Contents lists available at JESA

Journal of Engineering and Science Application

journal homepage: https://jesa.aks.or.id/index.php/jesa/index

Particle Swarm Optimization and Genetic Algorithms for Efficient Fertilizer and Pesticide Use

Alwi Baharudin[a], Shelvi Eka Tassia[b]

[a]*Informatics Engineering Study Program, Universitas Pamulang, South Tangerang, Banten, Indonesia*

Article Info

Keywords:

Farmland Management
Particle Swarm Optimization
Genetic Algorithm
Fertilizer
Pesticide

ABSTRACT

Farmers in Cibaregbeg Village, Cianjur Regency, generally determine fertilizer and pesticide doses from habit, shop-owner advice, or package labels, ignoring field variables such as soil pH, rice variety, and pest type. This study designs and builds a web-based farmland management application that recommends fertilizer and pesticide types and doses for rice. Particle Swarm Optimization (PSO) determines fertilizer doses as a continuous numerical problem, while a Genetic Algorithm (GA) selects pesticides as a discrete combinatorial problem. The application was developed with the Agile method and evaluated through black box, computation, and database integrity testing. All 37 black box scenarios across eight modules produced the expected results. For a 1-hectare plot of the Ciherang variety with grasshopper infestation and a soil pH of 6.5, PSO produced a combination of Urea, NPK Phonska, and KCl supplying 120 kg N, 36 kg P, and 66 kg K, meeting the variety requirements (120, 36, and 60 kg/ha), while GA selected Decis 25 EC at 500 ml/ha with a fitness value of 1.0. Recommendations were stored consistently in a relational database. The application is functionally feasible as a decision-support tool for farmers; field validation of efficiency and yield remains future work.

This is an open access article under the CC BY license.

Corresponding Author:

Alwi Baharudin
Informatics Engineering Study Program
Universitas Pamulang
South Tangerang, Banten, Indonesia
Email: alwibaharudin15@gmail.com

1. INTRODUCTION

Cibaregbeg Village in Cibeber District, Cianjur Regency, West Java, is known as one of the rice-producing areas of Cianjur, and most of its residents work as farmers and farm laborers. However, the quality and quantity of rice production in the village tend to fluctuate and decline. One cause is the limited literacy and accurate information available to farmers, especially novice farmers, in managing their land. Fertilizer and pesticide doses are generally determined by habit, advice from farm-shop owners, or package labels, without considering field variables such as soil pH, crop variety, and pest type. Imprecise use risks lowering yields, wasting operational costs, and, in the long term, depleting soil nutrients and damaging the agricultural ecosystem. Improper pesticide use can also pollute soil and water [1].

Determining the ideal combination of fertilizers and pesticides is essentially a multivariable optimization problem that calls for an intelligent computing approach. Fertilizers and pesticides pose different kinds of problems. Determining fertilizer doses, for example the kilograms of N, P, and K, is a

continuous numerical problem, whereas selecting a pesticide for a particular pest is a discrete combinatorial selection problem. *Particle Swarm Optimization* (PSO) converges quickly with relatively few parameters [2], but is less able to explore a wider solution space [3]. In contrast, the *Genetic Algorithm* (GA) can handle complex problems with a very large search space through selection, *crossover*, and mutation, although it requires many generations to reach the optimal value [3]. These complementary characteristics underlie the division of roles in this study, namely PSO for fertilizer types and doses and GA for pesticide selection.

Several studies have applied PSO and GA in agriculture and other optimization fields. Budiman and Parama Yoga [4] applied a PSO-based K-Means algorithm to cluster regions in Indonesia by vegetable crop production level into three clusters. Violina *et al.* [5] optimized three parameters of *Triple Exponential Smoothing* forecasting using GA to predict rice harvests in the Madura region and obtained a *Mean Absolute Percentage Error* (MAPE) of 9.74%. Dika Prasisti and Nugroho [6] compared PSO and GA on production scheduling; PSO produced a *makespan* of 273 minutes, whereas GA produced 281 minutes. For field monitoring, Hamidah and Ismaya [7] applied *smart farming* based on the *Internet of Things* (IoT) in Karawang, which shifted farming from intuition-based to data-based practice and increased average yields by 13–17% per hectare. These studies differ from the present one in commodity, object, or system output, and none of them produces an integrated recommendation of fertilizer and pesticide types and doses from farmers' field parameters. The novelty of this study lies in combining PSO for fertilizer and GA for pesticide in a single web-based farmland management application that also calculates estimated cost and yield potential, with a case study of rice fields in Cibaregbeg Village.

The objectives of this study are (1) to determine fertilizer and pesticide types and doses suited to rice field conditions in Cibaregbeg Village based on field parameters and crop conditions; (2) to apply a combination of PSO and GA to produce accurate and efficient fertilizer and pesticide recommendations; and (3) to design, build, and test a farmland management application that helps farmers determine fertilizer and pesticide types and doses. This manuscript is the authors' original research, developed from the first author's undergraduate thesis using the authors' own ideas.

2. RESEARCH METHOD

2.1. Data Collection and Development Stages

Data and system requirements were collected in three ways. First, structured interviews with local farmers elicited system requirements, including historical data, pest characteristics, and soil quality standards. Second, direct observation in Cibaregbeg Village examined field conditions, planting methods, and prevailing patterns of fertilizer and pesticide use. Third, a literature study covered scientific journals, books, and other publications on PSO, GA, precision agriculture, and software engineering.

The application was developed using the *Agile* approach, an iterative and incremental software development approach that involves collaboration between developers and stakeholders [8]. This approach was chosen because it allows the system to adapt to changes or additions in user requirements during development [9]. The seven stages used are shown in Figure 1 and detailed in Table 1.



Figure 1. Agile development stages of the system.

Table 1. System development stages and research activities

Stage	Research activity
Planning	Identification of user requirements, literature review, determination of the main feature (recommendation calculation module), and development roadmap.
Design	Design of a farmer-friendly user interface (UI/UX), system architecture, and PSO and GA integration logic using UML modeling and flowcharts.
Develop	Program code implementation, PSO and GA module integration, and construction of the database of field variables, soil pH, pests, and user history.
Test	Internal testing of application functions and of PSO and GA computational accuracy using simulated field data samples.
Deploy	Limited deployment to farmer group representatives to observe user interaction and response.
Review	Collection of user feedback and iteration back to the Planning or Design stage when necessary.
Launch	Release of the final version accompanied by outreach and provision of a user guide.

The results reported in this article cover the Develop and Test stages, namely application implementation and internal testing of the functions and algorithm computations.

2.2. Research Data and Scope

The study is limited to rice with seven varieties, namely Ciherang, Sintanur, Inpari 32, Mekongga, Cibatu 06, Pandan Wangi, and IR64. The recommended fertilizers are limited to Urea, NPK Kujang, NPK Mutiara 16-16-16, NPK Phonska, and KCl. The pesticides are limited to Furadan 3GR, Spontan 400 SL, Regent 50SC, Niclosamide, Decis 25 EC, and raw cassava, targeting planthoppers, golden apple snails, grasshoppers, and rats [10]. The system output is restricted to technical farmland management recommendations, so it excludes financial management, recording of harvest sales transactions, and supply chain analysis.

The variety master data store the suitable pH range, harvest age, yield potential, and N, P, and K requirements per hectare. The fertilizer master data store nutrient content, price, minimum and maximum dose limits, and the suitable pH range (Table 2). The pesticide master data store the active ingredient, application method, price, unit, target pest, effectiveness, suitable weather conditions, and application cost.

Table 2. Fertilizer nutrient content in the master data

Fertilizer name	N (%)	P (%)	K (%)	pH range
Urea	46	0	0	5.5–7
NPK Mutiara 16-16-16	16	16	16	5.5–7
NPK Kujang	15	15	15	5.5–7
NPK Phonska	15	15	15	5.5–7
KCl	0	0	60	–

2.3. System Analysis and Workflow

In the current system, farmers observe crop and pest conditions directly, then determine fertilizer and pesticide needs based on observation, habit, advice from farm-shop owners, or package labels. Field data and records of agricultural input use are not managed in an integrated way, which makes monitoring and evaluation difficult. The proposed system is a web application whose workflow is shown in Figure 2: the farmer logs in and enters field data and farming parameters (field name, field area, rice variety, soil pH, and pest type); the system then optimizes fertilizer types and doses with PSO and pesticide types and doses with GA, displays the recommendation with an estimated cost, and saves the result as a history for evaluating harvest results.

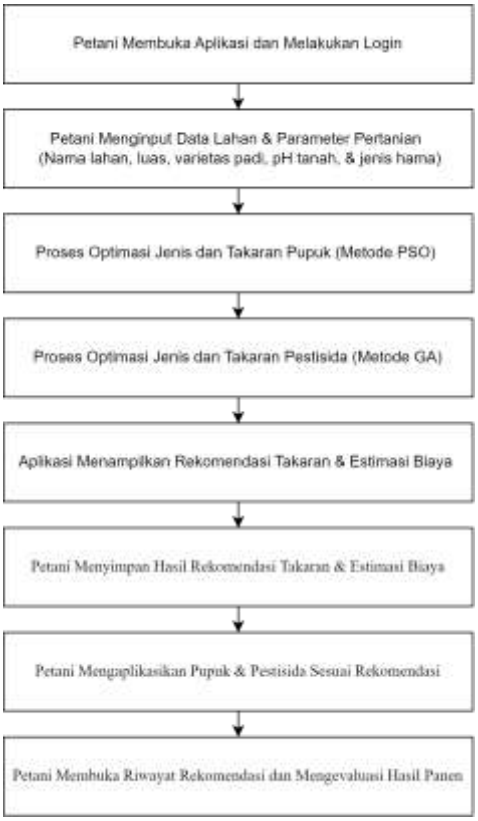


Figure 2. Workflow of the proposed system.

2.4. Particle Swarm Optimization for Fertilizer

PSO is inspired by the behavior of a flock of birds searching for food. Each particle represents one candidate combination of fertilizer doses and moves through the search space guided by two references: the best position reached by the particle itself (*pbest*) and the best position of the whole swarm (*gbest*). At each iteration, the velocity and position of the particle are updated using the following equations.

$$v_i^{t+1} = w \cdot v_i^t + c_1 \cdot r_1 \cdot (pbest_i - x_i^t) + c_2 \cdot r_2 \cdot (gbest - x_i^t) \quad (1)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (2)$$

where w is the inertia weight, c_1 and c_2 are the cognitive and social learning factors, and r_1 and r_2 are random numbers in the range 0–1. The inertia component maintains the previous direction of movement, the cognitive component directs the particle toward its own best experience, and the social component directs it toward the best solution of the population. The algorithm steps are determining the number of particles, initializing random positions and velocities, calculating *fitness*, determining *pbest* and *gbest*, updating velocity and position, and repeating until the maximum number of iterations or the stopping criterion is met. In this study the objective function is designed so that the recommendation matches the field requirements, uses fertilizer efficiently, and stays within the specified dose limits.

The nutrient supply from the recommendation is calculated from the nutrient content of each fertilizer using the following equation, where n_{jk} is the percentage content of nutrient k (N, P, or K) in fertilizer j and d_j is the dose of fertilizer j in kilograms.

$$S_k = \sum_j (n_{jk} \times d_j) \quad (3)$$

The parameters used in the computation test are inertia weight w = 0.5, $c_1 = c_2$ = 2.0, and random numbers r_1 = 0.6 and r_2 = 0.4. The values $c_1 = c_2$ = 2 and w in the range 0.4–0.9 are commonly used in PSO studies. The *gbest* target for nitrogen is set to 120 kg according to the requirement of the Ciherang variety.

2.5. Genetic Algorithm for Pesticide

GA imitates the process of biological evolution [11]. An initial population of solutions is generated randomly, each solution is evaluated with a *fitness* value, and the best solutions are then chosen through selection, *crossover*, and mutation. Chromosomes can be represented as binary, integer, or real numbers depending on the characteristics of the problem [12]. In this study the pesticide index in the database is

encoded as a 6-bit binary chromosome. Chromosome quality is evaluated with a *fitness* function that minimizes the penalty for mismatch with the target pest and the penalty for violating the dose limit.

$$\text{Fitness} = \frac{1}{1+(\text{Pest Penalty} + \text{Dose Limit Penalty})} \quad (4)$$

Parent selection uses *Roulette Wheel Selection* [13], with the probability of selecting chromosome *i* given by Equation (5). The higher the *fitness* value, the higher the probability that the chromosome is chosen as a parent.

$$P_i = \frac{\text{Fitness}_i}{\sum_{j=1}^{n} \text{Fitness}_j} \quad (5)$$

Recombination uses *single-point crossover* with a rate of P_c = 0.8, while mutation uses *bit-string mutation* with a rate of P_m = 0.1. Mutation keeps the system from stopping at a local solution that looks good early on and keeps it searching for other, more effective and safer combinations.

2.6. Database and Implementation Environment

The application is web-based with two user roles, namely *user* (farmer) and *admin*. The *user* can log in to the system, view the *dashboard*, manage field data, obtain fertilizer and pesticide recommendations, view recommendation history and details, manage the profile, and log out. The *admin* can manage the master data of varieties, pests, fertilizers, pesticides, and fields and view recommendation results and history. Data are stored in a PostgreSQL relational database [14] accessed through Drizzle ORM. The database consists of 10 tables, namely roles, users, user_preferences, varietas (varieties), hama (pests), pupuk (fertilizers), pestisida (pesticides), lahan (fields), rekomendasi (recommendations), and rekomendasi_detail (recommendation details). The rekomendasi table stores relations to the field, variety, and pest together with soil pH, estimated cost, yield potential, algorithm, and status, whereas rekomendasi_detail stores the details of each fertilizer and pesticide in one recommendation. The access rights of the two actors are depicted in the *use case diagram* (Figure 3), and the development and testing environment is summarized in Table 3.

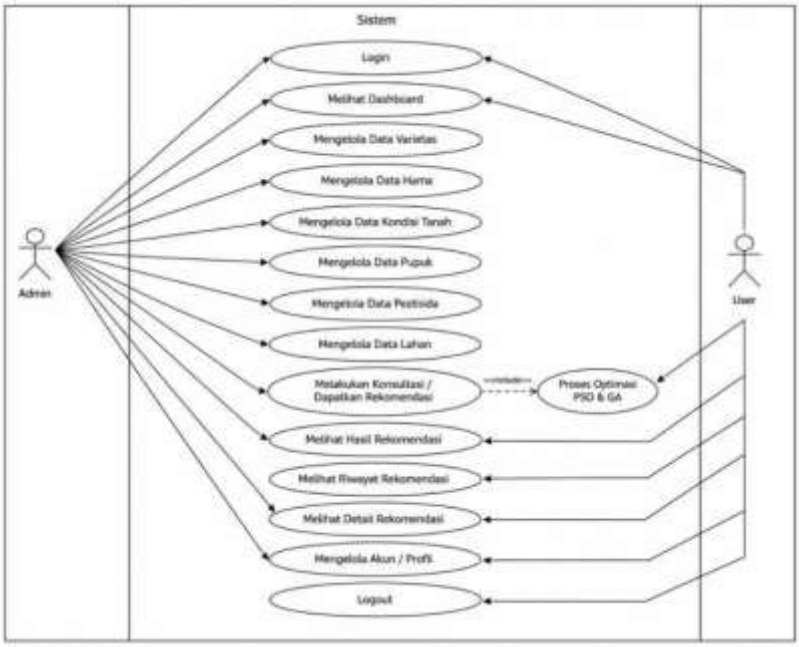


Figure 3. Use case diagram of the application with user and admin actors.

Table 3. Development and testing environment

Component	Specification
Processor	AMD Athlon Silver 3050U (2.3 GHz, up to 3.2 GHz)
Memory and storage	4 GB DDR4; 256 GB NVMe SSD
Operating system	Windows 11 Home
Software	Visual Studio Code, pgAdmin, web browser, Draw.io
Database	PostgreSQL with Drizzle ORM

2.7. Testing Strategy

Testing was carried out in three forms. First, *black box* testing ensured that every function works as required by giving input to each feature and observing the output without looking at the program code [15]. Second, data testing ensured that the field parameter inputs are processed accurately by PSO and GA and that

the results stay within agronomic dose limits. Third, database integrity testing ensured that calculation results are stored intact without decimal truncation or duplication. The data test scenario uses a field named Sawah Luhur of 1.00 ha, the Ciherang variety (standard nutrient requirements N = 120 kg/ha, P = 36 kg/ha, and K = 60 kg/ha), grasshopper as the main pest, and a soil pH of 6.5 (normal).

3. RESULTS AND DISCUSSION

3.1. Application Implementation

The resulting application has eight main modules: login page, user dashboard, recommendation result, recommendation history, recommendation history detail, settings, master data management by the admin, and logout. After the user enters field data, the system displays a recommendation result page containing fertilizer types and doses, pesticide types and doses, estimated cost, and yield potential (Figure 4). In the test case, the system recommended Urea 182.61 kg, NPK Phonska 240 kg, and KCl 50 kg as fertilizers, and Decis 25 EC 500 ml as the pesticide for grasshopper, with an estimated cost of Rp2,660,275 and a yield potential of 8.5 tons. The page also carries an agronomic note that Urea should be applied in stages (*split application*) to improve nitrogen uptake during the vegetative phase. Recommendation results can be saved to the history so that farmers can compare and evaluate input use in the following season.

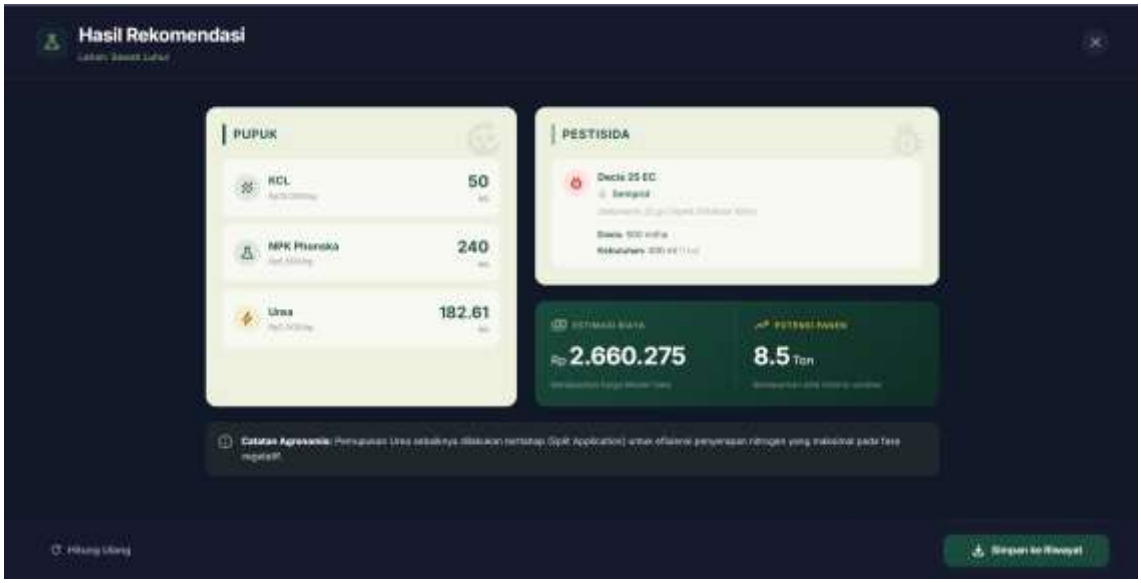


Figure 4. Fertilizer and pesticide recommendation result page.

3.2. Black Box Testing Results

Black box testing was carried out on eight modules with a total of 37 scenarios. All scenarios produced output matching the expected results (Table 4). In the login module, for example, logging in as a user or admin with correct credentials redirected to the respective dashboard, whereas logging in with a wrong username was rejected with an error message.

Table 4. Summary of black box testing results

Module	Function tested	Scenarios	As expected
Login	Authentication of user and admin	6	6
User dashboard	User dashboard display	4	4
Recommendation result	Recommendation processing and presentation	5	5
Recommendation history	History listing, search, and deletion	5	5
History detail	Field data and recommendation result details	5	5
Settings	Account and profile settings	4	4
Admin master data	Master data management by admin	6	6
Logout	Exit from the system	2	2
Total		**37**	**37**

3.3. PSO Computation Testing Results

To illustrate the particle update mechanism, one iteration step was calculated for Urea. The current particle position is assumed to be x = 100 kg, velocity v = 5 kg, and *pbest* = 110 kg, with *gbest* = 120 kg. With the parameters in Subsection 2.4, the new velocity is $v = (0.5 \times 5) + (2.0 \times 0.6 \times (110 - 100)) + (2.0 \times$

$0.4 \times (120 - 100)) = 2.5 + 12 + 16 = 30.5$, so the new position is $x = 100 + 30.5 = 130.5$ kg. This calculation shows that the particle moves toward the nutrient target as pulled by the cognitive and social components.

After the process reached the convergence point, the PSO output was Urea 182.61 kg, NPK Phonska 240.00 kg, and KCl 50.00 kg. The agreement of the result was checked by calculating the pure nutrient supply using Equation (3) and the nutrient content in Table 2 (Urea 46% N; NPK Phonska 15% N, 15% P, and 15% K; KCl 60% K). The results are presented in Table 5.

Table 5. Nutrient supply from the PSO recommendation (1 ha field, Ciherang variety)

Fertilizer	Dose (kg)	N (kg)	P (kg)	K (kg)
Urea	182.61	84.0006	0.00	0.00
NPK Phonska	240.00	36.00	36.00	36.00
KCl	50.00	0.00	0.00	30.00
Total supply	472.61	120.0006	36.00	66.00
Ciherang requirement		120.00	36.00	60.00
Difference		+0.0006	0.00	+6.00

The nitrogen supply of 120.0006 kg meets the minimum requirement of 120 kg/ha of the Ciherang variety, so the fertilizer combination produced by PSO fulfils the nitrogen target. The phosphorus supply of 36 kg exactly equals the requirement of 36 kg/ha, showing that the NPK Phonska dose recommended by the system meets the phosphorus requirement. The potassium supply of 66 kg meets the requirement of 60 kg/ha with a surplus of 6 kg; KCl, with a potassium content of 60%, acts as an additional potassium source that helps the system fulfil the potassium requirement. These results show that PSO converges three fertilizers simultaneously to meet the N, P, and K thresholds of Ciherang rice, rather than merely searching for random values.

3.4. GA Computation Testing Results

In the test case with grasshopper as the pest, the product Decis 25 EC has chromosome 100101. Because this product targets grasshopper and the calculated dose of 500 ml/ha is within the recommended dose limit of 400–500 ml/ha, both the pest penalty and the dose limit penalty equal 0, so *fitness* = 1/(1 + (0 + 0)) = 1.0, the optimal value. The simulation of the genetic operators is shown in Table 6. Single-point *crossover* at gene 3 between Decis 25 EC and Regent 50 SC produces two new offspring, while mutation at bit 2 changes chromosome 100101 into 110101. Through *roulette wheel* selection, the chromosome with the highest pest match is retained until the final generation, and the system outputs 500 ml of Decis 25 EC.

Table 6. Crossover and mutation simulation in the GA

Operation	Input	Parameter	Output
Crossover	100|101 (Decis 25 EC) and 011|010 (Regent 50 SC)	$P_c = 0.8$; cut point at gene 3	100|010 and 011|101
Mutation	100101	$P_m = 0.1$; bit 2 flipped	110101

3.5. Database Integrity Testing Results

Integrity testing ensures that the calculation results shown on the interface are stored intact in the PostgreSQL relational tables without decimal truncation. The results (Table 7) show that the field name and area are stored precisely, that the pH value and yield potential are stored intact and linked to the user through a *foreign key*, and that one recommendation is split into four detail rows (three fertilizers and one pesticide) bound to the same relation ID. All tested entities were consistent.

Table 7. Database integrity testing results

Entity	Condition in PostgreSQL	Status
lahan	nama_lahan = "Sawah Luhur"; luas (area) = 1.00 ha in DECIMAL(10,2) type	Consistent
rekomendasi	ph_tanah = 6.5 and potensi_panen = 8.5 tons stored intact and bound to the user	Consistent
rekomendasi_detail	4 rows: Urea 182.61; NPK Phonska 240.00; KCl 50.00; Decis 500.00 with 1 relation ID	Consistent

3.6. Discussion

The test results show that dividing the algorithms' roles according to the nature of each problem can be implemented in a single application. PSO works in the continuous numerical space of fertilizer doses and produces a combination that meets the nutrient requirements, whereas GA works in the discrete choice space to select a pesticide suited to the target pest. This approach differs from Dika Prasisti and Nugroho [6], who compared PSO and GA on a single scheduling problem; in this study the two algorithms are not compared but placed on different sub-problems within one recommendation. It also differs from Hamidah and Ismaya [7], who emphasize IoT sensor hardware, whereas this study emphasizes the computational processing behind the dashboard, so the two can complement each other if sensor data are later used as system input.

The differences between the current and proposed systems are summarized in Table 8, and the algorithm parameter configuration used in testing is summarized in Table 9.

Table 8. Comparison of the current and proposed systems

Aspect	Current system	Proposed system
Initial process	Farmers observe crop and pest conditions directly	Farmers log in to the application
Data collection	Unstructured	Input of field name, area, variety, soil pH, and pest type
Basis of determination	Habit, farm-shop advice, or package labels	PSO (fertilizer) and GA (pesticide) optimization based on field parameters
Output	Doses based on estimates	Types and doses, estimated cost, and yield potential
Recording	Not yet integrated	Recommendation history stored in the database

Table 9. Algorithm parameters used in testing

Algorithm	Parameter	Value
PSO	Inertia weight (w)	0.5
PSO	Learning factors (c_1, c_2)	2.0
PSO	Random numbers (r_1, r_2)	0.6 and 0.4 (manual calculation)
PSO	Nitrogen *gbest* target	120 kg
GA	Chromosome representation	6-bit binary (pesticide index)
GA	Selection	Roulette wheel
GA	Crossover	Single-point, $P_c = 0.8$
GA	Mutation	Bit-string, $P_m = 0.1$

Including the single fertilizer KCl in the master data lets the system supply potassium separately from phosphorus. NPK Phonska contains equal amounts of N, P, and K, so a dose of 240 kg meets the phosphorus requirement of 36 kg/ha exactly while also supplying 36 kg of potassium; KCl then covers the remaining potassium gap and Urea covers the remaining nitrogen gap. A small potassium surplus of 6 kg (10% of the requirement) remains in the recommendation and could be reduced in further research by refining the objective function or the allowed dose steps.

This study has several limitations. First, the computation test was carried out on only one field scenario, so it does not yet describe algorithm performance across variations of variety, pest, and soil pH. Second, the reduction in input use and the increase in yield in the field have not been measured and compared with farmers' conventional practice, so the efficiency claim in this article is limited to the agreement of the recommendation with nutrient requirements and dose limits. Third, the displayed yield potential is an estimate based on historical variety data, not a prediction of actual yield. Fourth, weather conditions are not yet an input to the current implementation, and the sensitivity of PSO and GA parameter values such as population size and number of iterations has not been analyzed.

4. CONCLUSION

This study successfully designed and built a web-based farmland management application that recommends fertilizer types and doses using PSO and pesticides using GA based on field data, rice variety, pest type, and soil pH, complete with estimated cost and yield potential. All 37 *black box* test scenarios across eight modules produced the expected output. In the test case of a 1-hectare Ciherang field with grasshopper and pH 6.5, the fertilizer recommendation met the N, P, and K nutrient requirements, the pesticide recommendation obtained an optimal *fitness* value of 1.0, and the recommendation results were stored consistently in the database. The application is thus functionally feasible as a decision-support tool for farmers in Cibaregbeg Village in moving from habit-based dosing toward data-based farmland management. The contribution of this study is a design that divides the roles of PSO and GA between numerical and discrete problems in a single agricultural application that can be extended to other commodities and regions. Future research is suggested to (1) compare the performance of PSO and GA with other optimization methods such as *Artificial Bee Colony* and *Ant Colony Optimization* to determine which method gives the most optimal recommendations; (2) add more complete and up-to-date agricultural data, such as weather, soil conditions, varieties, pests, and additional single fertilizers and pesticides; (3) conduct field trials comparing input use and yields between farmers who follow the recommendations and those who use conventional methods; and (4) develop the application further, for example a mobile version, data visualization, and reports of fertilizer, pesticide, cost, and yield use.

REFERENCES

[1] B. R. Sinambela, "Dampak penggunaan pestisida dalam kegiatan pertanian terhadap lingkungan hidup dan kesehatan," (in Indonesian) *Jurnal Agrotek*, vol. 8, no. 2, pp. 178–187, 2024.

[2] S. Serliana, R. Rahman, H. Hastuti, F. Yusuf, and A. M. Abidin, "Penerapan algoritma particle swarm optimization (PSO) pada sistem penjadwalan kegiatan mentoring baca tulis Al-Qur'an," (in Indonesian) *Switch: Jurnal Sains dan Teknologi Informasi*, vol. 3, no. 1, pp. 206–222, 2025, doi: 10.62951/switch.v3i1.337.

[3] M. H. P. Swari, C. A. Putra, and I. P. S. Handika, "Analisis perbandingan algoritma genetika dan modified improved particle swarm optimization dalam penjadwalan mata kuliah," (in Indonesian) *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, vol. 11, no. 2, pp. 92–101, 2022, doi: 10.23887/janapati.v11i2.49061.

[4] Budiman and T. Parama Yoga, "Optimalisasi K-Means berbasis particle swarm optimization untuk hasil produksi tanaman sayuran di Indonesia," (in Indonesian) *Jurnal Nuansa Informatika*, vol. 17, 2023. [Online]. Available: https://journal.uniku.ac.id/index.php/ilkom

[5] D. Violina, D. Nuraini, D. R. Anamisa, B. K. Khotimah, A. Jauhari, and F. A. Mufarroha, "Prediksi panen padi di Madura dengan triple exponential smoothing (TES) dan algoritma genetika," (in Indonesian) *JUSIFOR: Jurnal Sistem Informasi dan Informatika*, vol. 4, no. 1, pp. 9–16, 2025, doi: 10.70609/jusifor.v4i1.5860.

[6] Dika Prasisti and Y. A. Nugroho, "Optimasi penjadwalan produksi untuk meminimalkan makespan dengan pendekatan particle swarm optimization dan genetic algorithm," (in Indonesian) *Jurnal Teknologi dan Manajemen Industri Terapan*, vol. 2, no. 2, pp. 111–118, 2023, doi: 10.55826/tmit.v2i2.134.

[7] E. Hamidah and B. Ismaya, "Penerapan teknologi smart farming dalam meningkatkan produktivitas dan efisiensi usahatani padi di Karawang," (in Indonesian) *SABAJAYA Jurnal Pengabdian Kepada Masyarakat*, vol. 4, no. 03, pp. 105–116, 2026.

[8] N. Nuroji, "Penerapan metode Agile dalam permodelan sistem informasi inventory barang," (in Indonesian) *Journal of Artificial Intelligence and Technology Information (JAITI)*, vol. 1, no. 4, pp. 138–145, 2023, doi: 10.58602/jaiti.v1i4.82.

[9] A. R. Ismail, "Penerapan metode Agile pada perancangan sistem informasi pengajuan nomor surat di pemerintahan desa," (in Indonesian) *Jurnal Teknologi dan Sistem Informasi Bisnis*, vol. 7, no. 2, pp. 284–289, 2025, doi: 10.47233/jteksis.v7i2.1927.

[10] I. Octaviana and S. Ekawati, "Inventarisasi hama dan musuh alami pada tanaman padi di Kecamatan Pulau Laut Timur," (in Indonesian) *Jurnal Pertanian Terpadu*, vol. 10, no. 1, pp. 24–36, 2022, doi: 10.36084/jpt.v10i1.379.

[11] Y. S. Nugraha, U. Darusalam, and A. Iskandar, "Implementasi algoritma genetika pada perancangan aplikasi penjadwalan instalasi antivirus berbasis website menggunakan metode waterfall," (in Indonesian) *Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)*, vol. 6, no. 1, pp. 125–137, 2022, doi: 10.35870/jtik.v6i1.417.

[12] I. Lubis, M. D. Amaliah, and H. Lubis, "Sistem penjadwalan mata pelajaran pada Pondok Pesantren Daarul Istiqlal Medan," (in Indonesian) *Jurnal Krisnadana*, vol. 2, no. 1, pp. 222–232, 2022, doi: 10.58982/krisnadana.v2i1.231.

[13] W. Y. Untoro, T. Widiartin, and N. W. Karyanto, "Replacement of faulty resistors using optimization of parallel-series-parallel configuration based on genetic algorithm," *Jambura Journal of Electrical and Electronics Engineering*, vol. 7, no. 2, pp. 157–166, 2025, doi: 10.37905/jjeee.v7i2.31388.

[14] D. A. B. Prasetyo, "Implementasi information schema database pada PostgreSQL untuk pembuatan tabel informasi dengan menggunakan Python di PT XYZ," (in Indonesian) *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 9, no. 3, pp. 1961–1972, 2022, doi: 10.35957/jatisi.v9i3.2221.

[15] Y. D. Wijaya and M. W. Astuti, "Pengujian blackbox sistem informasi penilaian kinerja karyawan PT Inka (Persero) berbasis equivalence partitions," (in Indonesian) *Jurnal Digital Teknologi Informasi*, vol. 4, no. 1, p. 22, 2021, doi: 10.32502/digital.v4i1.3163.