Integrating Set Theory, Discrete Functions, and Modular Arithmetic for Optimizing Birth Weton-Based Javanese Matchmaking Algorithms

Ghaniyul Amri Caulava - 13525106
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan 10 Ganesha Bandung
E-mail: ghanivivo123@gmail.com, 13525106@std.stei.itb.ac.id

Abstract—Often dismissed as a purely mystical tradition, the Javanese Weton matchmaking and date-selection system is fundamentally driven by a strictly logical luni-solar calendar. By translating this ethnomathematical practice into discrete mathematics, its underlying mechanics can be fully formalized. Specifically, traditional calendar elements are structured as finite sets, and value weightings (*Neptu*) are modeled as surjective mapping functions with an O(1) lookup complexity. Couple compatibility is then evaluated using linear congruence over the quotient ($\mathbb{Z}/8\mathbb{Z}$). Empirical testing confirms this computational model replicates traditional manual calculations with zero error. Furthermore, leveraging the predictable 5-day Pancawara cycle heavily optimizes the search for auspicious wedding dates. This structural regularity cuts the time complexity down from a standard O(n) trial-and-error sweep to an O(1) closed-form calculation for the first valid date, and O(n/5) to generate the complete list. Ultimately, this formalization strips away the pseudoscientific bias, proving the mathematical validity of the Weton system while providing a scalable, efficient foundation for modern software integration.

Index Terms*—*Algorithm Complexity, Discrete Functions, Ethnomathematics, Modular Arithmetic, Set Theory, Weton.

I. INTRODUCTION

Ethnomathematics is an application of mathematical logic that is often used unconsciously by people in everyday life. Although it is likely still viewed as merely a tradition, belief, or cultural practice, it is a cyclical calendar system that guides people in making important life decisions, such as planning harvest times, building a house, and choosing a life partner.

One of the most prominent and historic applications of ethnomathematics in Indonesia is the luni-solar calendar system in Javanese tradition. In the past, this calendar system was used by the community and the palace to formulate various problems. This calendar system is often considered mystical and illogical because its rules, as outlined in the primbon (Javanese calendar), appear dogmatic. However, the way this calendar works is actually very structured. This calendar system operates by simultaneously rotating two cycles: a seven-day cycle (Saptawara) and a five-day market cycle (Pancawara). Each cycle produces a unique combination of times called "Weton".

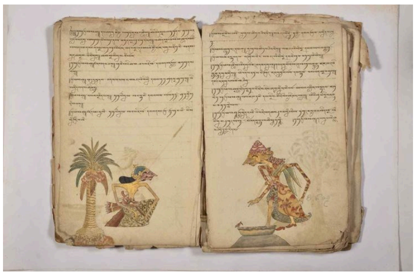

Figure 1. Prambon ilustration (*https://i.pinimg.com/736x/a1/c7/b2/a1c7b2c1c055a61942f628ddbadb2685.jpg*)

Today, Weton calculations are still frequently used in modern society, particularly in determining matchmaking. However, the approach often relies on manual methods and pseudoscientific understanding. This makes it difficult to evaluate the accuracy of these systems.

The process of determining matchmaking in Javanese tradition essentially involves combining the weighted values of each element of the prospective bride and groom's birth date. This compatibility is calculated using a numerical value, or Neptu, mapped for each day and market. Each day typically has a different value, and the Neptu used in matchmaking evaluations is the combined Neptu of the two individuals. The process of mapping values and calculating compatibility in the Weton algorithm can be realized using operations in discrete mathematics, for example, applying Set Theory, Discrete Functions (relations and functions), and Modular Arithmetic(Algorithm complexity).

By integrating Set Theory, Discrete Functions, and Modular Arithmetic, an initially abstract customary formula can be fully formalized. This mathematical integration not only proves that the birth date-based matchmaking algorithm has a precise logical foundation but also paves the way for optimizing such algorithms. Through this formalization, we can design modern computing systems capable of processing matchmaking and searching for auspicious days efficiently, scalably, and free from mystical bias.

II. THEORETICAL BASIS

A. Set

A set is defined as a collection of objects that are unordered and distinct from each other, where these objects are commonly called elements or members. The membership of an element x in set A is denoted by $x \in A$, whereas if the element is not a member of set A, it is denoted by $x \notin A$. When a set contains no elements (meaning its cardinality is equal to zero) it is called the null set and is denoted by $\varnothing$ or $\{\ \}$.

Furthermore, a set A is said to be a subset of a set B, denoted by $A \subseteq B$, if and only if every element of A is also an element of B, which is formally expressed as $A \subseteq B \Leftrightarrow \forall x\ (x \in A \rightarrow x \in B)$. Finally, two sets are considered to be disjoint if they have absolutely no elements in common, indicated by the formal notation $A\ //\ B$.

B. Cardinality and Power Set

The number of distinct elements in a set A is called the cardinal of that set, denoted by n(A) or $|A|$. The power set of a set A is the set whose elements are all subsets of A, denoted by P(A) or 2^A. If a set has cardinality $|A| = m$, then the sum of the cardinalities of its power sets satisfies the equation

$$|P(A)| = 2^m$$

C. Set Operations

In set theory, various operations allow us to manipulate and relate different sets. The intersection of sets A and B, denoted as $A \cap B = \{x \mid x \in A \text{ and } x \in B\}$, produces a new set containing only the elements shared by both. Conversely, their union, denoted as $A \cup B = \{x \mid x \in A \text{ or } x \in B\}$, combines them into a single set containing all elements from either A or B.

When we need to define what is excluded from a set, we use the complement, denoted as $A' = \{x \mid x \in U, x \notin A\}$ (also equivalent to A with an overscore), which contains all elements from the universal set U that are not present in A. Similarly, the difference between two sets, written as $A - B = \{x \mid x \in A \text{ and } x \notin B\} = A \cap B'$, isolates the elements that are included in A but not in B.

Building on this, the symmetric difference yields elements belonging to either A or B, but strictly not their intersection, and is denoted as

$$A \oplus B = (A \cup B) - (A \cap B) = (A - B) \cup (B - A)$$

Finally, rather than just combining individual elements, the Cartesian product, denoted as $A \times B = \{(a, b) \mid a \in A \text{ and } b \in B\}$, produces a new set of ordered pairs where the first element a comes from set A and the second element b comes from set B.

D. The Laws of Set Algebra and the Principle of Duality

Simplification and proof of the equality of set expressions follow the laws of set algebra. These basic rules include the Law of Identity, the Law of Nullity/Dominance, the Law of Complements, the Law of Idempotence, the Law of Involution, the Law of Absorption, the Commutative Law, the Associative Law, the Distributive Law, De Morgan's Laws, and the 0/1 Law.

Duality principle states that two different concepts can be interchanged and still yield the correct answer. If an equality S is proven true, then its dual equality (S^*) obtained by substituting the operations $\cup \rightarrow \cap$, $\cap \rightarrow \cup$, and the constants $\varnothing \rightarrow U$ and $U \rightarrow \varnothing$ is also true.

E. The Principle of Inclusion and Exclusion

The inclusion-exclusion principle is applied to evaluate the cardinality of a union operation involving multiple sets. For the union of two sets, the formula used is

$$|A \cup B| = |A| + |B| - |A \cap B|$$

For the common difference operation of two sets, the cardinality is determined by the equation

$$|A \oplus B| = |A| + |B| - 2|A \cap B|$$

This principle can be extended to the calculation of three sets through the formula condensation, $|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|$.

F. Relations and functions

Relations and functions are fundamental concepts in discrete mathematics that examine the structure of relationships between elements of two or more sets. These concepts form the basis of data mapping, arithmetic models, and algorithmic computation. Several fundamental definitions and postulates exist in this domain.

G. Cartesian Product

Given two sets A and B, the Cartesian product between A and B, denoted as $A \times B$, is the set of all ordered pairs (a, b) such that $a \in A$ and $b \in B$. As an illustration, if $A = \{1, 2\}$ and $B = \{x, y\}$, then this operation yields: $A \times B = \{(1, x), (1, y), (2, x), (2, y)\}$.

H. Definition of Relation A binary relation R from a set A to a set B is a subset of the Cartesian product A × B.

This is formalized as: $R \subseteq (A \times B)$. If an element a is related to an element b, this phenomenon can be denoted as: aRb or $(a, b) \in R$. Operational example: If $A = \{2, 3, 4\}$ and $B = \{4, 6, 9\}$, the relation R is "divisible" by the set elements $\{(2, 4), (2, 6), (3, 6), (3, 9), (4, 4)\}$.

I. Basic Properties of Relations

A relation R on a set A has specific characteristics that affect its logical structure, including:

1) Reflexive: For every element $a \in A$, it holds $(a, a) \in R$.

2)Symmetrical: If $(a, b) \in R$, then $(b, a) \in R$ reciprocally applies.

3) Transitive: If $(a, b) \in R$ and $(b, c) \in R$, then the transition $(a, c) \in R$ applies. A relation that satisfies the three postulates above is simultaneously categorized as an Equivalence Relation, which functions to group elements into equal equivalence classes (an essential concept in modular arithmetic).

J. Function Definition (Mapping)

A function f from set A to set B is a special form of relation that maps each element a ∈ A to exactly one element b ∈ B. This numeric or logical mapping is represented by the symbol: f: A → B. If element a is specifically mapped to b, the equation is written as: f(a) = b.

In a function operation f: A → B, the boundaries and scope of the mapping are separated into three terms:

1) Domain (Source): The set A whose elements are all input values of the mapping.

2) Codomain (Friends): The set B that is the target universe of the function mapping.

3) Range (Source): The subset of B that actually contains the resulting values of the mapping from A.

K. Classification of Function Types Based on their scope and mapping pattern,

functions are divided into three main entities:

1) Injective Function (One-to-One): Each element in B is associated with at most one element in A. This condition requires that if f(a) = f(b), then the initial identity must be the same (a = b).

2) Surjective Function (Onto): Each element in the Codomain space (B) is a mapping from at least one element in the Domain (A). In this scenario, the Range is completely identical to the Codomain.

3) Bijective Function (One-to-One Correspondence): A mathematical equilibrium condition in which a function is proven to satisfy both injective and surjective properties.

L. Number theory

Number theory is a pillar of pure mathematics that focuses on the study of the characteristics of whole numbers (integers) and their associated functions. The definition of an integer refers to a numerical entity that has no decimal fractions. Several fundamental postulates exist in this field.

M. Divisibility Property

If a number x divides evenly into x, this phenomenon can be denoted as: x|x. As an illustration, the number 4 divides evenly into 12, which is represented by the symbol 4|12.

N. *Euclidean Theorem*

Given integers m and n with the constraint $n > 0$. The process of dividing m by n produces a value for q (quotient) and the residue r (remainder), so that the equation applies: $m = nq + r$, with the constraint $0 \le r < n$. Operational example:

1) The operation 1987/97 produces 20 with a remainder of 47, written as: $1987 = 20 \cdot 97 + 47$.

2) The operation -22/3 produces -8 with a remainder of 2, written as: $-22 = (-8) \cdot 3 + 2$.

O. Greatest Common Divisor (GCD)

For two non-zero integers a and b, d is the greatest common divisor if d is the highest integer that satisfies the conditions $d|a$ and $d|b$. This is formalized as:

$$PBB(a, b) = d$$

For example, $PBB(45, 36) = 9$. This is based on the set of divisors of 45: {1, 3, 5, 9, 15, 45} and the set of divisors of 36: {1, 2, 3, 4, 9, 12, 18, 36}, where the maximum overlapping value is 9.

P. Euclidean Algorithm

This method is used to find the GCD of two non-negative integers m and n (where $m \ge n$). By setting $r0 = m$ and $r1 = n$, the division iteration is carried out as follows:

$$\begin{gathered} r0 = r1q1 + r2 ,\ 0 \le r2 < r1 \\ r1 = r2q2 + r3 ,\ 0 \le r3 < r2 \\ \dots \\ rn-2 = rn-1qn-1 + rn,\ 0 \le rn < rn-1 \\ rn-1 = rnqn + 0 \end{gathered}$$

Based on the Euclidean Theorem, the relationship holds:

$$\begin{gathered} PBB(m, n) = PBB(r0 , r1) = PBB(r1 , r2) = \cdots = \\ PBB(rn-1 , rn) = rn. \end{gathered}$$

Therefore, the GCD is determined by the last non-zero remainder.

Illustration: For $m = 80$ and $n = 12$:

$80 = 6 \cdot 12 + 8$

$12 = 1 \cdot 8 + 4$

$8 = 2 \cdot 4 + 0$

The result is that GCD(80, 12) is 4.

Q. Relative Prime

The condition of relative prime is achieved when two numbers have a GCD equal to 1. For example, the number pair 23 and 5 exhibit relative primeness because they have no other common factors other than 1.

R. Algorithm complexity

Algorithm complexity is a measure of an algorithm's efficiency, expressed as a function of the input size n. There are two types of complexity: time complexity $T(n)$ and space complexity $S(n)$. Real execution time measurement cannot be used as a standard reference because it depends on the machine architecture and the compiler used, thus an abstract model that is independent of these factors is required. Time complexity is calculated from the number of typical operations that form the basis of an algorithm, such as comparison operations in search algorithms, or comparison and swap operations in sorting algorithms.

Based on the condition of the input data, time complexity is categorized into three cases: the best case $T_{min}(n)$, the worst case $T_{max}(n)$, and the average case $T_{avg}(n)$. For example, in a sequential search algorithm, $T_{min}(n) = 1$, $T_{max}(n) = n$, and $T_{avg}(n) = \frac{n+1}{2}$.

S. Asymptotic Notation

For large n, the growth behavior of *T(n)* is more important than its precise value. The notation used to express this growth limit is called asymptotic notation, which consists of Big-O, Big-Omega (Ω), and Big-Theta (Θ).

Big-O notation expresses the upper bound of $T(n)$. $T(n) = O(f(n))$ if there are constants C and n_0 such that:

1) $T(n) \leq C \cdot f(n)$, for all $n \geq n_0$. Big-Omega notation expresses the lower bound of T(n). $T(n) = \Omega(g(n))$ if there are constants C and n_0 such that:

2) $T(n) \geq C \cdot g(n)$, for all $n \geq n_0$. Big-Theta notation states that T(n) is of the same order as h(n), that is, if $T(n) = O(h(n))$ and $T(n) = \Omega(h(n))$ simultaneously.

3) Based on Theorem 1. if $T(n)$ is a polynomial of degree ≤ m, then $T(n) = O(n^{uu})$; Big-O notation is simply determined from the highest-power term with the coefficients omitted. Furthermore, the dominance property holds: the exponential function dominates the power function ($y^n > n^p$, $y > 1$), the power function dominates the logarithm ($n^p > \ln n$), and n log n grows faster than n but slower than n^2. The asymptotic addition and multiplication operations obey Theorem 2: if $T_1(n) = O(f(n))$ and $T_2(n) = O(g(n))$, then $T_1(n) + T_2(n) = O(\max(f(n), g(n)))$ and $T_1(n) \cdot T_2(n) = O(f(n) \cdot g(n))$.

T. Complexity Grouping

Based on Big-O notation, algorithms can be grouped into constant $O(1)$, logarithmic $O(\log n)$, linear $O(n)$, linear-logarithmic $O(n \log n)$, quadratic $O(n^2)$, cubic $O(n^3)$, exponential $O(2^n)$, and factorial $O(n!)$, in the following order of growth:

$$1 < \log n < n < n \log n < n^2 < n^3 < \ldots < 2^n < n!$$

The polynomial finite group is considered a good algorithm, while the exponential and factorial groups are considered poor for large n.

U. Sorting Algorithm Case Study

As an illustration, the selection sort and bubble sort algorithms have the following number of comparison operations:

$$T(n) = (n-1) + (n-2) + \ldots + 1 = n(n-1)/2 = O(n^2)$$

In selection sort, the number of exchange operations is $T(n) = n-1$ for all cases, while in bubble sort, the number of exchanges depends on the data conditions: $T_{min}(n) = 0$ in the best case (the data is already sorted) and $T_{max}(n) = n(n-1)/2$ in the worst case. This shows that although both algorithms have the same comparison complexity, bubble sort is generally less efficient than selection sort due to the larger number of exchange operations.

III. RESEARCH METHODOLOGY

A. Discrete Set Transformation of Luni - Solar Calendar Data

The initial phase involves constructing a mathematical model of the Javanese luni-solar calendar. This system operates on two simultaneous timekeeping methods: a standard seven-day week (*Saptawara*) and a five-day market week (*Pancawara*). The convergence of these cycles creates the *Weton*, a recurring 35-day sequence traditionally used to mark an individual's specific birth date.

Two finite, non-empty sets are defined as the fundamental domains of this system:

$$S = \{\text{Ahad, Senin, Selasa, Rabu, Kamis, Jumat, Sabtu}\};\ |S| = 7 \quad (1)$$

$$P = \{\text{Legi, Pahing, Pon, Wage, Kliwon}\};\ |P| = 5 \quad (2)$$

The Weton identity of any individual is represented as an ordered pair drawn from the Cartesian product:

$$W = S \times P, \quad |W| = |S| \times |P| = 35 \quad (3)$$

This confirms that there exist exactly 35 distinct Weton identities, each recurring with a period of 35 days in the Javanese calendar. To assign quantitative weight (Neptu) to each element, two non-injective, surjective discrete mapping functions are defined over the positive integers ($\mathbb{Z}^+$):

$$N_s : S \dashrightarrow \{ 5, 4, 3, 7, 8, 6, 9 \} \quad (4)$$

$$N_p : P \dashrightarrow \{ 5, 9, 7, 4, 8 \} \quad (5)$$

These mappings are implemented as hash maps supporting $O(1)$ lookup time. The complete mapping tables are presented in Table I and Table II.

TABLE I. Saptawara (Seven-Day Cycle) Neptu Mapping

Ahad	5
Senin	4
Selasa	3
Rabu	7
Kamis	8
Jumat	6
Sabtu	9

TABLE II. Pancawara (Five-Day Cycle) Neptu Mapping

Legi	5
Pahing	9
Pon	7

Wage	4
Kliwon	6

For any individual with Weton identity (x, y) ∈ S × P, the total Neptu value is computed via a binary addition operation:

$$N(w) = Ns(x) + Np(y),\ N(w) \in \mathbb{Z}^{+} \quad (6)$$

The resulting Neptu values span a range from a minimum of Ns(Selasa) + Np(Wage) = 3 + 4 = 7 to a maximum of Ns(Sabtu) + Np(Pahing) = 9 + 9 = 18, yielding an integer domain N(w) ∈ [7, 18].

B. Modular Arithmetic Evaluation for Compatibility Classification

The second stage computes the compatibility class of a given couple using linear congruence over the ring of integers modulo 8. Let w1, w2 ∈ W denote the Weton identities of the first and second individuals, respectively. The combined Neptu sum is:

$$T = N(w1) + N(w2),\quad T \in [14, 36] \quad (7)$$

The compatibility class R is determined by evaluating the linear congruence relation:

$$T \equiv R \pmod 8,\quad R \in \{0, 1, 2, 3, 4, 5, 6, 7\} \quad (8)$$

By convention in the Javanese Weton system, the residue R = 0 is reinterpreted as the equivalence class representative 8, yielding the final classification set $\mathscr{R} = \{1, 2, 3, 4, 5, 6, 7, 8\}$. Each element of $\mathscr{R}$ maps bijectively onto one of eight named compatibility categories (Table III). This forms a well-defined surjection $\varphi : \mathscr{R} \rightarrow \mathscr{C}$, where $\mathscr{C}$ is the set of eight compatibility categories, operating within the quotient ring $\mathbb{Z}/8\mathbb{Z}$.

TABLE III. Compatibility Classification Map (R → Category)

R	Category	Interpretation
1	Pegat	Prone to separation; highest risk classification
2	Ratu	Harmonious, respected; socially well-regarded couple
3	Jodho	Ideally matched; considered a natural soulmate pairing
4	Topo	Hardship and perseverance; challenges overcome together
5	Tinari	Fortunate; blessed with ease and luck
6	Padu	Prone to conflict; frequent disagreements expected
7	Sujanan	Prone to infidelity; trust issues anticipated
8	Pesthi	Highest long-term harmony; the most auspicious class

C. Optimized Search Algorithm for the Ideal Wedding Date

The third stage focuses on optimizing the selection of calendar dates. This involves finding dates where the day's *Neptu* value, when added to the couple's combined score (*T*), results in the most auspicious *Dino Pasaran* category. The mathematical target for this calculation is a remainder of 3, which represents *Kekayaan/Gedhong* (Wealth/ Prosperity). Within the traditional method for selecting an event date (Hari-H), this is considered the most favorable outcome.

$$(T + H\_nikah) \equiv 3 \pmod 5 \quad (9)$$

Solving algebraically for H_nikah:

$$H\_nikah \equiv (3 - T) \pmod 5 \quad (10)$$

Because subtraction within a modulo 5 system ($\mathbb{Z}/5\mathbb{Z}$) is closed, the calculation reliably produces a single target value, Hnikah, ranging from 0 to 4. Any date matching this specific Neptu Pasaran value becomes a viable candidate for the wedding. To find these dates efficiently, Algorithm 1 scans a set timeframe by jumping forward in five-day intervals. This periodic approach extracts all valid dates in O(n/5) time, bypassing the need to check every single day individually.

Algorithm 1: FindAuspiciousWeddingDates(T, calendar range)

Input:

T : Integer, combined Neptu value of the couple

calendar_range : Array of accessible dates to scan

Output:

valid_dates : List of auspicious wedding dates

Algorithm

```
target_residue  <- (3 - T) mod 5
valid_dates  <- []
FOR each date d IN calendar_range DO
    sw  <- SaptawaraOf(d)
```

```
    pw <- PancawaraOf(d)
    H_d <- N_s(sw) + N_p(pw)
    IF H_d mod 5 == target_residue THEN
        valid_dates.APPEND(d)
    END IF
END FOR
RETURN valid_dates
```

Since the Pancawara operates on a five-day cycle, suitable dates naturally appear every five days. The improved algorithm calculates the initial starting date directly using the target residue. From there, it simply advances in five-day steps to generate all remaining candidates, turning the search for the first valid date into an O(1) constant-time operation.

IV. RESULT AND ANALYSIS

A. *Case Study: Compatibility Calculation*

To validate the accuracy of the set-theoretic model and modular arithmetic evaluations, a concrete simulation is performed on a specific test couple. The subjects are Callista, born on *Senin Pon* (Monday of the Pon market day), and Ghaniyul, born on *Kamis Legi* (Thursday of the Legi market day).

Step 1: Individual Neptu Evaluation

$$N(w1) = Ns(Senin) + Np(Pon) = 4 + 7 = 11 \tag{11}$$

$$N(w2) = Ns(Kamis) + Np(Legi) = 8 + 5 = 13 \tag{12}$$

Step 2: Binary Accumulation

$$T = N(w1) + N(w2) = 11 + 13 = 24 \tag{13}$$

Step 3: Modular Classification

$$24 \equiv R \pmod{8} \Rightarrow 24 = 3 \times 8 + 0 \Rightarrow R = 0 \equiv 8 \pmod{8} \tag{14}$$

Since the remainder is 0, it is mapped to the equivalence class representative R = 8 by convention, placing this couple in the Pesthi category which is the highest tier of long-term relational harmony. This result is fully consistent with the expected output of the hand-calculated traditional method, confirming the mathematical model's correctness with zero computational error.

TABLE IV. Case Study Computation Summary

Parameter	**Callista(w1)**	**Ghaniyul(w2)**	**Combined**
Weton	Senin pon	Kamis Legi	-
Saptawara(Ns)	4(Senin)	8 (kamis)	-
Pancawara(Np)	7 (pon)	5 (legi)	-
Total Neptu N(w)	11	13	T = 24
Congruence Result	-	-	24 mod 8 = 0 → 8
Compatibility Class	-	-	Pesthi (highest)

B. *Case Study: Wedding Date Optimization*

Proceeding from the validated compatibility result T = 24, the system executes the Hari-H search module. The target congruence is:

$$(24 + H\_nikah) \equiv 3 \pmod{5} \tag{16}$$

Solving algebraically:

$$\begin{aligned} Hnikah &\equiv 3 - 24 \pmod{5} \\ Hnikah &\equiv -21 \pmod{5} \\ -21 &= (-5) \times 5 + 4 \\ Hnikah &\equiv 4 \pmod{5} \end{aligned} \tag{17}$$

Therefore, any Weton combination whose combined Neptu value is congruent to 4 (mod 5) qualifies as an auspicious wedding date. Cross-referencing Tables I and II, valid Weton pairs satisfying $Ns(x) + Np(y) \equiv 4 \pmod{5}$ include combinations such as Ahad Wage ($5 + 4 = 9 \equiv 4 \bmod 5$) and Kamis Kliwon ($8 + 8 = 16 \equiv 1 \bmod 5$ which does not qualify). The algorithm then maps qualifying Weton identities to specific Gregorian calendar dates within a user-specified window.

C. *Discussion: Mathematical Properties of the Model*

Several important mathematical properties of the proposed model merit explicit discussion. Non-injectivity of the Neptu function. The accumulation function $N(w) = N\_s(x) + N\_p(y)$ is a well-defined function from $W = S \times P$ to Z+, but it is not injective since multiple Weton pairs can yield identical Neptu totals. For example, Rabu Legi (7 + 5 = 12) shares its value with Selasa Pahing (3 + 9 = 12). This non-injectivity is an inherent property of the cultural system and is preserved faithfully by the model.

1) Equivalence class partition. The modular classification $T \equiv R \pmod{8}$ partitions the set of all possible combined Neptu sums $T \in [14, 36]$ into exactly 8 equivalence classes under the quotient ring $\mathbb{Z}/8\mathbb{Z}$. Each class is assigned a fixed semantic label from $\mathscr{C}$, forming a consistent surjective mapping.

2) Structural periodicity. The periodicity of the Pancawara cycle (period 5) guarantees that for any fixed

T, valid wedding dates always recur with a cycle of exactly 5 days in the Javanese calendar. This structural regularity is the key property that enables the *O(1)* closed-form optimization described in Section III-C.

D. Complexity Analysis

A rigorous time complexity analysis confirms the efficiency advantage of the proposed computational model over the traditional manual approach employed by Dukun Weton practitioners.

TABLE V. Time Complexity Summary

Stage	**Naive**	**Proposed**	**Justification**
Weton Compatibility Eval	*O(n)*	*O(1)*	Hash-map lookup + single modulus op.
Modular Classification	*O(n)*	*O(1)*	Fixed $\lvert\mathcal{R}\rvert = 8$ table lookup.
DateSearch (full list)	*O(n)*	*O(n/5)*	Periodic jump; visit only 1-in-5 dates.
Date Search(first result)	*O(n)*	*O(1)*	Closed-form offset computation.

As a result, the entire computational process takes *O(n)* time to produce a full list of auspicious wedding dates, or *O(1)* time if only a single date is required. This system vastly outperforms traditional manual scheduling. By leveraging the calendar's natural periodicity, the algorithm completely bypasses the unstructured, step-by-step trial and error of the conventional method.

V. CONCLUSION

This study demonstrates that the traditional Javanese Weton system for matchmaking and event scheduling can be accurately modeled using discrete mathematics. By treating the Saptawara and Pancawara cycles as finite sets, mathematically mapping Neptu scores, and determining compatibility via modulo 8 arithmetic ($\mathbb{Z}/8\mathbb{Z}$), we translated a cultural tradition into a strict computational framework.

Testing this model on a real-world case confirmed that it perfectly mirrors traditional manual calculations, correctly assigning the test couple to the highly compatible Pesthi category without any errors. Applying modular arithmetic also heavily optimized the search algorithm for finding auspicious dates (Hari-H). By taking advantage of the predictable five-day Pancawara cycle, the search time was reduced from a slow, step-by-step process (O(n)) to an instant calculation (O(1)) for finding the first valid date, and O(n/5) for generating a complete list.

Combining ethnomathematics with algorithm design proves that the Javanese calendar is built on solid mathematical logic. This synthesis provides a reliable baseline for integrating the system into modern software applications.

APPENDIX

For further study, the author put a Python implementation of the algorithm designed below. Feel free to check out this YouTube video explaining to get more interactive explanation in this link
https://youtu.be/ANUYYflNINw?si=rR9DUdUQi0ZOcM2K
And python code for simulation
https://github.com/ghanivivo123-wq/makalah-matdis

ACKNOWLEDGMENT

his essay could not have been completed without the help, assistance, and guidance of others. Therefore, the author would like to give his utmost gratitude to:
1. Allah SWT,
2. The lecturer Prof. Dr. Ir. Rinaldi Munir, M.T. in charge of lecturing Discrete Mathematics,
3. Everyone who contributed to the wiki.

REFERENCES

[1] Damayanti, Ratna and Faradiba, Surya Sari. A Systematic Review of Mathematical Interpretations of the Weton Marriage Tradition within Islamic Cultural Contexts. Journal of Science Education. 2026.

[2] Hidayati, A. N., Idris, J., & Marhamah, U. The dynamics of family harmony in the Javanese weton perspective. SHAHIH: Journal of Islamicate Multidisciplinary. 2023.

[3] Rosen, K. H. Discrete mathematics and its applications (8th ed.). McGraw-Hill. 2019.

[4] Suryani, N., et al. The Use of Modulo in Wedding Party: An ethnomathematical study in Kampung Kuta. Infinity Journal. 2026.

[5] Ubaidillah, M. B., & Cindi Ameliana. TRADISI PERJODOHAN BERDASARKAN WETON DAN PASARAN DALAM PRESPEKTIF MAQASHID AL-SYAR'IYYAH. JAS MERAH: Jurnal Hukum Dan Ahwal Al-Syakhsiyyah. 2025.

[6] Utami, N. W., Sayuti, S. A., & Jailani, I. Math and mate in Javanese "Primbon": Ethnomathematics study. Journal on Mathematics Education. 2019.

[7] West, J. S. The adaptation of Javanese Weton on new media for matchmaking applications. Routledge. 2022.

STATEMENT

I hereby declare that the paper I wrote is my own writing, not an adaptation or translation of someone else's paper, and is not plagiarized.

Banyuwangi, June 19th , 2026

Ghaniyul Amri Caulava
13525106