Evaluating Greedy Heuristics for Football Player Decision-Making in Real-Time Attacking Scenario Simulation

Reinsen Silitonga - 13524093
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
Email: [1]13524093@std.stei.itb.ac.id, [2]reinsilitonga12@gmail.com

***Abstract*—This paper presents Tacstra, a real-time two-dimensional football attacking scenario simulator built to evaluate the effectiveness of Greedy algorithm heuristics for player decision-making. Inspired by competitive programming environments such as MIT Battlecode, Tacstra allows users to configure attacking scenarios of up to five attackers against four defenders, assign a distinct Greedy heuristic to each player, and observe the resulting outcome of the simulated possession. Three composite on-ball heuristics (Box Predator, Progressive Carrier, and Safe Builder) and two off-ball movement heuristics (Run Into Box and Find Space) were implemented and tested across six tactical combinations in a 3v4 attacking scenario, with ten runs per combination. The results show that combinations in which the centre forward and the wide attackers employ heuristics with differentiated local-optimum criteria substantially outperform combinations in which all attackers share an identical greedy choice function, achieving up to 100% shot-conversion rate compared to 0% for the worst-performing combination. These findings are discussed in relation to the greedy choice property and are used to derive heuristic recommendations for specific attacking positions.**

***Index Terms*—greedy algorithm, simulation, football, heuristic, decision-making, real-time system**

I. Introduction

Football tactics are commonly described through the lens of player roles and playing styles – for example, a striker who plays as a "goal poacher" making runs in behind the last defender, or a winger acting as a "cross specialist" who stays wide before delivering the ball into the box [2]. While these descriptions are qualitative, the underlying decisions a player makes during an attacking sequence – whether to pass, dribble, or shoot – can be modeled as a sequence of local optimization problems, which is precisely the domain in which the Greedy algorithm operates.

This paper investigates the question: *given a fixed set of Greedy decision rules, which combination of heuristics across attacking positions produces the most effective attacking outcomes in a simulated football scenario?* To answer this, a real-time simulation environment named Tacstra was developed, allowing arbitrary placement of attackers and defenders on a half-pitch, configurable per-player heuristics, and a goalkeeper governed by a difficulty parameter. The simulation terminates upon one of four outcomes: goal, the shot being blocked by a defender, the shot being saved by the goalkeeper, or a seven-second timeout.

The contribution of this paper is twofold. First, it presents the design and implementation of Tacstra as an experimental testbed – analogous to MIT Battlecode – in which Greedy heuristics can be authored, swapped, and compared under identical field conditions. Second, it presents an empirical comparison of six tactical heuristic combinations across sixty simulation runs, and derives a recommendation for which Greedy heuristic is most suitable for the centre forward position versus the wide attacker (winger) position.

II. Greedy Algorithm Background

A. Definition and Principle

A Greedy algorithm solves an optimization problem step by step. At each step, it selects the locally optimal choice available at that moment, without regard for the consequences of that choice on subsequent steps – the so-called "take what you can get now" principle. The algorithm proceeds under the hope that a sequence of locally optimal choices will lead to a globally optimal solution [1].

A Greedy algorithm is characterized by six components: (1) a *set of candidates*, containing the choices available at each step; (2) a *set of solutions*, containing the candidates already selected; (3) a *solution function*, which determines whether the solution is complete; (4) a *selection function*, which chooses a candidate according to a particular – inherently heuristic – greedy strategy; (5) a *feasibility function*, which checks whether a selected candidate may legally be added to the solution set; and (6) an *objective function*, which is to be maximized or minimized [1].

It is important to note that a Greedy algorithm does not guarantee an optimal solution. Because it does not exhaustively evaluate every possible solution, and because multiple valid selection functions may exist for the same problem, the choice of selection function critically determines the quality of the outcome [1]. This property motivates the central experiment of this paper: rather than asking whether Greedy heuristics

work, we ask which selection function performs best for which player role.

B. Mapping Greedy Components to Tacstra

In Tacstra, the six components above are mapped onto the football attacking scenario as follows. The *set of candidates* is the set of possible actions available to the ball carrier at a given simulation frame – shoot, pass to the most advanced teammate, pass to the nearest teammate, or pass to the nearest teammate who is not in an offside position. The *set of solutions* is the sequence of actions (passes and the eventual shot) that have already occurred during the possession. The *solution function* determines whether the simulation has reached a terminal state, corresponding to one of the four end conditions: goal, blocked, saved, or timeout. The *selection function* is the Greedy heuristic assigned to each player – for instance, the `BoxPredator` heuristic selects "shoot" whenever the carrier is inside the penalty box with an unobstructed shooting lane. The *feasibility function* verifies whether a candidate pass target is valid, most notably whether the receiver would be in an offside position at the moment the pass is released. Finally, the *objective function* is to maximize the probability of scoring a goal, as measured empirically across repeated simulation runs.

III. Tacstra: System Design

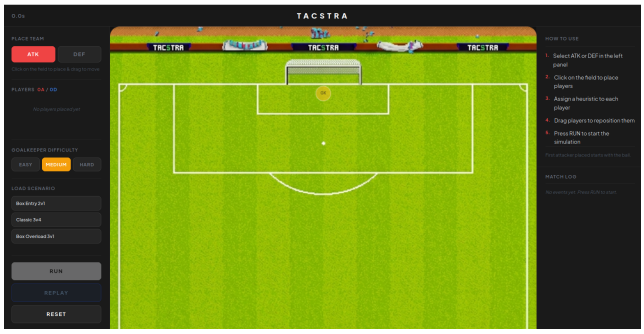

Fig. 1. Tacstra setup interface, showing the field, player placement panel, and per-player heuristic dropdowns.

A. Coordinate System

Tacstra simulates a half-pitch in portrait orientation, with logical dimensions of 680×525 units (approximately one unit per meter, derived from half of a FIFA standard $105\text{m} \times 68\text{m}$ pitch). The origin $(0, 0)$ is located at the top-left of the field, the goal line lies at $y = 0$ with goalposts at $x = 290$ and $x = 390$, and the halfway line lies at $y = 525$. The attacking direction is therefore from high y (own half) toward $y = 0$ (the goal).

B. Player and Ball Model

Each player is represented as an agent with a position, velocity, team affiliation (attacker or defender), and an assigned heuristic. Attackers additionally carry two independent heuristic assignments: an *off-ball heuristic*, governing movement when the player does not possess the ball, and an *on-ball heuristic*, governing decision-making when the player is the ball carrier. The ball itself is modeled as a separate entity with its own position and velocity; while in flight (i.e., between a pass being released and being received), no player controls it.

The simulation runs as a real-time continuous loop driven by `requestAnimationFrame`, updating player and ball positions every frame based on a fixed per-player movement speed and the ball's pass or shot velocity (600 and 900 units/second respectively).

C. Heuristic Catalogue

1) On-ball (ball carrier) heuristics: Each on-ball heuristic implements the *selection function* of the Greedy algorithm: given the current game state S and the ball carrier p, it returns the locally optimal action $a \in \{shoot, pass\}$ together with a target when applicable. Let $T(S, p)$ denote the set of teammates of p that are not in an offside position (the *feasibility function*), and let $blocked(S, p)$ be true if a defender lies within a clearance threshold of the straight line between p and the goal.

Box Predator. Prioritizes shooting once inside the box, and otherwise looks for a teammate already positioned inside the box before falling back to a forward pass.

function BoxPredator(S, p)
if $p \in box$ **and not** $blocked(S, p)$ **then return** *shoot*
end if
$B \leftarrow \{t \in T(S, p) : t \in box\}$
if $B \neq \emptyset$ **then return** *pass*, $\arg\max_{t \in B}$ dist$(t,$ nearestDefender$)$
end if
$F \leftarrow \{t \in T(S, p) : y_t < y_p\}$ ▷ teammates further forward
if $F \neq \emptyset$ **then return** *pass*, $\arg\max_{t \in F}$ dist$(t,$ nearestDefender$)$
end if
if $blocked(S, p)$ **and** $T(S, p) \neq \emptyset$ **then return** *pass*, $\arg\max_{t \in T(S,p)}$ dist$(t,$ nearestDefender$)$
end ifreturn *shoot* ▷ forced shot, no better option

Progressive Carrier. Prioritizes shooting from distance once within range and unblocked, and otherwise seeks a teammate substantially further forward than the carrier (a long progressive pass).

function ProgressiveCarrier(S, p)
if dist$(p, goal) < 200$ **and not** $blocked(S, p)$ **then return** *shoot*
end if
$A \leftarrow \{t \in T(S, p) : y_t < y_p - 80\}$ ▷ far-ahead teammates
if $A \neq \emptyset$ **then return** *pass*, $\arg\max_{t \in A}$ dist$(t,$ nearestDefender$)$
end if
if $p \in box$ **and not** $blocked(S, p)$ **then return** *shoot*
end if
$F \leftarrow \{t \in T(S, p) : y_t < y_p\}$
if $F \neq \emptyset$ **then return** *pass*, $\arg\max_{t \in F}$ dist$(t,$ nearestDefender$)$

end if
if *blocked*(S,p) **and** $T(S,p) \neq \emptyset$ **then return** *pass*, $\arg\max_{t \in T(S,p)}$ dist$(t,$ nearestDefender$)$
end ifreturn *shoot*

Safe Builder. Prioritizes safety: passes immediately when under defensive pressure, and shoots only under low-risk conditions.

function SafeBuilder(S, p)
$D \leftarrow$ defenders within 60 units of p
if $D \neq \emptyset$ **then**
  $F \leftarrow \{t \in T(S,p) : y_t < y_p\}$
  if $F \neq \emptyset$ **then return** *pass*, $\arg\max_{t \in F}$ dist$(t,$ nearestDefender$)$
  else if $T(S,p) \neq \emptyset$ **then return** *pass*, $\arg\max_{t \in T(S,p)}$ dist$(t,$ nearestDefender$)$
  end if
end if
if $p \in$ *box* **and not** *blocked*(S,p) **and** no defender within 100 **then return** *shoot*
end if
$F \leftarrow \{t \in T(S,p) : y_t < y_p\}$
if $F \neq \emptyset$ **then return** *pass*, $\arg\max_{t \in F}$ dist$(t,$ nearestDefender$)$
end if
if *blocked*(S,p) **and** $T(S,p) \neq \emptyset$ **then return** *pass*, $\arg\max_{t \in T(S,p)}$ dist$(t,$ nearestDefender$)$
end ifreturn *shoot*

2) Off-ball (movement) heuristics: **Run Into Box.** Evaluates a fixed set of key positions K inside the penalty box (penalty spot, near-post, far-post, and box-center areas). While the ball is outside the box, the player moves to the unoccupied key position furthest from the nearest defender; once the ball enters the box, the player instead moves to the unoccupied *safe* key position closest to goal.

function RunIntoBox(S, p)
$K \leftarrow$ {6 fixed key positions inside the box}
$U \leftarrow \{k \in K : \nexists\, t \in$ *teammates*(p), dist$(k,t) < 60\}$
if $y_{ball} > y_{boxLine}$ **then** ▷ ball still outside box
  if $U = \emptyset$ **then return** *penaltySpot*
  end ifreturn $\arg\max_{k \in U}$ dist$(k,$ nearestDefender$)$
else
  $V \leftarrow \{k \in U : \nexists\, d \in$ *defenders*, dist$(k,d) < 50\}$
  if $V = \emptyset$ **then return** *penaltySpot*
  end ifreturn $\arg\min_{k \in V}\ y_k$ ▷ closest to goal
end if

Find Space. Evaluates nine candidate offsets around the player's current position and scores each candidate c as a weighted sum of distance from the nearest defender, proximity to goal, and a penalty for clustering near teammates.

function FindSpace(S, p)
$C \leftarrow$ {9 candidate positions: self $\pm$ 70 units in each direction}
$C \leftarrow \{c \in C : c$ is within field bounds, onside$\}$
if $C = \emptyset$ **then return** move slightly backward
end if
for $c \in C$ **do**
  spaceScore$(c) \leftarrow \min($dist$(c,$ nearestDefender$), 300)/300$
  goalScore$(c) \leftarrow 1 - y_c/$*fieldHeight*
  spread$(c) \leftarrow \max(0, (60 -$ dist$(c,$ nearestTeammate$))/60)$
  score$(c) \leftarrow 0.55 \cdot$*spaceScore*$(c) + 0.35 \cdot$*goalScore*$(c) - 0.1 \cdot$*spread*$(c)$
end forreturn $\arg\max_{c \in C}$ *score*(c)

3) Defender heuristics: **Reactive Blocker.** Switches behavior based on whether the ball carrier has entered the penalty box: presses directly if so, otherwise covers the midpoint between the carrier and the most advanced attacker.

function ReactiveBlocker(S, p)
$c \leftarrow$ *ball.carrier*
if $c \in$ *box* **then return** moveTo(c)
end if
$a \leftarrow \arg\min_{t \in attackers}\ y_t$ ▷ most advanced attacker
return moveTo$(\frac{c+a}{2})$ ▷ cover passing lane midpoint

High Presser. Always moves toward the ball carrier; once in close range, repositions onto the line between the carrier and the goal to intercept a potential shot or through-pass.

function HighPresser(S, p)
$c \leftarrow$ *ball.carrier*
if dist$(p,c) > 30$ **then return** moveTo(c)
elsereturn moveTo$(\frac{c+goal}{2})$
end if

D. End Conditions

A simulation run terminates under one of four conditions: *goal*, when the ball crosses the goal line within the post boundaries; *blocked*, when a defender intercepts the shot trajectory before it reaches the goal; *saved*, when the ball enters the goalkeeper's reaction radius (a parameter scaled by a configurable difficulty setting of easy, medium, or hard); and *timeout*, when the possession exceeds seven seconds without resolution.

IV. Experimental Design

A. Scenario Configuration

Fig. 2. Initial configuration of the 3v4 attacking scenario used across all sixty experimental runs, showing the starting positions of the centre forward, wide attackers, central defenders, full-backs, and goalkeeper.

To evaluate heuristic effectiveness under a tactically realistic setup, all experiments used a fixed 3-versus-4 attacking scenario, reflecting common real-world formations such as 4-3-3 or 4-2-3-1 in which three attackers face a back four. The attacking unit consisted of one centre forward (CF) and two wide attackers (LWF and RWF), while the defending unit consisted of two central defenders and two full-backs. Defender heuristics were held constant across all experiments: the two central defenders used *Reactive Blocker*, and the two full-backs used *High Presser*, reflecting a typical defensive shape in which central defenders cover dangerous passing lanes while wide defenders press aggressively. The goalkeeper was set to *easy* difficulty for all runs. Player positions were held constant across all sixty runs (with a small positional jitter of ± 10 units to avoid degenerate determinism), with the LWF starting in possession of the ball. The initial layout is illustrated in Fig. 2.

B. Tactical Combinations

Six tactical combinations (K1–K6) were tested, each defined by a uniform on-ball heuristic and a uniform off-ball heuristic applied to all three attackers, as shown in Table I. Each combination was run ten times, for a total of sixty simulation runs.

Fig. 3. A mid-possession frame from a K2 simulation run, showing the centre forward making a run into the box while the wide attackers occupy separate areas of the pitch.

TABLE I
TACTICAL COMBINATIONS TESTED

ID	Name	Off-ball	On-ball
K1	Full Counter Attack	Run Into Box	Box Predator
K2	CF Finisher, Winger Creator	Find Space	Progressive Carrier
K3	CF Distributor, Winger Finisher	Run Into Box	Box Predator
K4	CF Creative, Winger Supportive	Run Into Box	Progressive Carrier
K5	Full Possession	Find Space	Safe Builder
K6	CF Direct, Winger Circulation	Find Space	Safe Builder

It should be noted that K2 and K6 additionally differ from K1, K3, K4, and K5 in that the centre forward retains the *Run Into Box* off-ball behavior and *Box Predator* or *Progressive Carrier* on-ball behavior respectively, while the wide attackers are assigned a contrasting heuristic pair (*Find Space* with *Progressive Carrier* for K2, and *Find Space* with *Safe Builder* for K6). This differentiation between the centre forward's role and the wide attackers' role is the central manipulation under investigation.

V. RESULTS

A. Outcome Distribution

Table II presents the outcome distribution across the ten runs of each combination.

TABLE II
OUTCOME DISTRIBUTION PER COMBINATION (10 RUNS EACH)

ID	Goal	Saved	Blocked	Offside	Timeout
K1	0 (0%)	0 (0%)	10 (100%)	0	0
K2	10 (100%)	0 (0%)	0 (0%)	0	0
K3	1 (10%)	0 (0%)	9 (90%)	0	0
K4	1 (10%)	2 (20%)	7 (70%)	0	0
K5	0 (0%)	2 (20%)	8 (80%)	0	0
K6	8 (80%)	0 (0%)	2 (20%)	0	0

No offside outcomes were recorded across any of the sixty runs. This is attributable to the *Reactive Blocker* heuristic, which continuously repositions central defenders between the ball carrier and the most advanced attacker, structurally preventing the offside condition from arising during forward passes in this particular defensive configuration.

B. Goal Rate by Combination

The goal conversion rate varied dramatically across combinations, ranging from 0% (K1, K5) to 100% (K2). Combinations in which the centre forward and wide attackers were assigned differentiated heuristics (K2, K6) substantially outperformed combinations in which all three attackers shared an identical heuristic pair (K1, K3, K4, K5).

Average possession duration also varied notably: K1 (all *Run Into Box* / *Box Predator*) resolved fastest at approximately 780 ms, reflecting an immediate but easily blocked shot attempt, whereas K5 (all *Find Space* / *Safe Builder*) took the longest at approximately 2200 ms on average, reflecting a patient build-up that nonetheless failed to produce a clean shooting opportunity in any of the ten runs.

VI. DISCUSSION

A. Why Differentiated Heuristics Outperform Uniform Heuristics

The central finding of this study is that uniform heuristic assignment across all attacking positions (K1, K5) consistently produced inferior outcomes compared to differentiated assignment (K2, K6). This result can be explained directly through the greedy choice property: when all three attackers apply the same selection function under the same local conditions, they tend to converge on the same locally optimal target – in the case of K1, all three attackers run toward the penalty box simultaneously. Because the defending side's *Reactive Blocker* heuristic is itself reactive to the position of the most advanced attacker, a homogeneous attacking threat is comparatively easy

for the defense to neutralize, as a single covering position can address multiple attackers clustered in the same area.

In contrast, when the centre forward is assigned a box-oriented selection function (*Box Predator* or *Progressive Carrier* with *Run Into Box* movement) while the wide attackers are assigned a space-oriented selection function (*Find Space*), the resulting attacking shape is structurally diversified. The defending heuristics, which are themselves Greedy and therefore reactive to only the single most immediate local threat, are unable to simultaneously address a central runner and two attackers occupying separate areas of the pitch. This is consistent with the theoretical observation that a Greedy algorithm's quality is highly sensitive to the chosen selection function, and that no single selection function is universally optimal independent of context [1].

B. Real-World Validity of the Full Counter Attack and Full Possession Failures

The fact that the two worst-performing combinations correspond to two recognizable real-world tactical philosophies – a direct, all-out counter-attack (K1) and a patient, all-out possession game (K5) – is not an artifact of poorly chosen heuristics, but rather a result that is consistent with known limitations of those styles when key supporting mechanics are absent.

Full Counter Attack (K1). In real football, a counter-attacking team that simply passes and immediately advances forward will frequently fail to progress the ball against a well-organized high press, because ground passes into advancing runners are easily intercepted once the defending line compresses space. A counter-attack of this kind is typically only effective when supported by a *lofted (through) pass* that is precisely timed against a forward run, allowing the ball to bypass the defensive line entirely rather than being threaded through it. Tacstra does not currently implement a lofted pass mechanic – all passes travel along the ground at a fixed height and speed – and consequently the *Run Into Box* / *Box Predator* combination cannot exploit the timing-based passing lane that real counter-attacking play depends on. The 0% goal conversion rate observed for K1 is therefore a valid and expected consequence of this missing mechanic, rather than evidence that direct counter-attacking play is inherently ineffective.

Full Possession (K5). Conversely, despite producing visibly more organized passing sequences than K1 (averaging 6 passes per run compared to 2 for K1), Full Possession also failed to register a single goal. This mirrors a well-documented requirement of real-world possession-based tactics: sustained possession against a compact defensive block requires individual *dribbling* ability to break the defensive line, not passing alone. Close control and feints allow an attacker to bypass a marker who is actively engaging them, which is precisely what opens a passing lane that did not previously exist [3]. Furthermore, when a dribbler beats their marker, the defending unit is forced to shift its shape to cover the breach, which creates additional space and passing angles elsewhere on the pitch [4]. Tacstra's attacker agents do not implement a dribbling action – the ball carrier may only pass or shoot – so the *Find Space* / *Safe Builder* combination can recirculate possession indefinitely without ever being able to individually beat a defender to create the penetration that real possession football relies on. The 0% conversion rate for K5 is therefore likewise a valid result given the current scope of the simulation, and points to dribbling as a natural extension for future versions of Tacstra.

C. Positional Recommendations

Based on the experimental results, this paper recommends position-specific Greedy heuristic assignments as follows:

- **Centre Forward (CF):** The *Run Into Box* off-ball heuristic combined with either *Box Predator* or *Progressive Carrier* on-ball heuristic is recommended. This combination consistently positioned the CF as the primary goal threat across the two best-performing combinations (K2 and K6).
- **Wide Attacker (LWF/RWF):** The *Find Space* off-ball heuristic is recommended over *Run Into Box*. When wide attackers also rush into the box (as in K1, K3), the attacking shape collapses into a single contested area, and the resulting goal rate drops to 10% or below.
- **Defender (Central):** The *Reactive Blocker* heuristic proved effective at preventing offside opportunities and at covering the most dangerous passing lane, though it remained vulnerable to geometrically diversified attacks, as demonstrated by its failure to prevent goals in K2 and K6.

D. Limitations

This study is subject to several limitations. First, all experiments were conducted against a fixed defensive configuration (*Reactive Blocker* centrally, *High Presser* on the flanks, and an *easy*-difficulty goalkeeper); the relative ranking of attacking combinations may differ under alternative defensive heuristic assignments. Second, the sample size of ten runs per combination, while sufficient to reveal clear directional trends in this study, is modest for fine-grained statistical comparison between closely performing combinations such as K3 and K4. Third, the Greedy heuristics implemented here are reactive and memoryless – they do not account for the trajectory of play prior to the current frame – which is an inherent characteristic of the Greedy paradigm rather than a limitation specific to this implementation.

VII. Conclusion

This paper presented Tacstra, a real-time football attacking scenario simulator designed as an experimental testbed for comparing Greedy algorithm heuristics across player positions. Through sixty simulation runs across six tactical combinations in a 3v4 attacking scenario, this study found that attacking effectiveness is maximized not by assigning the most individually aggressive heuristic to every player, but by assigning *differentiated* Greedy selection functions across positions –

specifically, a box-oriented heuristic for the centre forward paired with a space-seeking heuristic for wide attackers, which achieved goal conversion rates of 80–100%, compared to 0% when all attackers shared an identical heuristic. These findings reinforce the theoretical principle that the performance of a Greedy algorithm is fundamentally governed by the choice of its selection function, and demonstrate that this principle extends naturally to the domain of multi-agent tactical decision-making in a simulated sports environment.

VIDEO LINK AT YOUTUBE

A demonstration video of Tacstra, including a walkthrough of the implementation and the K1 and K2 simulation runs, is available at: https://youtube.com/PLACEHOLDER

SOURCE CODE REPOSITORY

The source code of Tacstra is publicly available at: https://github.com/re1nsilitonga/greedy-football-sim

ACKNOWLEDGMENT

The author would like to express gratitude to Mr. Rinaldi Munir and all IF2211 teaching assistants for the valuable knowledge provided, which supported the development of this program.

REFERENCES

[1] R. Munir, "Algoritma Greedy (Bagian 1)," Bahan Kuliah IF2211 Strategi Algoritma, Program Studi Teknik Informatika, Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, 2026.

[2] "Playing Styles Explained," PES My Club Guide. [Online]. Available: https://pesmyclubguide.com/playing-styles/

[3] "Why Dribbling is Crucial for Breaking the Press," YouTube. [Online]. Available: https://youtu.be/J8HAHeRZfkw

[4] Frontiers in Psychology, "The role of dribbling and individual ball-carrying skill in creating overloads against organized defenses," *Front. Psychol.*, 2023. [Online]. Available: https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2023.1197039/full

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 19 Juni 2026

Reinsen Silitonga 13524093