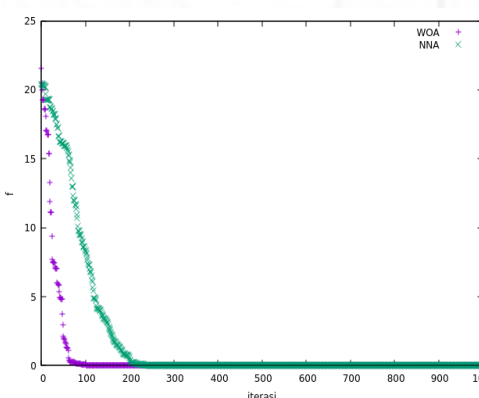
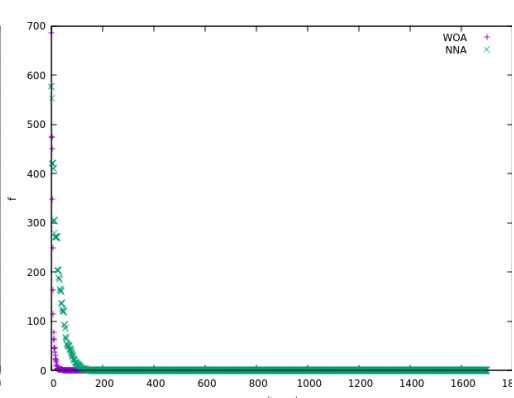
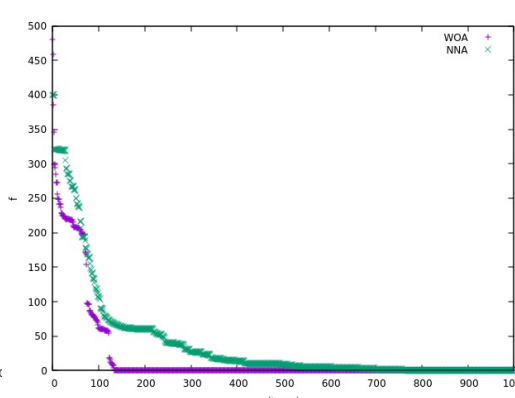
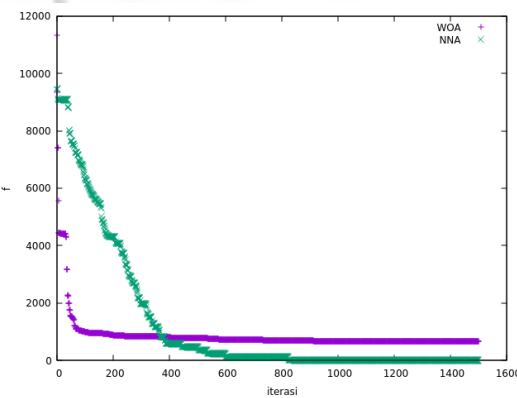
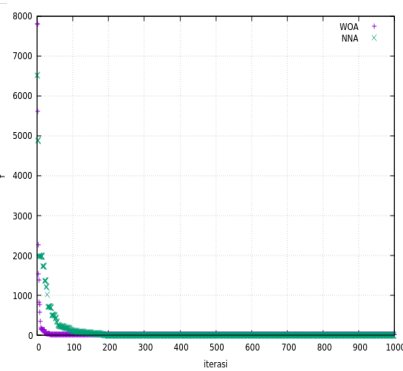
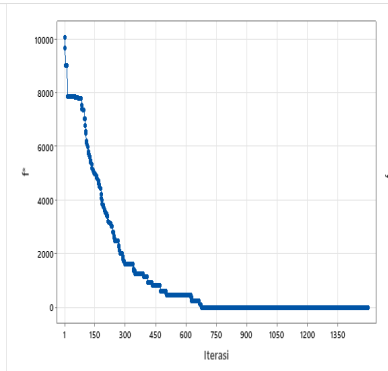
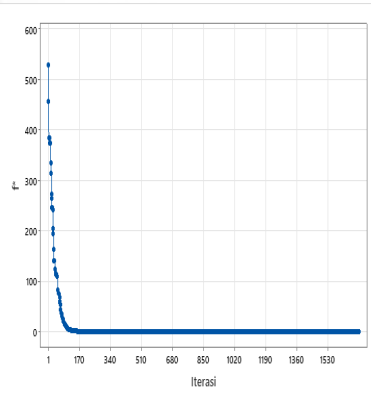
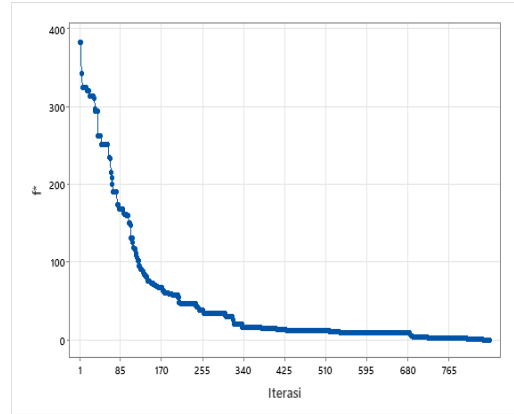
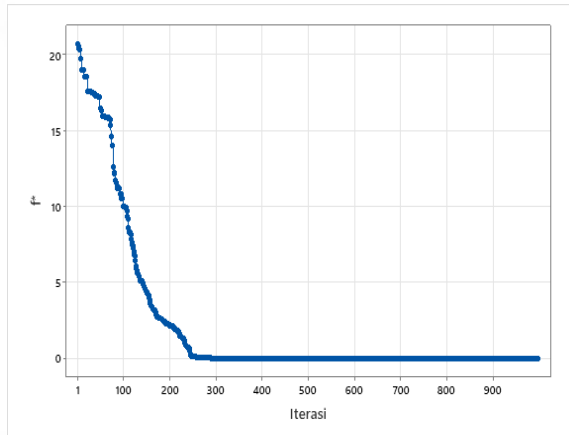


## 4. Percobaan Numerik



Untuk melakukan verifikasi terhadap performa dari metode yang diusulkan pada penelitian ini, kami melakukan sejumlah percobaan numerik yang diambil dari berbagai referensi terkait optimisasi dan membandingkan performa yang dihasilkan oleh metode yang diusulkan pada penelitian ini dengan berbagai metode yang telah diusulkan oleh para peneliti seperti Whale Optimization Algorithm (Mirjalili dkk.2016), Enhanced Jaya Algorithm (Zhang dkk.2021), Spiral Dynamic Optimization Algorithm (Tamura dkk. 2011), Social Network Search (Bayzidi dkk.2021), Hybrid multi-level cross entropy moth-flame (Hamzekolaei dkk.2021), Cuckoo Search Algorithm (Gandomi dkk. 2013), Rule based reinforcement learning (Radaideh dkk.2021), WCA (Eskandar dkk. 2012), Artificial bee colony algorithm (Akay dkk.2012), IPSO (Guedria 2015), Search and rescue optimization algorithm (Shabani dkk.2020), Chaos game optimization (Talatahari dkk.2020), Mine blast algorithm (Sadollah dkk.2013). Pengujian dilakukan terhadap fungsi-fungsi yang terdiri dari  $n=3, 30, 100$  dimensi dengan kondisi maximum iterasi pada fungsi berdimensi 100 adalah  $K_{max} = 1500$  dan maximum  $NPOP$  pada fungsi berdimensi 100 adalah 120. Pengujian dilakukan pada komputer dekstop dengan sistem operasi fedora workstation, linux, RAM 4 Gb dengan bahasa pemrograman c++ dan compiler g++. Untuk setiap hasil yang membutuhkan plot/ visualisasi laju konvergensi dengan metode lainnya, visualisasi dilakukan menggunakan Gnuplot.

## 4.1 WOA(Mirjalili dkk.2016) vs NNA

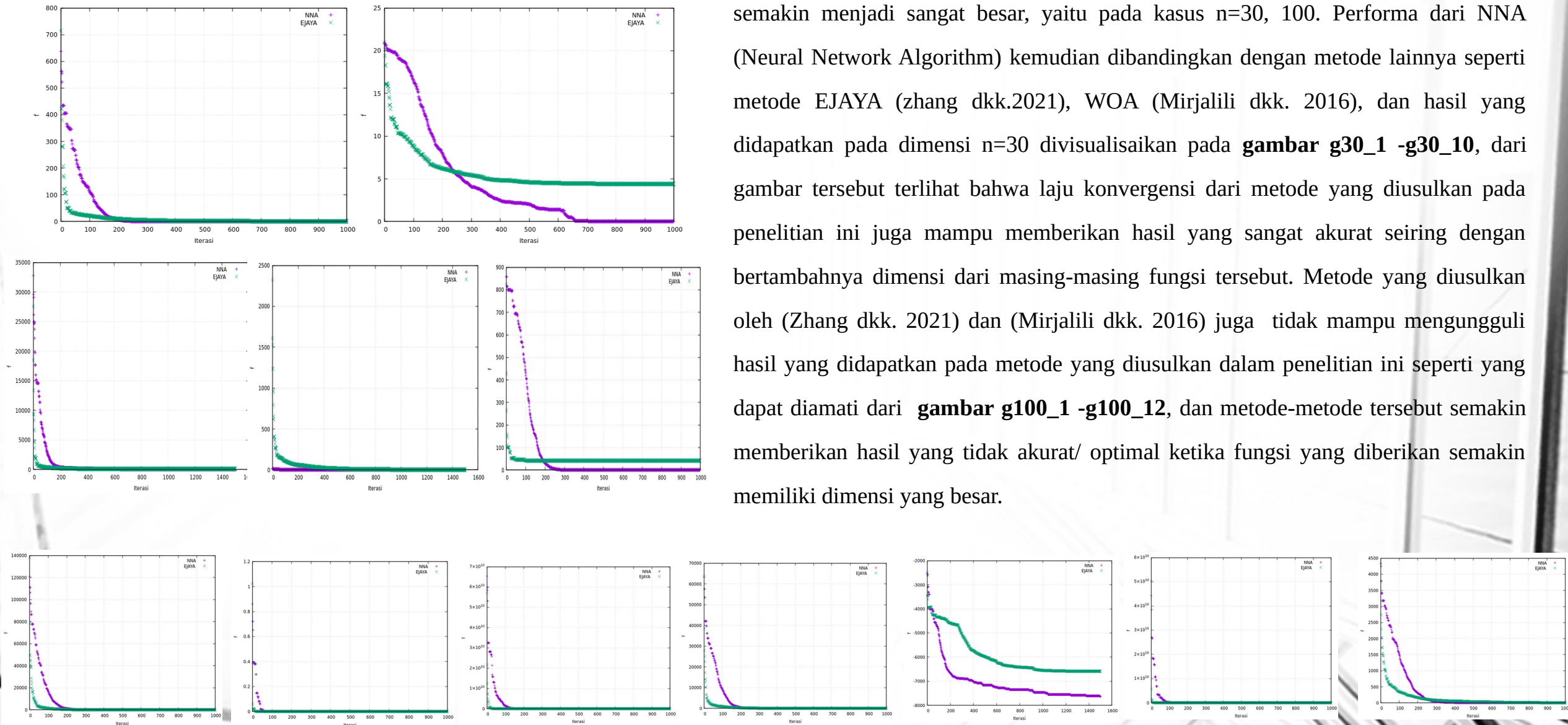


Gambar g03\_1 - g03-10 menunjukkan laju konvergensi dari metode yang diusulkan menggunakan metode NNA, NNA berhasil konvergen sangat cepat untuk permasalahan yang melibatkan 3 dimensi. Hasil percobaan yang sama juga telah dilakukan pada metode SDOA (Tamura, 2011), dimana untuk menghasilkan hasil yang akurat seperti yang telah diimplementasikan pada NNA, metode tersebut harus diseekekusi secara berulang-ulang serta dilakukan pemilihan parameter yang tepat. Table.1 juga menunjukan NNA dapat menemukan titik global minimum dalam satu kali eksekusi dengan tanpa pemilihan parameter-parameter yang berubah-ubah seperti yang diusulkan oleh (zhiwen cheng dkk, 2021), (Wang dkk. 2019) dan (Cheng dkk. 2019). Pada gambar tersebut juga terlihat bahwa metode yang diusulkan pada penelitian ini berhasil mendapatkan hasil optimal/ titik global minimum dengan ketelitian yang sangat baik, bahkan hasil yang didapatkan mampu mengungguli metode yang diusulkan (Zhang dkk.2020) pada fungsi-fungsi yang memiliki banyak sekali titik lokal, seperti yang terlihat dari **gambar g03\_1 – g03-10**.



## 4.1 EJAYA(ZHANG dkk. 2021) vs NNA

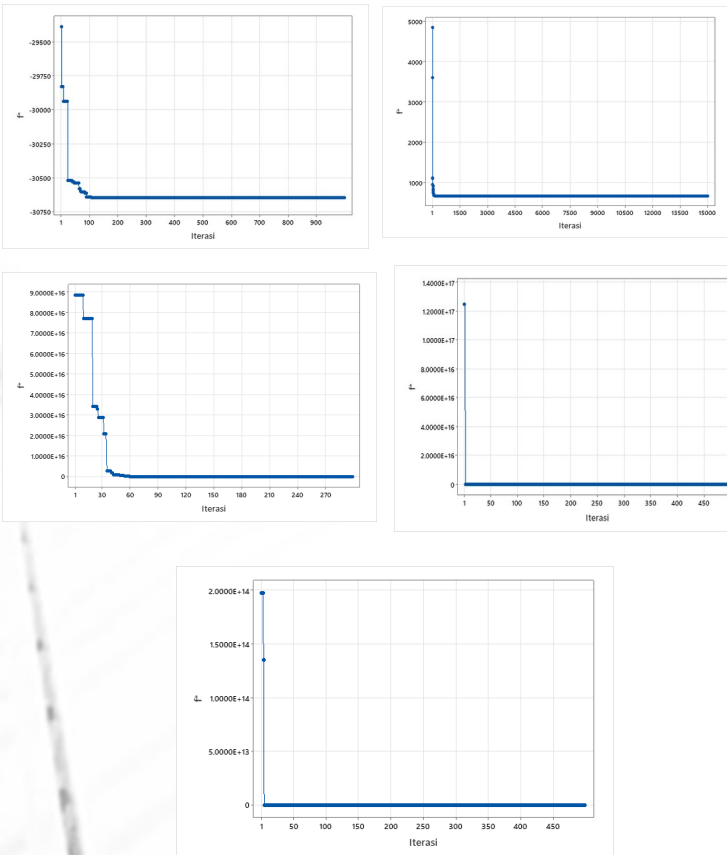
Untuk mendapatkan perbandingan yang lebih akurat, metode yang diusulkan pada penelitian ini juga diuji kembali pada kasus dimana dimensi dari suatu fungsi semakin menjadi sangat besar, yaitu pada kasus  $n=30, 100$ . Performa dari NNA (Neural Network Algorithm) kemudian dibandingkan dengan metode lainnya seperti metode EJAYA (Zhang dkk. 2021), WOA (Mirjalili dkk. 2016), dan hasil yang didapatkan pada dimensi  $n=30$  divisualisasikan pada **gambar g30\_1 -g30\_10**, dari gambar tersebut terlihat bahwa laju konvergensi dari metode yang diusulkan pada penelitian ini juga mampu memberikan hasil yang sangat akurat seiring dengan bertambahnya dimensi dari masing-masing fungsi tersebut. Metode yang diusulkan oleh (Zhang dkk. 2021) dan (Mirjalili dkk. 2016) juga tidak mampu mengungguli hasil yang didapatkan pada metode yang diusulkan dalam penelitian ini seperti yang dapat diamati dari **gambar g100\_1 -g100\_12**, dan metode-metode tersebut semakin memberikan hasil yang tidak akurat/ optimal ketika fungsi yang diberikan semakin memiliki dimensi yang besar.



Problem	Dim		Optimal Solution	Maximum Iteration MAX=100					Maximum Iteration MAX=1000				
				NNA	Spiral 1	Spiral 2	Spiral 3	Spiral 4	NNA	Spiral 1	Spiral 2	Spiral 3	Spiral 4
Griewank	3	Mean	0	0.0222148436774482	0.06	0.06	0.12	0.13	0.00530191051332028	0.1	0.1	0.03	0.03
		Best		1.18454E-10	0.01	0.04	0.03	0.04	0	0.01	0	0	0
		Worst		0.0962749	0.2	0.17	0.23	0.3	0.0221913	0.2	0.2	0.1	0.2
		Std. Dev		0.0246691374524685	0.04	0.04	0.04	0.1	0.00665937048944994	0.04	0.04	0.02	0.03
	30	Mean		0.12710849525	1.6	1	3.3	2	0.00579195612361332	1.5	1	0.8	0.2
		Best		0.000161573	1.1	0.7	2.2	1.8	0	1.1	0.3	0.1	0.01
		Worst		1.24733	3.2	1.1	4.7	2.1	0.0221561	3.2	1.1	0.5	0.9
		Std. Dev		0.284656220383767	0.4	0.1	0.5	0.07	0.00738350627741022	0.4	0.2	0.4	0.2
	100	Mean		0.83330335	8.6	1.8	18	4.5	0.004441456455005	8.3	1.7	2.5	1.6
		Best		0.324137	4.6	1.5	15	4.1	0.00000000248822	4.3	1.4	1.3	1.4
		Worst		4.83659	12	2	20	4.9	0.0197176	11	1.9	4.9	1.9
		Std. Dev		1.01397868944545	1.4	0.1	1.1	0.1	0.00718193113874875	1.4	0.1	1	0.1
Rastrign	3	Mean	0	0.000429622438546253	1.87	1.46	3.86	5.7	1.13687E-14	1.8	1.4	0.7	1.4
		Best		8.52651E-14	0.01	0	0.23	1.6	0	0	0	0	0
		Worst		0.00859071	7.96	5.98	8.57	11	1.03029E-13	8	6	3	6
		Std. Dev		0.00192092070576189	1.39	1.15	0.23	2.2	2.582822E-14	1.4	1.1	0.7	1.1
	30	Mean		112.65927	230	98	380	272	33.9604975	209	71	149	55
		Best		49.6442	152	49	304	238	4.97481	133	27	69	23
		Worst		181.811	391	189	456	317	169.204	369	161	290	124
		Std. Dev		35.3338609483892	40	26	35	15	38.9973495147472	39	26	39	21
	100	Mean		623.86525	1174	628	1622	1032	193.28869	1049	550	777	445
		Best		343.231	954	516	1503	950	28.1404	810	432	596	345
		Worst		1023.59	1313	729	1754	1093	483.107	1202	659	1000	595
		Std. Dev		152.220807413329	70	48	50	27	99.9877499666829	72	48	79	52
Schwefel	3	Mean	0	5.03805807554087E-05	0	0	0.2	0.3	7.16562413E-18	0	0	0	0
		Best		2.06535E-17	0	0	0.01	0.02	3.68364E-94	0	0	0	0
		Worst		0.00100761	0.03	0	0.66	2	5.74099E-17	0.03	0	0	0
		Std. Dev		0.000225308426473062	0.01	0	0.13	0.2	1.45693263530252E-17	0	0	0	0
	30	Mean		17.7815989884	99	20	226	154	3.395943859015E-17	95	17	13	3
		Best		0.000750834	25	4.1	107	67	4.65108E-87	22	2.9	0.8	1
		Worst		344.33	265	91	447	487	5.55226E-16	259	85	44	17
		Std. Dev		76.8845294156442	51	16	70	64	1.23363685202368E-16	51	15	8	2
	100	Mean		1057.16095	1293	255	2915	1553	0.00000558933673	1235	225	731	73
		Best		20.4652	618	88	1267	875	0.0000000272067	571	75	291	38
		Worst		15141.6	2777	964	7104	3203	0.00010912	2698	894	1821	243
		Std. Dev		3415.00919943056	384	147	972	456	2.43688319140391E-05	376	137	297	35
2 <sup>nd</sup> minima	3	Mean	-78Dim	-234.9883	-229	-221	-224	-211	-234.99665	-229	-221	-234	-222
		Best		-234.997	-235	-235	-235	-234	-234.997	-235	-235	-235	-235
		Worst		-234.905	-174	-178	-198	-168	-234.994	-174	-178	-207	-178
		Std. Dev		0.0205198235346905	14.3	15.3	7	16	0.000745159820374153	14	15	5	15
	30	Mean		-1892.425	-1798	-1846	-1076	-1246	-2307.3215	-1815	-1868	-1995	-1989
		Best		-2177.11	-2007	-2038	-1375	-1360	-2349.97	-2024	-2057	-2194	-2208
		Worst		-1614.27	-1563	-1649	-682	-1080	-1952.2	-1568	-1671	-1802	-1728
		Std. Dev		168.77507992966	96	81	150	57	100.833531039244	94	79	76	73
	100	Mean		-5001.226	-4373	-4724	-2001	-3816	-7211.546	-4864	-4775	-6317	-5192
		Best		-5924.5	-5417	-5074	-2703	-4141	-7458.74	-5553	-5108	-6744	-5546
		Worst		-4068.04	-4055	-4336	-1076	-3483	-4955.42	-4167	-4367	-5942	-4805
		Std. Dev		543.415260710105	267	159	311	109	121.983426156352	266	160	186	154
Rosenbrock	3	Mean	0	7.140303005E-11	N.A	N.A	N.A	N.A	2.0722175E-13	N.A	N.A	N.A	N.A
		Best		6.10768E-33	N.A	N.A	N.A	N.A	5.60914E-125	N.A	N.A	N.A	N.A
		Worst		3.19211E-10	N.A	N.A	N.A	N.A	2.38385E-12	N.A	N.A	N.A	N.A
		Std. Dev		1.1174808929E-10	N.A	N.A	N.A	N.A	5.4032039E-13	N.A	N.A	N.A	N.A
	30	Mean		0.3873193470275	N.A	N.A	N.A	N.A	5.794450469E-11	N.A	N.A	N.A	N.A
		Best		0.00000326103	N.A	N.A	N.A	N.A	1.13352E-14	N.A	N.A	N.A	N.A
		Worst		3.55806	N.A	N.A	N.A	N.A	4.37839E-10	N.A	N.A	N.A	N.A
		Std. Dev		0.893176097092027	N.A	N.A	N.A	N.A	1.1444768055E-10	N.A	N.A	N.A	N.A
	100	Mean		116.0749904	N.A	N.A	N.A	N.A	6.58085041035941	N.A	N.A	N.A	N.A
		Best		0.634495	N.A	N.A	N.A	N.A	0.00000000347557	N.A	N.A	N.A	N.A
		Worst		987.914	N.A	N.A	N.A	N.A	131.617	N.A	N.A	N.A	N.A
		Std. Dev		257.061210271952	N.A	N.A	N.A	N.A	29.4304558028704	N.A	N.A	N.A	N.A

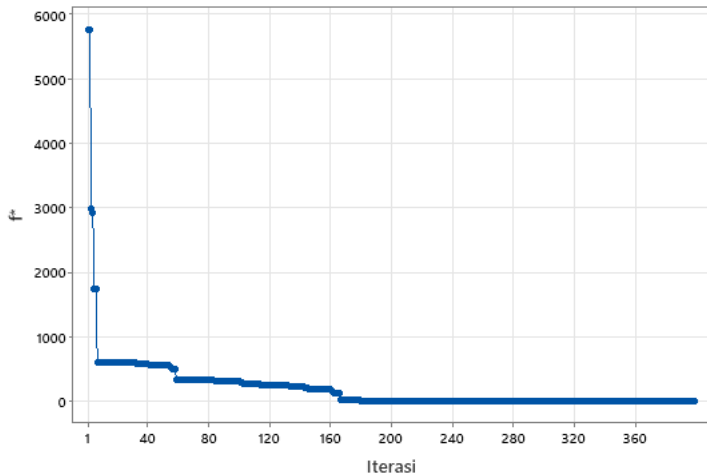
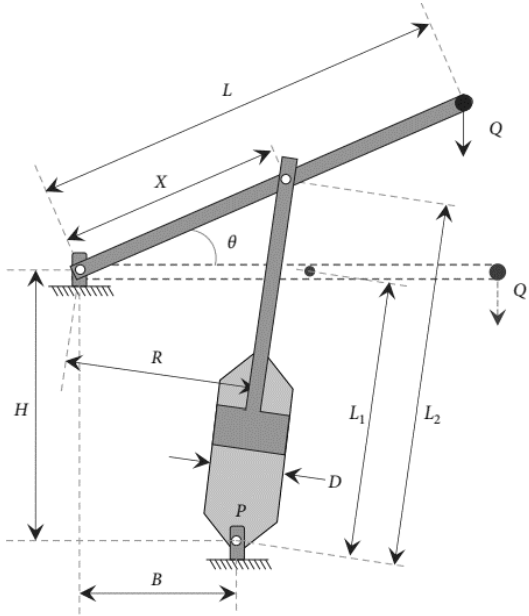
Table 1. Hasil Percobaan Numerik, Base case performance, average case, worst case performance and standart deviation

# Benchmark A.



Setelah melakukan pengujian terhadap fungsi berdimensi  $n=3, 30, 100$ , pada bagian ini fungsi-fungsi benchmark berkendala diuji kembali untuk melihat bahwa metode yang diusulkan mampu menghasilkan performa terbaik seperti yang telah diimplementasikan pada kasus fungsi benchmark tanpa kendala berdimensi besar. Pengujian dilakukan terhadap sejumlah fungsi-fungsi berkendala yang diusulkan oleh beberapa peneliti seperti (Hailong Wang dkk. 2017), (Qiang Zhao dkk. 2020), (K. Hans Raj dkk. 2016). Untuk mendemonstrasikan efektivitas dari metode yang diusulkan pada penelitian ini, hasil yang didapatkan akan dibandingkan terhadap metode-metode yang telah digunakan dalam menyelesaikan permasalahan yang sama. Untuk setiap permasalahan yang diusulkan, pengujian dilakukan dengan maksimal  $NPOP = 120$ , hasil statistik yang didapatkan untuk performa terbaik, performa rata-rata dan performa terburuk dirangkum dalam table 1. Tidak seperti metode yang diusulkan oleh (Hailong Wang dkk. 2017), (Qiang Zhao dkk. 2020), (K. Hans Raj dkk. 2016), Whale Optimization Algorithm (Mirjalili dkk.2016), Enhanced Jaya Algorithm (Zhang dkk.2021), Spiral Dynamic Optimization Algorithm (Tamura dkk. 2011). Jika dipandang dari segi implementasi, NNA memiliki kesederhanaan dalam menyelesaikan setiap permasalahan pada fungsi benchmark berkendala. Selain itu hasil yang didapatkan pada permasalahan g01 – g02 juga telah berhasil mendapatkan hasil optimal seperti yang didapatkan oleh (Hailong Wang dkk. 2017). Pada fungsi benchmark g03-g04 juga dapat dilihat bahwa metode yang diusulkan pada penelitian ini mampu memberikan hasil yang sangat akurat seperti yang diusulkan oleh (Qiang Zhao dkk. 2020) dan hasil yang didapatkan pada fungsi benchmark berkendala yang diusulkan oleh (K. Hans Raj dkk. 2016) juga telah mendekati hasil optimal seperti yang diusulkan oleh (K. Hans Raj dkk. 2016). Berdasarkan **gambar 1 - gambar 10** juga terlihat bahwa metode yang diusulkan dalam penelitian ini mampu dan memiliki laju konversi yang sangat baik seperti pada metode Whale Optimization Algorithm (Mirjalili dkk.2016), Enhanced Jaya Algorithm (Zhang dkk.2021).

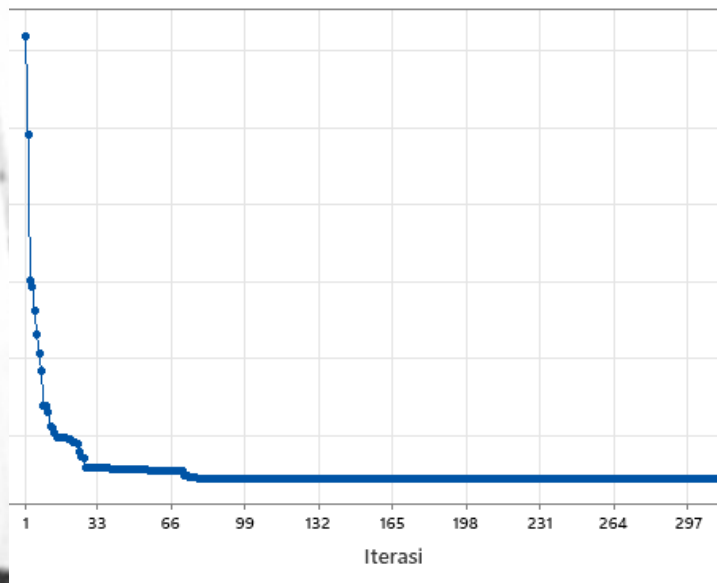
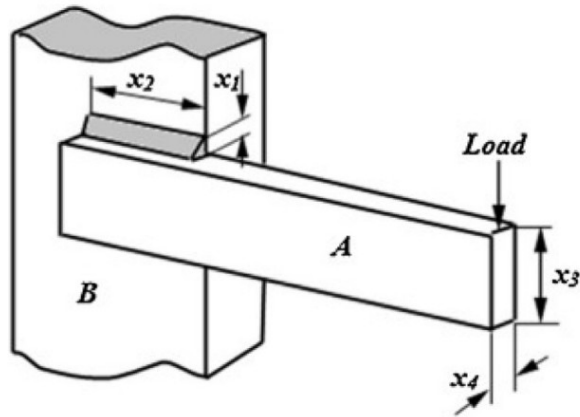
## 4.2 Benchmark - Piston Lever



**Gambar 5.** menunjukkan struktur dari sebuah piston lever, permasalahan piston lever telah dikaji oleh beberapa peneliti seperti (Bayzidi dkk. 2021) dengan memanfaatkan metode social network search, (Hamzehkolaei dkk. 2021) melakukan proses hybrid multi-level cross-entropy-based moth-flame dan (Gandomi dkk.2013) memanfaatkan Cuckoo search algorithm. Inti utama dari permasalahan ini adalah menempatkan/ mengalokasikan komponent,  $H$ ,  $B$ ,  $D$ ,  $X$  dengan meminimumkan/ meminimalkan volume oli pada saat tuas (lever) piston diangkat dari sebagaimana digambarkan pada gambar 5. Kendala/ constraint yang diberikan pada saat melakukan optimisasi adalah kesetimbangan gaya, momen lentur maksimum tuas, stroke piston minimum dan kondisi geometris. Dalam penelitian ini hasil terbaik didapatkan pada dengan . Hasil yang didapatkan pada penelitian ini kemudian dibandingkan dengan penelitian yang telah dilakukan oleh (Bayzidi dkk. 2021), (Hamzehkolaei dkk. 2021) dan (Gandomi dkk. 2013). (Bayzidi dkk. 2021) mendapatkan nilai optimal untuk best case sedangkan untuk worst case metode yang diusulkan mendapatkan nilai optimal , (Gandomi dkk. 2013) dengan memanfaatkan metode Cuckoo search hanya mampu mendapatkan nilai optimal .4271 dengan performa terburuk , sedangkan (Hamzehkolaei dkk. 2021) harus mendefenisikan kembali range desain saat melakukan optimisasi agar dapat menghasilkan nilai optimal dalam menyelesaikan permasalahan piston lever, hal ini tentunya membuat metode yang diusulkan oleh (Hamzehkolaei dkk. 2021) memenuhi nilai optimal namun melanggar kendala pada model matematis yang dikaji pada penelitian ini . Jika dibandingkan dari hasil penelitian yang disebutkan diatas dan hasil perbandingan dari table 1, metode yang diusulkan pada penelitian ini sangat mampu mengungguli hasil optimal yang didapatkan oleh (Gandomi dkk. 2013), (Hamzehkolaei dkk. 2021) dan (Bayzidi dkk. 2021). Selain berhasil konvergen menuju solusi optimal, dari gambar 2 juga memperlihatkan bahwa metode yang diusulkan pada penelitian ini berhasil konvergen hanya dalam 240 iterasi dengan waktu komputasi yang sangat efisien menuju solusi optimal seperti yang diusulkan pada penelitian (Bayzidi dkk. 2021).



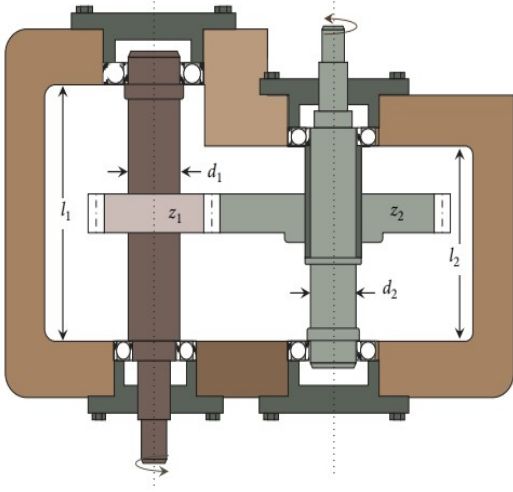
## 4.3 Benchmark - Welded Beam



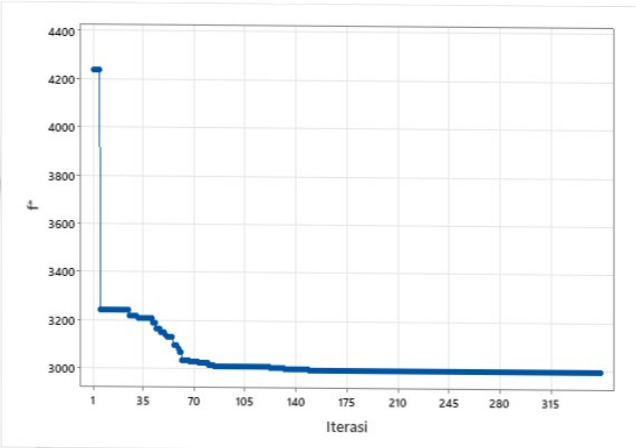
**Gambar 3.** menunjukkan struktur dari welded beam, permasalahan pada gambar 3. adalah meminimumkan biaya pabrikan (fabrication) dari sebuah welded beam, dengan tujuh kendala, yang terdiri dari dua kendala bersifat linear dan lima kendala nonlinear yang masing-masing adalah shear stress, bending stress, buckling load dan deflection serta range desain dari welded beam adalah . (Ferreira dkk. 2018) menyebutkan bahwa solusi optimal yang pernah ditemukan adalah (Ferreira dkk. 2018) juga membandingkan hasil yang didapatkan dengan berbagai metode yang ada seperti PSO-DE, HPSO, UPSO, MBA, CMA-ES, MVDE dimana hasil yang didapatkan lebih baik dari beberapa metode tersebut, (Radaideh dkk. 2021) memanfaatkan Rule-based reinforcement learning pada metode PSO, DE, dengan masing-masing hasil optimal yang didapatkan adalah **1.75005**, **1.72755**, (coello. 2000) memberikan hasil optimal pada permasalahan welded beam dengan **1.7858346524**, (Eskandar dll. 2012) memanfaatkan metode WCA untuk menyelesaikan permasalahan welded beam dengan nilai optimal 1.724856, dalam penelitian ini metode yang diusulkan memberikan hasil yangungguli metode-metode yang telah disebutkan diatas kecuali pada metode yang diusulkan oleh (Ferreira dkk. 2018). Namun (Ferreira dkk. 2018) tidak menyebutkan bahwa metode tersebut memiliki laju konvergensi yang sangat cepat dalam menemukan nilai optimal dari permasalahan welded beam, sementara dari **gambar 4.** terlihat bahwa metode yang diusulkan dalam penelitian ini sangat cepat konvergen menuju solusi optimal pada permasalahan welded beam, yaitu dengan nilai optimal 1.7248535, dengan . **Table 2.** merangkum hasil performa terbaik, performa rata-rata dan performa terburuk dari metode yang disebutkan diatas beserta perbandingannya pada metode yang diusulkan dalam penelitian ini .



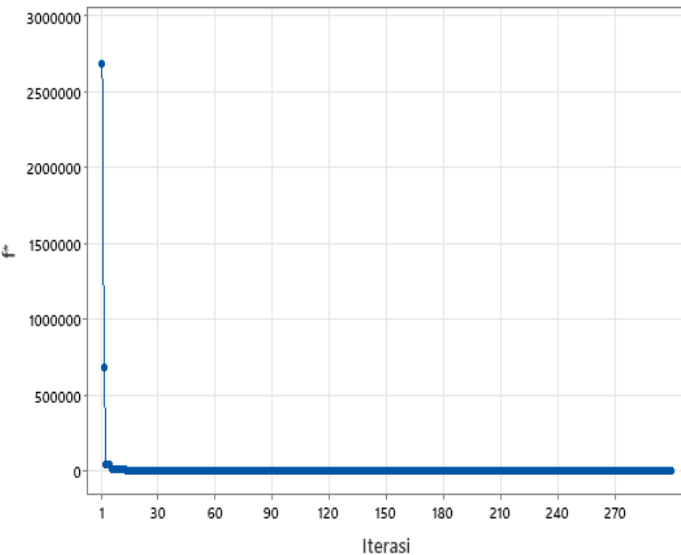
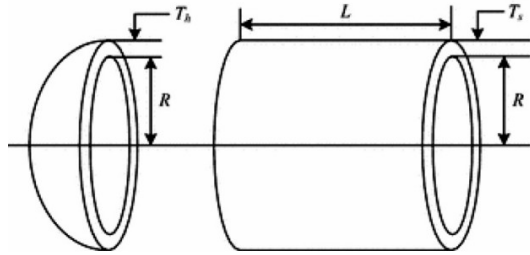
## 4.4 Benchmark - Speed Reducer



Permasalahan speed reducer pertama kali diusulkan oleh (Golinski 1973), tujuan utama dari permasalahan ini adalah meminimumkan bobot dari sebuah speed reducer. **Gambar 5.** menunjukkan struktur dari sebuah speed reducer, ketujuh variable digunakan untuk mengoptimalkan bobot dari sebuah speed reducer, dengan sebelas kendala yang perlu dipertimbangkan untuk melakukan optimisasi yang mana masing-masing kendala bersifat nonlinear. Untuk mendapatkan bobot optimal dari permasalahan speed reducer, (Radaideh dkk. 2021) mengusulkan konsep Rule-based reinforcement learning pada metode PSO, DE, NES, dengan bobot optimal yang didapatkan menggunakan RL-PSO adalah 2996.85284 dan RL-NES 3000.15554, (Ferreira dkk. 2018) memanfaatkan metode C-ITGO dalam menyelesaikan permasalahan speed reducer dengan bobot optimal yang didapatkan adalah 2996.85284, metode yang diusulkan oleh (Ferreira dkk. 2018) melakukan sebesar 856,400 *function call*, (Akay dkk. 2012) melakukan optimisasi pada permasalahan speed reducer dengan memanfaatkan metode Artificial bee colony algorithm, bobot optimal yang didapatkan oleh (Akay dkk. 2012) adalah 2997.058412, (Shabani dkk. 2020) mengusulkan sebuah metode Search and rescue optimization algorithm dalam menyelesaikan masalah optimasi khususnya dalam menyelesaikan permasalahan speed reducer, bobot optimal yang didapatkan oleh (Shabani dkk. 2020) adalah 2994.471066 dengan 22,000 *function call*, (Zhang dkk. 2021) memanfaatkan metode Enhanced Jaya algorithm dalam menyelesaikan permasalahan speed reducer, metode yang diusulkan mampu mendapatkan bobot yang sangat optimal yaitu 2994.471066 dengan 17,000 *function call*. Dalam penelitian ini, metode yang diusulkan mampu mengungguli bobot optimal yang didapatkan oleh (Radaideh dkk. 2021), (Ferreira dkk. 2018) dan (Akay dkk. 2012) dalam menyelesaikan permasalahan speed reducer dan juga mampu mendekati bobot optimal yang didapatkan pada penelitian (Zhang dkk. 2021) dan (Shabani dkk. 2020) dengan laju konvergensi yang sangat cepat. Dari **Gambar 6.** juga terlihat bahwa metode yang diusulkan dalam penelitian ini mampu konvergen sangat cepat mendekati bobot optimal seperti yang diusulkan oleh (Ferreira dkk. 2018) dan (Akay dkk. 2012) dengan hanya 200 -300 iterasi. Table 1. merangkum performa terbaik, terburuk dan performa rata-rata dari metode yang diusulkan para peneliti dalam menyelesaikan permasalahan speed reducer.

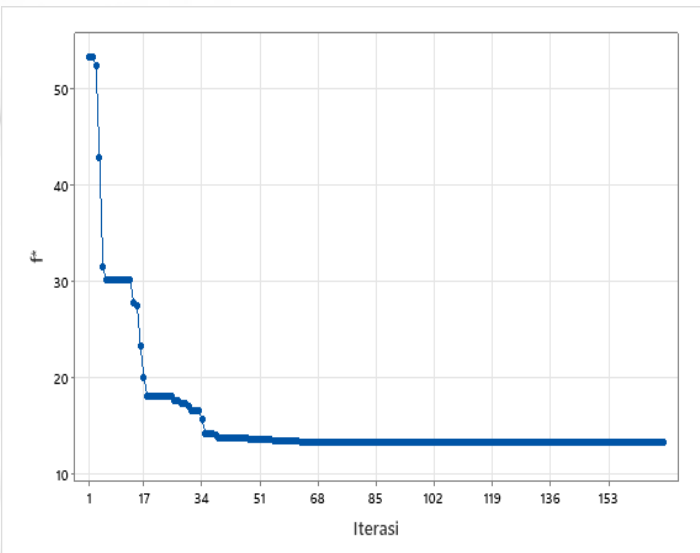
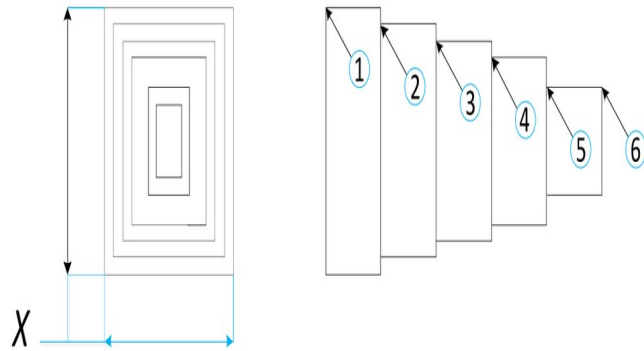


## 4.5 Benchmark - Pressure vessel



**Gambar 2** menunjukkan struktur dari Pressure vessel, pada permasalahan Pressure Vessel, fungsi objektif yang diberikan bertujuan untuk mengoptimalkan total biaya manufacturing/ manufacturing, termasuk biaya material, biaya pembentukan dan pengelasan. Permasalahan ini terdiri dari 3 kendala linear dan sebuah kendala nonlinear serta terdiri dari 4 variabel  $T_s$ ,  $T_h$ ,  $R$ ,  $L$  yang masing-masing merupakan ketebalan shell, ketebalan head, inner radius dan Panjang bagian silinder. (Ferreira dkk. 2018) menyebutkan bahwa merupakan hasil optimal dari metode yang diusulkan, dengan 9 parameter yang perlu dipertimbangkan dalam menyelesaikan permasalahan Pressure Vessel, (He dkk. 2008) mendapatkan hasil optimal dengan memanfaatkan metode CPSO, dengan melakukan pengujian sebanyak 30 kali (Machado-Coelho dkk. 2017) mendapatkan hasil optimal 6059.7143, (Melo dkk. 2012) memanfaatkan metode DE untuk menyelesaikan permasalahan ini dengan mendapatkan nilai optimal = 6059.71433504844, (Guedria. 2015) melakukan improvisasi pada algoritma PSO dan mengimplementasikan metode tersebut pada permasalahan ini dengan mendapatkan nilai optimal = 6090.5314. Sementara, pada penelitian ini hasil optimal yang didapatkan adalah **5952.9974**, hasil yang didapatkan mengungguli semua metode yang disebutkan diatas, **Gambar 3** juga menunjukkan metode yang diusulkan untuk menyelesaikan permasalahan ini hanya memerlukan 30-50 iterasi untuk konvergen ke solusi optimal. **Table 2**. merangkum hasil performa terbaik, performa rata-rata dan performa terburuk dari metode yang disebutkan diatas beserta perbandingannya pada metode yang diusulkan dalam penelitian ini. Selain memiliki keakuratan yang sangat baik dalam menyelesaikan permasalahan pressure vessel, dari data statistik yang didapatkan, terlihat bahwa metode yang diusulkan oleh (Ferreira dkk. 2018), (Guedria. 2015), (Machado-Coelho dkk. 2017) tidak mampu memberikan performa optimal selama pengujian yang berulang-ulang pada metode yang telah diusulkan dalam menyelesaikan permasalahan Pressure Vessel.

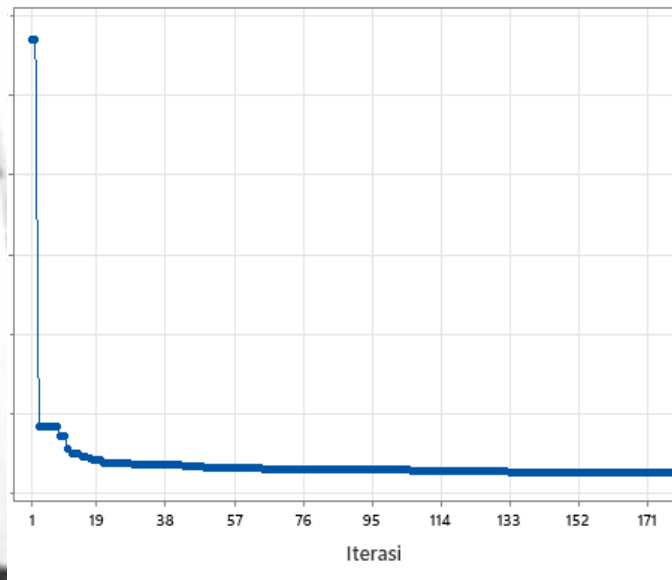
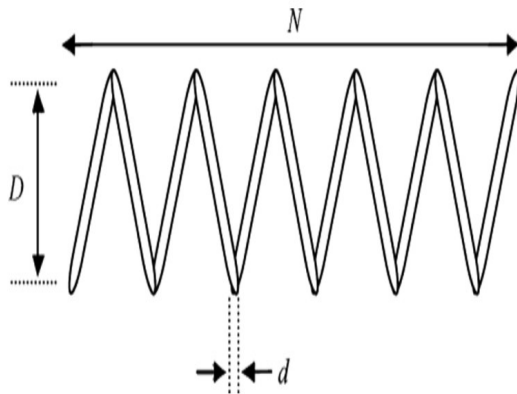
## 4.6 Benchmark - cantilever beam



**Gambar 2** menunjukkan struktur dari *cantilever beam*, yang dibentuk oleh lima blok berongga persegi. Permasalahan yang diberikan bertujuan untuk meminimumkan atau mengoptimalkan bobot dari sebuah cantilever beam. Pada permasalahan ini diberikan 5 variable ( dan sebuah kendala/ constraint dalam menentukan bobot yang optimal dari permasalahan yang diberikan, serta range dari variable saat melakukan optimisasi  $0.01 \leq \leq 100$ ,  $0.01 \leq \leq 100$ ,  $0.01 \leq \leq 100$ ,  $0.01 \leq \leq 100$  dan  $0.01 \leq \leq 100$ . Permasalahan ini telah dikaji dalam banyak literatur, (Cheng dkk. 2019) mendapatkan bobot yang sangat optimal pada permasalahan ini dengan **bobot = 13.032514**, namun kendala/ constraint hanya memiliki ketelitian 1 digit yaitu **0.05**, (Mirjalili dkk,2016) mendapatkan bobot optimal **bobot = 13.399595**, (cheng dkk.2014) mendapatkan bobot optimal **bobot = 13.3996**, (Gandomi dkk.2011) mendapatkan bobot optimal **bobot =13.3999**, sedangkan metode yang diusulkan dalam penelitian ini mampu memberikan bobot yang optimal yaitu bobot = **13.3652** dan juga memenuhi kendala/ constraint yang diberikan pada permasalahan yang ada. Dalam 50 iterasi metode yang diusulkan juga mampu konvergen dengan sangat cepat, hal ini diilustrasikan pada **gambar 6**. Jika dibandingkan dengan metode yang diusulkan oleh (Cheng dkk. 2019), (Mirjalili dkk,2016), (cheng dkk.2014), terlihat bahwa metode yang diusulkan dalam penelitian ini mampu memberikan hasil atau bobot yang optimal serta memenuhi kendala/ constraint yang ada. Performa terburuk serta performa rata-rata pada pengujian yang dilakukan juga menunjukkan metode yang diusulkan dalam penelitian ini tidak begitu sensitive dengan nilai random yang dibangkitkan selama 20 kali pengujian. **Table 1** merupakan perbandingan hasil dari penelitian yang pernah dilakukan pada permasalahan cantilever beam.

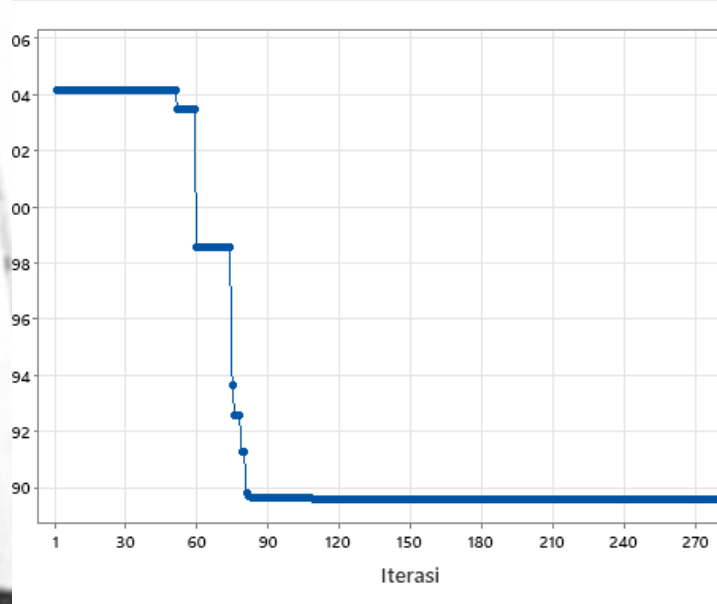
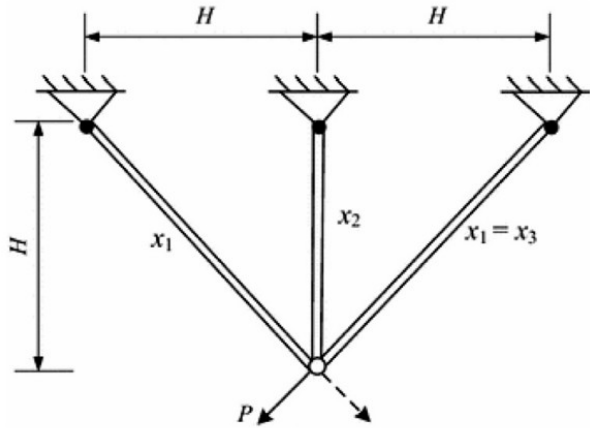


## 4.7 Benchmark - Tension/ Compression



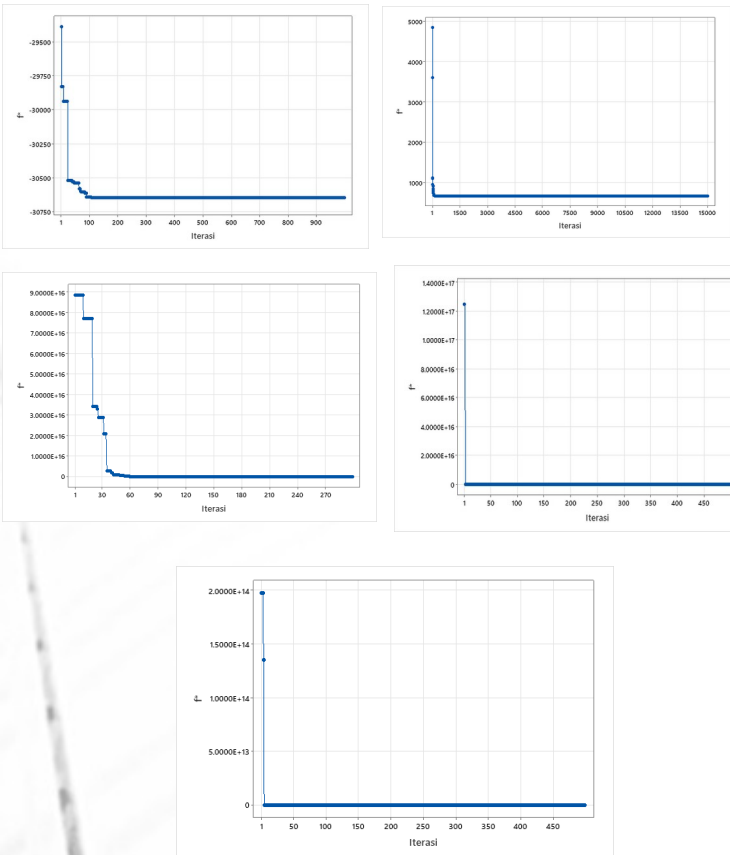
**Gambar 5.** menunjukkan struktur dari sebuah tension/ compression spring, permasalahan tension/ compression spring merupakan salah satu permasalahan yang ada pada bidang engineering, permasalahan ini dapat dimodelkan sebagai suatu permasalahan optimasi terkendala. Pada permasalahan tension/ compression design, fungsi objektif merupakan bobot dari sebuah tension/ compression yang perlu dioptimisasi (talatahari dkk.2020) dengan , merupakan kendala yang diberikan pada permasalahan tension/ compression design dimana keempat kendala tersebut berbentuk pertidaksamaan serta yang masing-masing merupakan diameter dari wire ( $d$ ), rata-rata diameter dari coil ( $D$ ) dan jumlah dari active coil ( $N$ ). Hasil dari metode yang pernah dilakukan oleh para peneliti dengan memanfaatkan metode metaheuristik untuk permasalahan tension/ compression design dirangkum dalam **table 3**. Dari table tersebut terlihat bahwa (He dkk. 2006) menggunakan metode co-evolutionary particle swarm optimization untuk menyelesaikan permasalahan tension/ compression design dengan nilai optimal yang didapatkan adalah dengan , (talatahari dkk. 2020) juga memanfaatkan metode chaos game optimization untuk penyelesaian permasalahan tension/ compression design dengan mendapatkan bobot optimal = **0.012665246** dengan , 11.3265749), (Coello dkk. 2002) menggunakan metode GA untuk menyelesaikan permasalahan tension/ compression dengan bobot optimal sebesar = 0.012681 dengan , 0.363965, 10.890522), (Chen dkk. 2020) melakukan modifikasi pada metode sine cosine algorithm untuk menyelesaikan permasalahan tension/ compression design dengan bobot optimal yaitu dengan ), sedangkan dari table tersebut terlihat bahwa metode yang diusulkan dalam penelitian ini mampu mendapatkan bobot optimal yaitu dengan . Bobot optimal yang didapatkan pada penelitian ini mampu mengungguli hasil optimal yang diusulkan oleh (Coello dkk. 2002), dan jika dibandingkan pada metode yang diusulkan oleh (Chen dkk. 2020) serta (talatahari dkk. 2020), metode yang diusulkan dalam penelitian ini lebih mudah diimplementasikan karena tidak memerlukan parameter yang begitu banyak dalam melakukan optimisasi pada permasalahan tension/ compression design. **Gambar 5** juga menunjukkan bahwa metode yang diusulkan dalam penelitian ini mampu konvergen dengan sangat cepat mendekati bobot optimal seperti yang disebutkan oleh (talatahari dkk.2020).

## 4.8 Benchmark - Three-bar truss



**Gambar 4.** three-bar truss design merupakan permasalahan yang bertujuan mengoptimalkan volume dari sebuah three-bar truss yang dimuat secara statis. Permasalahan yang diberikan terdiri dari 2 variable bersifat kontiniu (*continuous*) dan terdiri dari tiga kendala berupa pertidaksamaan bersifat nonlinear dengan range design terletak diantara Hasil terbaik yang disebutkan dalam beberapa literatur adalah (Liu dkk.2010), 0.788675134746, 0.408248290037 dengan 17.600 evaluasi terhadap fungsi objektif yang diberikan, (Melo dkk. 2013) memberikan hasil optimal =263.89584337 dengan standar deviasi sebesar 2.576062e-7 dan variable optimal untuk permasalahan Three-bar truss adalah , (Saddollah dkk. 2013) memanfaatkan algoritma MBA untuk menyelesaikan permasalahan three-bar truss dan memberikan hasil optimal untuk performa terbaik adalah 263.895852 dan performa terburuk yang didapatkan selama percobaan yang berulang-ulang pada algoritma MBA (Saddollah dkk. 2013) adalah dengan iterasi sebanyak 500 kali, 0.7886751, , dan standar deviasi sebesar = 3.93E-03. Dalam penelitian ini metode yang diusulkan mampu memberikan laju konvergensi yang sangat cepat pada nilai optimal yang disebutkan dalam (Saddollah dkk. 2013), (Liu dkk.2010), (Melo dkk. 2013). Gambar 3. menunjukan laju konvergensi pada metode yang diusulkan dengan nilai optimal , terlihat bahwa hanya dalam 100 iterasi, metode yang diusulkan dalam penelitian ini konvergen dengan ketelitian sampai empat digit pada metode yang diusulkan oleh (Melo dkk. 2013) dan lima kali lebih cepat dari algoritma yang diusulkan oleh (Saddollah dkk. 2013) pada permasalahan three-bar truss.

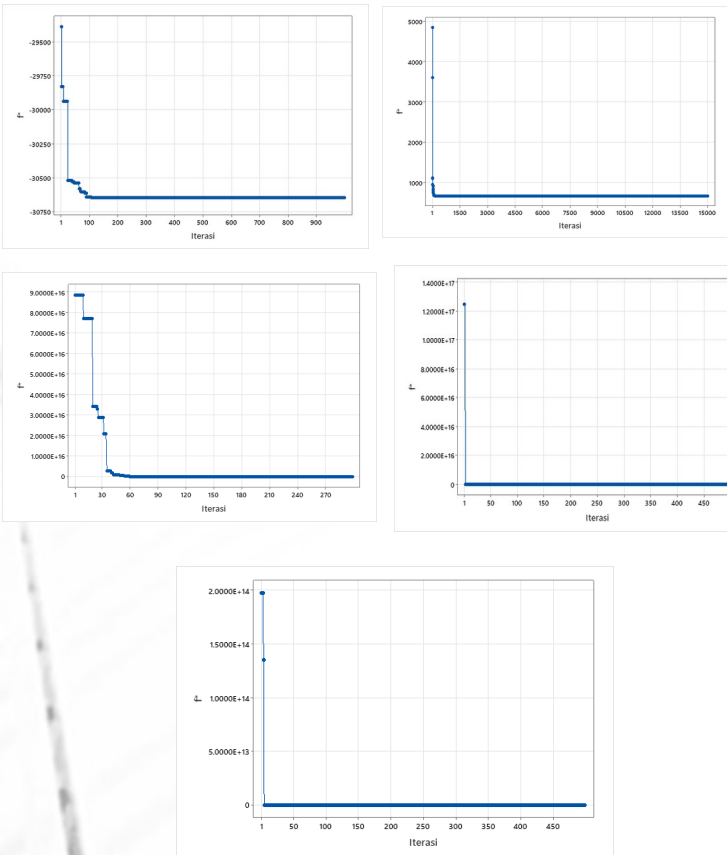
# Conclusions



Pada penelitian ini, kami mengusulkan metode NNA untuk menyelesaikan permasalahan optimasi berkendala, penyelesaian sejumlah permasalahan optimasi berkendala yang ditemukan dalam bidang engineering dan sejumlah fungsi benchmark berdimensi besar. Pada fungsi berkendala, diperlukan konstruksi ulang agar fungsi tersebut menjadi fungsi tidak berkendala. Kami telah mengkonfirmasi/ melakukan validasi terhadap metode yang kami usulkan dalam menyelesaikan sejumlah permasalahan yang ditemukan dari berbagai literatur. Hasil yang didapatkan telah dibandingkan dengan berbagai metode seperti Spiral Dynamic Optimization Algorithm (Tamura dkk. 2011), Social Network Search ( Bayzidi dkk. 2021), Hybrid multi-level cross entropy moth-flame (Hamzekolaei dkk. 2021), Cukoo Search Algorithm ( Gandomi dkk. 2013), Rule based reinforcement learning ( Radaideh dkk. 2021), WCA (Eskandar dkk. 2012), Artificial bee colony algorithm (Akay dkk. 2012), IPSO ( Guedria 2015), Search and rescue optimization algorithm (Shabani dkk.2020), Chaos game optimization ( Talatahari dkk. 2020 ), Mine blast algorithm (Sadollah dkk. 2013), metode yang kami usulkan untuk menyelesaikan permasalahan optimasi berkendala dan fungsi benchmark berdimensi besar tidak hanya menghasilkan keakuratan yang sangat baik namun juga memiliki laju konvergensi serta implementasi yang sederhana dibandingkan dengan metode-metode yang disebutkan diatas. Ada beberapa hal yang perlu dipertimbangkan dalam melakukan/ menyelesaikan setiap permasalahan yang diusulkan pada penelitian ini, pada kasus fungsi berdimensi besar dan multimodal sebaiknya nilai NPOP dibuat dalam rentang/ range (100-140) dan maksimum iterasi MAX = (1000-1500), sedangkan pada fungsi yang memiliki kendala, banyaknya kendala yang ada dan sifat nonlinear dari fungsi tersebut juga dapat dijadikan pertimbangan dalam memilih NPOP dan MAX. Disisi lain kerandoman bilangan juga perlu dipertimbangkan, dalam penelitian ini pemilihan bilangan random menggunakan standart library yang ada pada bahasa pemrograman c++ yang menjamin bahwa bilangan yang dibangkitkan pada setiap proses optimisasi bersifat acak.



# To do : bar chart compare Each Algorithm



Pada penelitian ini, kami mengusulkan metode NNA untuk menyelesaikan permasalahan optimasi berkendala, penyelesaian sejumlah permasalahan