Idealis: Indonesia Journal Information System
Volume 9, Nomor 2, Juli 2026
ISSN 2684-7280 (*online*)
Halaman 302-313
available online at http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index

A Static Analysis of Privacy by Design Compliance in Indonesian Quick-Service Restaurant Applications

Emanuel Ristian Handoyo[1*], Flourensia Sapty Rahayu[2]

[1,2]Sistem Informasi, Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta, Yogyakarta, Indonesia
Email: [1*]emanuel.handoyo@uajy.ac.id, [2]sapty.rahayu@uajy.ac.id
(*: *corresponding author*)

Abstract - *The adoption of Quick-Service Restaurant (QSR) apps in Indonesia tripled from 2020 to the following three years. However, technical privacy audits of local apps remain very limited, and no research has systematically evaluated QSR apps in Indonesia using technical methods. To address this gap as the study's objective, the privacy compliance of QSR applications was systematically evaluated. Methodologically, the PbD-MASVS-MobSF evaluation framework was implemented, wherein Privacy by Design (PbD) principles, OWASP MASVS privacy controls, and Mobile Security Framework (MobSF) processes were integrated. As a sample, seven QSR apps were analysed, and their privacy findings were compared to the provisions of Law No. 27 of 2022 on Personal Data Protection (UU PDP). Regarding the key results, it was found that the apps were at a MEDIUM to HIGH risk level, with MobSF scores of 39-54 out of 100. The compliance analysis found that the apps consistently failed to meet six of the ten MobSF subprocesses. Meanwhile, strong compliance was only identified in the absence of API access for device identification. These findings indicate that privacy risks in the Indonesian QSR sector are structural and sectoral. Crucially, because only static analysis was utilised in this assessment, the findings are considered indicative rather than conclusive, and legal non-compliance with the PDP Law cannot be independently established. As a primary contribution, an operational framework that connects static analysis results, privacy design principles, and national regulatory requirements is provided as a reference for application developers, privacy auditors, and regulators.*

Keywords*: Privacy by Design, static analysis, QSR applications, Android privacy, Personal Data Protection Law (UU PDP).*

1. INTRODUCTION

Indonesia's digital ecosystem has undergone a significant structural transformation, driven by an internet penetration rate reaching 81.72% of the population in 2026, representing over 235.3 million active users [1]. The demand for data protection has intensified after several high-profile breaches, including the Bjorka case, which exposed vulnerabilities in citizens' identity information [2]. Furthermore, surveys indicate that 35.1% of Indonesian internet users express concerns about corporate use of personal data [3], [4]. This trend aligns with restaurant-sector findings that perceived privacy risk is a significant determinant of consumer trust in digital food services [5]. These concerns are amplified by app-based loyalty programs, whose data-personalisation designs are frequently undermined by user privacy concerns [6] and which harvest not only customer preferences but also sensitive information, often under ambiguous and complex privacy policies that demand careful due diligence and information governance [7].

The risks are most acute in the quick-service restaurant (QSR) sector, the focus of this study. Documented incidents illustrate the sector's exposure: API vulnerabilities in a major food-delivery service have exposed sensitive customer identity data, including contact information, delivery addresses, and social-media profiles [8]; and a pizza-delivery service breach leaked extensive order and location records [9]. Although these incidents occurred abroad, the same multinational QSR brands operate localised applications in the Indonesian market. Consequently, their underlying data-collection architectures and third-party SDK dependencies not only expose Indonesian users to comparable technical privacy risks but also pose a significant challenge to national digital sovereignty, as citizen data is frequently processed by foreign digital platforms and stored outside domestic legal jurisdiction [10]. These cases reflect a broader pattern: a large-scale study of nearly two million Android applications found that the prevalence of third-party trackers has remained high and virtually unchanged since the GDPR, indicating persistent privacy risk attributable to insufficient technical compliance and weak enforcement [11]. In contrast, systematic reviews of the Android privacy literature confirm that over-declared permissions and third-party data leakage represent the most persistent risk categories across consumer mobile applications [12].

Privacy risk evaluation increasingly relies on static analysis, with the Mobile Security Framework (MobSF) as a primary tool [13] grounded in the OWASP Mobile Application Security Verification Standard (MASVS) [14]. Static analysis is one of three dominant evaluation mechanisms in the Android privacy literature [12], and MobSF has been validated for detecting insecure storage, weak cryptography, and third-party trackers in mobile banking [15], e-commerce [16], and popular Android applications [17], [18], [19]. A persistent methodological challenge, however, is translating technical findings into a broader normative framework such as Privacy by Design (PbD) [20], whose software-engineering implementation is impeded by the gap between principles and concrete guidelines [21], [22], [23], [24], [25]. To bridge this gap, the present study maps each MobSF finding, such as the detection of trackers, unencrypted storage, or static credentials, to relevant PbD principles, including data minimisation, security by default, and transparency, and then relates these to specific articles of the UU PDP, so

Idealis: Indonesia Journal Information System
Volume 9, Nomor 2, Juli 2026
ISSN 2684-7280 (*online*)
Halaman 302-313
available online at http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index

that technical evaluation culminates in governance- and compliance-oriented recommendations rather than vulnerability detection alone. A compact conceptual diagram illustrating this entire analytical flow is presented in the Research Method section (Figure 1).

The QSR sector is especially exposed because rapid digital transformation and cross-channel strategies [26] have turned loyalty applications into data-harvesting ecosystems that collect information beyond core-service needs [27], with documented concerns over tracker prevalence [11], geolocation [28], and risks to vulnerable groups such as children [29]. In Indonesia, consumer trust in digital platforms is closely tied to data security [5], [30], yet technical audits of local applications remain rare and case-specific rather than sectoral [31], and the privacy-violating default behaviour of third-party SDKs [32] together with discrepancies between declared and actual data collection [33] make systematic technical audits urgent. The UU PDP [34] embodies preventive data-protection principles aligned with Privacy by Design. It obliges controllers to implement technical and operational protection measures (Article 35), with sanctions of up to 2% of annual revenue effective October 2024. However, QSR loyalty applications in Indonesia tripled between 2020 and 2023 [35]. No public technical audit has assessed their UU PDP compliance, and prior work mapping local-to-international convergence [36], [37] has not connected specific technical weaknesses to sanctions exposure [13].

Accordingly, this study addresses three gaps in the Indonesian literature: the lack of MobSF-based evaluations of Indonesian QSR applications despite their widespread use [17], [28], [38]; the predominance of normative privacy-policy analyses such as those by Mulligan et al. [29] and Blancaflor et al. [28], over technical audits, and the limited integration of Privacy by Design (PbD) principles with the requirements of Indonesia's Personal Data Protection Law (UU PDP) in this sector. To address these gaps, the study examines the extent to which widely used Indonesian QSR applications comply with Privacy by Design principles through a comparative case study of seven applications, using MobSF to analyse permission practices, cryptographic implementations, and data-handling mechanisms. PbD principles and MASVS-PRIVACY controls from OWASP MASVS v2.1.0 [14] serve as the analytical bridge between technical findings and UU PDP obligations.

The novelty of the proposed PbD–MASVS–MobSF framework lies in its integration of three layers of analysis. While previous MobSF-based studies have focused primarily on general security maturity and prior QSR privacy research has concentrated on privacy-policy texts, this framework connects these two streams through a transparent process linking static code analysis, privacy-design evaluation, and national regulatory assessment. To the best of our knowledge, such an approach has not previously been applied to Indonesian QSR applications or to privacy audits aligned with the UU PDP. Building upon this novelty, this study makes three main contributions. First, it introduces a three-layer evaluation framework that operationalises PbD principles and MASVS-PRIVACY controls within MobSF-based analysis, creating a replicable audit methodology for the Indonesian QSR context. Second, it presents the first sector-level technical audit of seven widely used Indonesian QSR applications, complementing policy-focused studies [28], [29] with code-level evidence derived from static analysis. Third, it establishes an explicit mapping between technical findings and relevant UU PDP provisions, enabling governance and compliance implications to be derived directly from application behaviour rather than from privacy-policy review alone [34].

2. RESEARCH METHOD

This study uses a descriptive-comparative approach based on static analysis of Android Package Kit (APK) files, applying an integrated PbD-MASVS-MobSF framework that, to our knowledge, is the first to combine these three for Indonesian QSR applications. As visualised in Figure 1, the framework maps Privacy by Design (PbD) to OWASP MASVS-Privacy. It operationalises both through static-analysis subprocesses within the Mobile Security Framework (MobSF), linking privacy principles, technical standards, and code-level evidence into a single analytical process. A posteriori, the results are further related to relevant articles of the Personal Data Protection Law (UU PDP) [33], applied only after the technical analysis to keep regulatory interpretation evidence-grounded; its interpretive limits are addressed at the end of this section. All analysis is static: no execution or runtime interaction is involved.

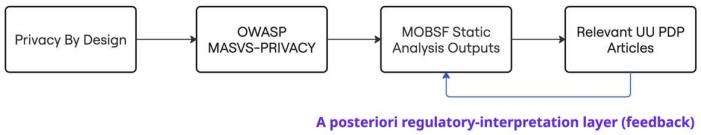

Figure 1. Conceptual chain of the proposed evaluation approach

The research objects are seven Android applications from QSR brands operating in Indonesia, selected based on independent market intelligence data from Statista ranking the leading official restaurant applications in

Idealis: Indonesia Journal Information System
Volume 9, Nomor 2, Juli 2026
ISSN 2684-7280 (*online*)
Halaman 302-313
available online at http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index

Indonesia by download volume [39]. The seven are the top-ranked official QSR apps in that ranking; concentrating on this leading tier provides representative coverage of the sector's most widely used applications while keeping the sample tractable for in-depth, per-artefact analysis[39]. All applications were sourced directly from the official Google Play Store; iOS was intentionally excluded because Android dominates the Indonesian mobile market [4], and MobSF's static-analysis capabilities are most standardised for Android Application Package (APK) artefacts. The selected applications represent the largest multinational and national QSR networks, each with extensive feature integration spanning online ordering, loyalty programs, and digital payments. Their identities are anonymised as Application A through Application G to keep the focus on sector-level patterns and to avoid publishing exploitable, brand-linked vulnerabilities; a controlled replication mechanism preserves reproducibility despite the anonymisation. Each APK's version is recorded for traceability: Application A, v3.46.1; B, v4.6.0; C, v3.1.20; D, v15.0.18; E, v4.2.7; F, v3.6.0; G, v3.22.5. All seven were scanned on 13 June 2026 using MobSF v4.5.0 with default static-analysis configuration (default ruleset, no custom rules or modified thresholds). SHA-256 hashes of all seven APKs are documented, and controlled access to these hashes and the unmasked brand mapping will be granted to verified researchers upon reasonable request, ensuring replication uses identical artefacts. The unit of analysis is a single, immutable APK file per application. Analysis is limited to static methods because dynamic analysis would require interaction with live backend servers and real user data, raising ethical and replicability concerns that static analysis of code and configuration avoids.

PbD, originally proposed by Cavoukian and since adopted internationally as a privacy framework, comprises seven principles: (PbD-1) proactive rather than reactive; (PbD-2) privacy by default via data minimization; (PbD-3) privacy embedded into design; (PbD-4) full functionality through a positive-sum approach; (PbD-5) end-to-end data protection; (PbD-6) visibility and transparency; and (PbD-7) respect for user privacy. PbD-4 is excluded from this study as an architectural meta-principle not directly observable through static APK analysis; an acknowledged limitation narrows the evaluation's scope relative to a full PbD assessment. The remaining six principles (PbD-1, PbD-2, PbD-3, PbD-5, PbD-6, PbD-7) are mapped to the four OWASP MASVS-Privacy controls [14]: data minimisation and third-party SDK accountability (MASVS-PRIVACY-1), unlinkability (MASVS-PRIVACY-2), transparency in data collection and sharing (MASVS-PRIVACY-3), and user control over personal data (MASVS-PRIVACY-4). The lead author performed the mapping from Cavoukian's foundational paper [20] and OWASP MASVS-Privacy v2.1.0 [14]; co-authors independently reviewed each assignment, with disagreements resolved by consensus. Each PbD–MASVS pair was assessed bidirectionally: whether the PbD principle's core concept is explicitly operationalised by the MASVS control's primary clause, and whether the control's primary theme directly enacts the principle's intent. Pairs were classified as Full Support when both directions were affirmative (e.g., MASVS-PRIVACY-1's requirement that SDKs operate 'based on user consent, not by default or without it' directly operationalizes PbD-2's data-minimization mandate), or Partial Support when the correspondence was limited to a secondary or implicit element (e.g., PbD-1's proactive orientation appears only in a secondary preventive clause of MASVS-PRIVACY-1, whose primary theme is data minimization rather than pre-emptive risk prevention).

MobSF, an open-source APK analysis tool, provides two categories of static analysis: Code Analysis (Manifest Weaknesses and Vulnerabilities) and Data Flow (Privacy Leaks). To limit researcher subjectivity, only outputs directly relevant to MASVS-Privacy requirements were selected. Other MobSF report outputs, including the overall Security Score, Certificate/Signing analysis, Network Security Configuration, and third-party Malware/VirusTotal scanning, were deliberately excluded from this mapping, as they assess general application security posture rather than privacy-specific compliance under MASVS-Privacy. This yielded ten subprocesses mapped across PbD, MASVS-Privacy, and MobSF in Table 1. The mapping density reflects each control's scope: MASVS-PRIVACY-1, being broadest (covering permissions, manifest configuration, tracker identification, and data flow), accounts for six subprocesses, while the three narrower controls each map to one; similarly, PbD-5 maps to three Data Flow subprocesses because verifying end-to-end lifecycle protection requires tracing data flow from source to sink, generating a call graph, and performing taint analysis as an integrated sequence.

Table 1. PbD-MASVS-MobSF framework mapping

PbD	MASVS-Privacy	MobSF	Support Level
PbD-1	MASVS-PRIVACY-1	Code Analysis: 1.2.1 Identifying Trackers	Partial
PbD-1	MASVS-PRIVACY-1	Code Analysis: 1.2.4 Determining Vulnerable Call	Partial
PbD-2	MASVS-PRIVACY-1	Code Analysis: 1.1.1 Checking Permissions	Full
PbD-2	MASVS-PRIVACY-1	Code Analysis: 1.1.3 Checking Configuration	Full
PbD-3	MASVS-PRIVACY-2	Code Analysis: 1.2.3 Extracting Methods	Partial
PbD-5	MASVS-PRIVACY-1	Data Flow: 2.1 Searching Method Calls	Full
PbD-5	MASVS-PRIVACY-1	Data Flow: 2.2 Generating Call Graphs	Full
PbD-5	MASVS-PRIVACY-1	Data Flow: 2.3 Taint Analysis	Full

Idealis: Indonesia Journal Information System
Volume 9, Nomor 2, Juli 2026
ISSN 2684-7280 (*online*)
Halaman 302-313
available online at http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index

PbD	MASVS-Privacy	MobSF	Support Level
PbD-6	MASVS-PRIVACY-3	Code Analysis: 1.2.2 Matching Sensitive Data	Full
PbD-7	MASVS-PRIVACY-4	Code Analysis: 1.1.2 Checking Protection Level	Full

Source: developed with reference to Cavoukian [20], OWASP MASVS-Privacy v2.1.0 [14], and the MobSF Static Analysis documentation [13].

The methodology followed five sequential stages: (1) APK collection, (2) MobSF scanning, (3) extraction of findings, (4) framework mapping, and (5) qualitative interpretation. Applications were installed on an Android smartphone via the official distribution platform, retrieved as APK files using the APK Extractor application, and transferred to a local computer, with each version accurately recorded. Each APK was then examined in MobSF static mode without running the application, ensuring no runtime interaction or data transmission. Findings from the ten subprocesses were compiled into a seven-applications-by-ten-subprocesses comparison matrix, enabling cross-application and cross-subprocess comparison. Each finding was then interpreted directly against the explicit requirements of its mapped MASVS-Privacy control and characterised as showing strong, weak, or variable alignment. For example, trackers operating without evidence of consent contradict MASVS-PRIVACY-1, whereas the absence of device-identification API access is consistent with MASVS-PRIVACY-2. Because static findings are deterministic for a given file, identical APK versions and configurations yield consistent, traceable results. The aggregate MobSF security score (0–100) is additionally used as a concise positioning metric: the standard MobSF algorithm starts from a base of 100, subtracting 15 points per high-severity finding and 10 per warning, adding 5 per good assessment, and bounding the result between 10 and 100. Risk thresholds follow MobSF's own categorisation (0–15 critical, 16–40 high, 41–70 medium, 71–100 low); this study collapses these into HIGH RISK (0–40), MEDIUM RISK (41–70), and LOW RISK (71–100), complementing rather than replacing qualitative interpretation [40].

Finally, each application's privacy profile was examined against the Indonesian Personal Data Protection Law (UU PDP) through an a posteriori review [34], conducted qualitatively rather than via numerical scoring since privacy findings often require contextual interpretation that simple metrics would flatten. Each finding was assessed for relevance to key legal obligations, including purpose limitation, consent-based processing, and transparency toward data subjects, with results presented narratively. The resulting relationships should be understood as indicative rather than definitive. They are based on thematic alignment between the technical findings and the substance of the relevant legal provisions and do not constitute a legal determination of non-compliance. Establishing whether a violation has occurred would require additional information beyond static code analysis, such as privacy notices, consent mechanisms embedded in the user interface, and the operational practices of data controllers, none of which can be fully assessed through APK analysis alone.

3. RESULTS AND DISCUSSION

3.1. Conceptual Mapping of PbD-MASVS-Privacy

To establish consensus reliability for this judgment-based process, the initial semantic mapping was performed by the lead author and subsequently subjected to independent verification by the co-authors. Any discrepancies in interpretation were evaluated through iterative consensus discussions until full agreement was achieved on the final categorisation. The bidirectional semantic analysis shows that four of the six mapped PbD-MASVS pairs achieve Full Support and two achieve Partial Support. Full Support holds for PbD-2 with MASVS-PRIVACY-1 (data minimisation and access restriction as the default state), PbD-5 with MASVS-PRIVACY-1 (SDK-supply-chain accountability across the data lifecycle), PbD-6 with MASVS-PRIVACY-3 (visibility of data collection), and PbD-7 with MASVS-PRIVACY-4 (consent, access, and control mechanisms). Partial Support holds for PbD-1 with MASVS-PRIVACY-1, whose preventive aspect is reflected only in secondary elements of the control, and for PbD-3 with MASVS-PRIVACY-2, whose unlinkability techniques imply but do not fully capture the embedded-in-architecture concept. MASVS-PRIVACY-1, therefore, has the broadest linkage, connecting three PbD principles and six of the ten subprocesses, which makes findings under this control the most representative indicators of overall compliance.

3.2. Privacy Risk Profiles of the Seven Applications

The MobSF static-analysis output for the seven applications shows that all fall into the MEDIUM to HIGH-RISK categories, with none reaching LOW RISK. On the 0 to 100 aggregate scale, Application A scored 47 (MEDIUM), with 9 dangerous permissions, 7 tracker SDKs, 0 Manifest HIGH, 18 Manifest WARNING, and 5 Code HIGH findings; Application B scored 49 (MEDIUM), with 6 dangerous permissions, 4 tracker SDKs, 1 Manifest HIGH, 5 Manifest WARNING, and 1 Code HIGH; Application C scored 54 (MEDIUM), with 5 dangerous permissions, 4 tracker SDKs, 1 Manifest HIGH, 9 Manifest WARNING, and 0 Code HIGH;

Idealis: Indonesia Journal Information System
Volume 9, Nomor 2, Juli 2026
ISSN 2684-7280 (*online*)
Halaman 302-313
available online at http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index

Application D scored 39 (HIGH RISK), with 11 dangerous permissions, 6 tracker SDKs, 8 Manifest HIGH, 13 Manifest WARNING, and 1 Code HIGH; Application E scored 47 (MEDIUM), with 8 dangerous permissions, 6 tracker SDKs, 2 Manifest HIGH, 14 Manifest WARNING, and 1 Code HIGH; Application F scored 54 (MEDIUM), with 7 dangerous permissions, 3 tracker SDKs, 1 Manifest HIGH, 6 Manifest WARNING, and 0 Code HIGH; and Application G scored 45 (MEDIUM), with 6 dangerous permissions, 5 tracker SDKs, 2 Manifest HIGH, 7 Manifest WARNING, and 0 Code HIGH. Application D thus has the highest risk profile, with the lowest score of 39 out of 100, eleven dangerous permissions, and eight HIGH manifest findings. In contrast, Applications C and F record the highest scores of 54 out of 100 yet remain MEDIUM. The score range is narrow, from 39 to 54. Still, individual indicators vary more widely, and applications with identical aggregate scores can differ at the subprocess level, so the aggregate score alone can mask important privacy differences across specific PbD and MASVS dimensions. These aggregate scores are visualised in Figure 2, which plots the seven applications side by side and shows the narrow 39-to-54 score band together with Application D's position closest to the HIGH-RISK threshold.

It is important to note that this quantitative risk profile is intentionally derived exclusively from MobSF's native severity-weighted scoring, whereas PbD and MASVS-Privacy are normative frameworks without an established numeric scoring convention; consequently, their compliance is assessed qualitatively, through the Full/Partial Support classification in Section 3.1 and the weak/strong/variable compliance labels applied later in this section, rather than through a composite numeric score. The MobSF score should therefore be read as the quantifiable technical evidence base, which Table 1 and the subprocess-level interpretation below then map onto specific PbD principles and MASVS-Privacy controls; it is not intended to stand as an independent or disjointed metric from the PbD-MASVS framework. Future work could extend this approach by developing a weighted composite score that integrates PbD and MASVS dimensions directly, converting the current side-by-side application of the three frameworks into a single measurable and repeatable index.

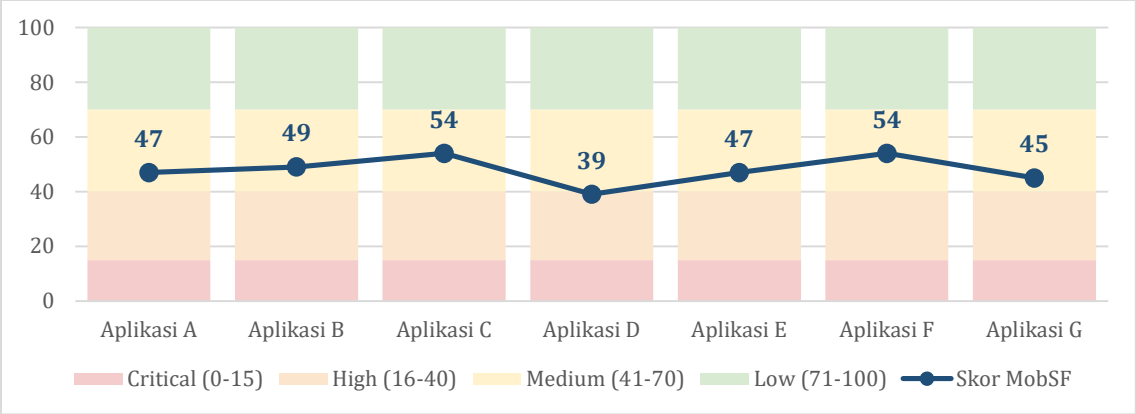

Figure 2. MobSF Scores by Application

Interpreting the ten subprocesses against the MASVS-Privacy requirements reveals a consistent pattern across all seven applications. For this interpretation, a subprocess is classified as weak compliance when its associated MobSF finding is present at Manifest or Code HIGH/WARNING severity in all or nearly all of the seven applications, indicating a systemic rather than an isolated gap; as strong compliance when the associated risk indicator is absent or fully mitigated across all seven applications; and as variable when the finding's presence or severity differs materially across applications. These labels are the authors' interpretive classification applied to MobSF's per-finding severity output and are not categorical judgments generated by the tool itself. Weak compliance appears in six subprocesses: dangerous-permission requests and their protection levels (1.1.1 and 1.1.2), insecure manifest configuration (1.1.3), the presence of three to seven tracking SDKs without evidence of consent (1.2.1), hardcoded credentials detected through taint analysis (2.3), and hardcoded sensitive data (1.2.2). These reduce to three sector-wide problems, namely weak data minimisation (violating PbD-2 and MASVS-PRIVACY-1), credential leakage (violating PbD-6 and MASVS-PRIVACY-4), and tracker presence (violating PbD-1, PbD-7, and MASVS-PRIVACY-3). Strong compliance appears in only one subprocess, the absence of device-identification API access (1.2.3), indicating a low fingerprinting risk that strongly aligns with MASVS-PRIVACY-2 (Unlinkability), while the structural dataflow subprocesses (2.1 and 2.2) merely confirm that analysis is feasible. The only inter-application variation occurs in vulnerable-call detection (1.2.4), where HIGH code findings range from zero to five, making it the principal differentiator among uniform profiles.

Idealis: Indonesia Journal Information System
Volume 9, Nomor 2, Juli 2026
ISSN 2684-7280 (*online*)
Halaman 302-313
available online at http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index

The high-risk profile assigned to Application D can be attributed to three main factors that explicitly contradict the PbD-MASVS framework. First, all eight Manifest HIGH findings stem from unresolved App Link asset-file configurations across eight exported activities. This configuration weakness leaves the application vulnerable to deep-link interception and is not mitigated by any observable code-level protection. Second, Application D requests eleven dangerous permissions, the highest number among the applications examined. These include READ_CONTACTS, WRITE_CONTACTS, RECORD_AUDIO, READ_PHONE_STATE, and CALL_PHONE, permissions that do not appear essential to the core functionality of a food-delivery service. This excess represents a quantifiable failure to meet the data minimisation mandates required by PbD-2 and MASVS-PRIVACY-1. Attributing the excess to a specific cause, such as inheritance from a third-party SDK ecosystem, would require dependency-level analysis beyond the static, permission-and-manifest scope of this study, and no such claim is made here. Third, the application integrates six tracking SDKs, including AppsFlyer and Sentry, which enable attribution tracking and error monitoring. No code-level evidence of a consent mechanism was identified before these data collection processes occur, further increasing the privacy risks associated with the credential exposure identified in Subprocess 2.3 and directly undermining the transparency and user control requirements of PbD-7 and MASVS-PRIVACY-3.

In contrast, Applications C and F demonstrate substantially lower risk levels and a proportionally stronger alignment with PbD principles. Neither application contains Code HIGH findings, both request relatively few dangerous permissions (five and three, respectively), and they incorporate the smallest number of tracking SDKs in the sample (four and three, respectively). Taken together, the absence of critical code vulnerabilities, a more limited permission footprint, and reduced third-party tracking contribute to a smaller attack surface. As a result, their higher overall evaluation scores are more appropriately interpreted as reflecting stronger privacy and security practices within the PbD-MASVS operational parameters, rather than being the product of statistical variation.

3.3. Relevance to the UU PDP

The risk profiles were subsequently related to the UU PDP, as presented in Table 2. The ten subprocesses examined in this study show an indicative association with eleven distinct articles of the law, including provisions on limited and specific processing (Article 27), technical protection measures (Article 35), high-risk processing assessment (Article 34), oversight of data processors (Article 37), prevention of unauthorized access (Article 39), protection throughout the processing lifecycle (Article 16), recordkeeping of processing activities (Article 31), protection against unlawful processing (Article 38), transparency toward data subjects (Article 21), and consent withdrawal and protection-level rights (Articles 9 and 22). The six subprocesses that exhibited weak compliance show an indicative association with seven of these articles (Articles 9, 21, 22, 27, 37, 38, and 39), suggesting that the identified risks are spread across multiple regulatory obligations rather than concentrated within a single area. These relationships should be interpreted as indicative and thematic rather than as legal determinations. Any formal assessment of compliance would require evaluation by competent authorities, taking into account privacy notices, consent mechanisms within the user interface, and operational practices of data controllers that cannot be observed from APK analysis alone.

Table 2. Indicative association of MobSF subprocesses with relevant UU PDP articles (full in Appendix A)

MobSF Subprocess	UU PDP Art.	Summary of Association
1.2.1 Identifying Trackers	Article 37	The Data Controller must supervise every party involved in the Processing of Personal Data under the Data Controller's control.
1.2.4 Determining Vulnerable Call	Article 35-point a	the preparation and implementation of technical operational measures to protect Personal Data from disruptions to Personal Data processing that are contrary to statutory provisions
1.1.1 Checking Permissions	Article 27	The Data Controller must process Personal Data in a limited and specific, lawful, and transparent manner.
1.1.3 Checking Configuration	Article 39 paragraphs (1)-(2)	The Data Controller must prevent unauthorized access to Personal Data. This is done by building a security system for Personal Data processed using electronic systems in a reliable, secure, and accountable manner.
1.2.3 Extracting Methods	Article 34 paragraph (2) point d	Processing of Personal Data for systematic evaluation, scoring, or monitoring of the Personal Data Subject is categorized as potentially high-risk, requiring a Personal Data Protection impact assessment.
2.1 Searching Method Calls	Article 16 paragraph (1)	The Processing of Personal Data includes: acquisition and collection; processing and analysis; storage; correction and updating; display, announcement, transfer, dissemination, or disclosure; and/or erasure or destruction.

Idealis: Indonesia Journal Information System
Volume 9, Nomor 2, Juli 2026
ISSN 2684-7280 (*online*)
Halaman 302-313
available online at http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index

MobSF Subprocess	UU PDP Art.	Summary of Association
2.2 Generating Call Graphs	Article 31	The Data Controller must record all Personal Data Processing activities.
2.3 Taint Analysis	Article 38	The Data Controller must protect Personal Data from unauthorized processing.
1.2.2 Matching Sensitive Data	Article 21 paragraph (1)	The Data Controller must provide information regarding: the legality of Personal Data processing; the purpose of processing; the type and relevance of Personal Data to be processed; the document retention period; details of the information collected; and the rights of the Personal Data Subject.
1.1.2 Checking Protection Level	Article 9 & Article 22 paragraph (1)	The Personal Data Subject has the right to withdraw consent for the processing of their Personal Data previously given to the Data Controller. Consent for Personal Data processing is given through written or recorded consent.

3.4. Discussion

The results answer the research question directly: compliance with Privacy by Design is consistently weak across the evaluated Indonesian QSR applications. As detailed in the results, structural weaknesses were observed in tracker integration, permission over-declaration, and hardcoded sensitive data, exposing broad regulatory implications across multiple UU PDP provisions. The study also closes the three identified gaps: it provides the first MobSF-based audit of Indonesian QSR applications; it demonstrates that technical audits are essential to corroborate normative privacy-policy analyses. Because code-level findings such as hardcoded sensitive data cannot be inferred from policies alone, this technical evidence directly corroborates and extends the work of Mulligan et al. [29], Blancaflor et al. [28], and Khedkar et al. [33]. Furthermore, compared to prior MobSF studies focusing on mobile banking or e-commerce [15], [16], which typically highlight cryptographic flaws as the primary vulnerability, this QSR evaluation reveals that third-party tracking networks and data minimisation failures pose the most severe structural privacy risks in the food-service sector. Finally, it integrates PbD-MASVS-MobSF analysis with the UU PDP at the level of specific technical findings, which prior convergence studies [36], [37] did not attempt.

The consistency of weak data minimisation, credential leakage, and tracker presence across applications of differing corporate scale indicates that these risks are structural-sectoral rather than individual-specific, consistent with arguments that digital transformation and omnichannel strategies drive sector-wide data harvesting [27] and with the broader literature on tracker prevalence and data-minimisation risk [11], [28], [29], [32]. These patterns converge on MASVS-PRIVACY-1 and MASVS-PRIVACY-3, the controls with the widest scope and the highest accumulation of weak findings. Only vulnerable-call detection (1.2.4) varies, suggesting that code-vulnerability handling depends on individual development practices and can be improved by individual teams. In contrast, the structural patterns, rooted in standard SDK suites, permission templates, and credential-handling conventions, will require action by SDK providers, industry associations, regulators, and developers.

The 31 tracker SDK instances identified across the seven applications can be grouped into two broad categories based on their primary functions. The first consists of analytics and crash-reporting tools, including Google Firebase Analytics, Google Crashlytics, Branch, and Sentry, which are used across most applications to support functions such as user engagement measurement and application performance monitoring. The second category includes advertising and profiling SDKs, such as AppsFlyer (Applications D and E), Facebook Analytics (A, E, and G), Facebook Login (A, B, D, E, and G), mParticle (A), and CleverTap (C). Unlike analytics tools, these SDKs are designed to associate user activity with persistent identities across multiple applications and services. As a result, they can facilitate user re-identification by combining information such as order histories, location events, and device-related signals, even when no direct personal identifiers are disclosed. Notably, profiling SDKs appear more frequently in the lower-scoring applications, particularly D and E, suggesting that greater investment in personalised marketing features may be associated with increased privacy exposure. The full anonymised breakdown of tracker SDK categories by application is provided in Appendix B. This observation aligns with previous studies on privacy risks in loyalty-program ecosystems [7], [27].

For individual users, these findings have three important implications. First, advertising SDKs such as Facebook Analytics and AppsFlyer can enable cross-application profiling, allowing marketing networks to infer information such as how often users order food, their dietary preferences, and their preferred brands, often without clear notice. Second, all seven applications declare the ACCESS_FINE_LOCATION permission in their manifests. Under the Android runtime permission model, users must explicitly grant this permission via a system dialogue during application execution. However, static analysis can only confirm the presence of this permission request and the embedding of tracking SDKs within the APK. It cannot definitively prove whether the application

actually transmits location data to these SDKs before the runtime prompt is accepted or only after explicit user consent is granted. Nevertheless, if these embedded SDKs initialise before the consent dialogue, this creates the possibility of continuous geo-behavioural tracking that extends beyond the application's active use. Third, when order histories and location records are combined with device-related signals linked to READ_PHONE_STATE and RECORD_AUDIO permissions, the risk of user re-identification increases. While each data point may appear harmless on its own, their combination can produce a distinctive behavioural profile capable of identifying individuals even within datasets that have been anonymised, a finding consistent with previous research on behavioural re-identification [37].

Static analysis cannot establish whether a privacy violation has occurred because runtime behaviour remains unobserved; thus, the weak indications reflect the absence of observable code-level evidence rather than definitive confirmation of non-compliance in practice. Furthermore, the sample is a single-point-in-time snapshot of specific APK versions, and the UU PDP linkages are indicative and thematic. Future work should therefore combine static and dynamic analysis longitudinally on the same sample to test whether user-interface consent genuinely constrains tracking-SDK activity, replicate the PbD-MASVS-MobSF framework in sectors beyond QSR to assess generalizability, and develop, in cooperation with the UU PDP supervisory authority, a technical audit protocol that can validate and operationalise the indicative mapping in Table 2. for regulatory enforcement.

4. CONCLUSION

This study systematically evaluates compliance with the privacy-by-design principle across seven leading quick-service restaurant (QSR) applications in Indonesia, using the proposed PbD-MASVS-MobSF framework. As its main contribution, this study establishes a transparent and replicable operational framework that links technical evidence at the code level to privacy design principles and Indonesia's Personal Data Protection Act (PDP Act). A key methodological limitation is that this evaluation relies entirely on static analysis: the resulting compliance mapping is thematic and indicates structural privacy risks, rather than a legal determination of non-compliance, which require dynamic runtime observation and, in future research, further investigation. Furthermore, the framework could be extended to other industrial sectors, and a weighted composite score could be developed that integrates the PbD and MASVS dimensions with the MobSF evidence base into a single measurable index. Based on these findings, the practical recommendations are that application developers should consciously reduce harmful permissions to align with the principle of data minimisation, implement secret management to eliminate the permanent embedding of credentials, actively audit third-party SDK data flows, and document runtime consent controls. At the same time, regulators should also strive to ensure the availability of sector-specific technical guidance and standardised audit procedures, rather than relying solely on improvements at the application level.

REFERENCES

[1] Asosiasi Penyelenggara Jasa Internet Indonesia (APJII), "PROFIL INTERNET INDONESIA 2026: Survei Penetrasi Internet dan Perilaku Penggunaan Internet," 2026. Accessed: Jun. 14, 2026. [Online]. Available: https://survei.apjii.or.id/survei/group/12

[2] D. E. Mahameru, A. Nurhalizah, H. Badjeber, A. Wildan, and H. Rahmadia, "Implementasi UU Perlindungan Data Pribadi Terhadap Keamanan Informasi Identitas di Indonesia," *Esensi Hukum*, vol. 5, no. 2, pp. 115–131, 2024, doi: 10.35586/esensihukum.v5i2.240.

[3] S. Kemp, "Digital 2026 Mid-Year Global Update Report," Apr. 2026. Accessed: May 30, 2026. [Online]. Available: https://datareportal.com/reports/digital-2026-mid-year-global-update-report

[4] S. Kemp, "Digital 2026: Indonesia — DataReportal – Global Digital Insights," Nov. 2026. Accessed: Jun. 14, 2026. [Online]. Available: https://datareportal.com/reports/digital-2026-indonesia

[5] E. Lee, J. Kim, J. Kim, and C. Koo, "Information Privacy Behaviors during the COVID-19 Pandemic: Focusing on the Restaurant Context," *Information Systems Frontiers*, vol. 25, no. 5, pp. 1829–1845, 2023, doi: 10.1007/s10796-022-10321-1.

[6] J. Sutrisno, A. Tarigan, and I. D. C. Purnomo, "Understanding the role of gamification and loyalty programs in restaurant apps: a systematic literature review and conceptual framework development," *Journal of Business & Applied Management*, vol. 19, no. 1, 2026, doi: 10.30813/jbam.v19i1.9775.

[7] U. D. Amama, F. Talgatuly, A. F. Kolawole, and O. Oluwatobi, "Data Engineering and Privacy Challenges in Loyalty Card Programs: Insights from Retail, Banking, and Hospitality," *Asian Journal of Advanced Research and Reports*, vol. 18, no. 11, pp. 340–357, 2024, doi: 10.9734/ajarr/2024/v18i11800.

[8] J. Singh, "Bugs in a major McDonald's India delivery system exposed sensitive customer data | TechCrunch," TechCrunch Media LLC. Accessed: May 31, 2026. [Online]. Available:

https://techcrunch.com/2024/12/19/bugs-in-a-major-mcdonalds-india-delivery-system-exposed-sensitive-customer-data/

[9] L. Abrams, "Domino's India discloses data breach after hackers sell data online," Bleeping Computer® LLC. Accessed: May 31, 2026. [Online]. Available: https://www.bleepingcomputer.com/news/security/dominos-india-discloses-data-breach-after-hackers-sell-data-online/

[10] Y. Yokotani, A. C. Kurnia, W. Wirazilmustaan, and R. D. Salfutra, "Pengendalian Data Pribadi Dan Ruang Siber Oleh Platform Digital Big Data Global Terkait Kedaulatan Digital Indonesia," *PROGRESIF: Jurnal Hukum*, vol. 19, no. 1, pp. 70–80, 2025, doi: 10.33019/cdt5p455.

[11] K. Kollnig *et al.*, "Before and after GDPR: tracking in mobile apps," *Internet Policy Review*, vol. 10, no. 4, 2021, doi: 10.14763/2021.4.1611.

[12] Z. R. Alkindi, M. Sarrab, and N. Alzeidi, "User Privacy and Data Flow Control for Android Apps: Systematic Literature Review," *Journal of Cyber Security and Mobility*, 2021, doi: 10.13052/jcsm2245-1439.1019.

[13] A. Abraham, "AppSec PNW: Android and iOS Application Security with MobSF," 2024. Accessed: May 30, 2026. [Online]. Available: https://mobsf.github.io/Mobile-Security-Framework-MobSF/presentations.html

[14] "OWASP MASVS - OWASP Mobile Application Security." Accessed: May 31, 2026. [Online]. Available: https://mas.owasp.org/MASVS/

[15] R. A. Faozi, N. W. A. Majid, and S. Widodo, "Security Maturity Assessment of Indonesian Android Mobile Banking Apps using MobSF and OWASP," *Edumatic: Jurnal Pendidikan Informatika*, vol. 10, no. 1, pp. 80–89, Mar. 2026, doi: 10.29408/edumatic.v10i1.33285.

[16] U. Kishnani and S. Das, "Security and Privacy Assessment of U.S. and Non-U.S. Android E-Commerce Applications," in *Springer*, 2026, pp. 444–454. doi: 10.1007/978-3-032-13714-2_27.

[17] S. A. Khan *et al.*, "An Android Applications Vulnerability Analysis Using MobSF," in *Proceedings - 2024 International Conference on Engineering and Computing, ICECT 2024*, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/ICECT61618.2024.10581312.

[18] J. Zhu, K. Li, S. Chen, L. Fan, J. Wang, and X. Xie, "A Comprehensive Study on Static Application Security Testing (SAST) Tools for Android," *IEEE Transactions on Software Engineering*, vol. 50, no. 12, pp. 3385–3402, 2024, doi: 10.1109/TSE.2024.3488041.

[19] P. Nur Izzati and K. Kasmawi, "Static Analysis-Based Security Enhancement for Mobile Applications Using Mobile Security Framework (MOBSF)," *Journal of Applied Informatics and Computing*, vol. 9, no. 4, pp. 1272–1279, 2025, doi: 10.30871/jaic.v9i4.9525.

[20] A. Cavoukian, "Privacy by design: the definitive workshop. A foreword by Ann Cavoukian, Ph.D," *Identity in the Information Society*, vol. 3, no. 2, pp. 247–251, 2010, doi: 10.1007/s12394-010-0062-y.

[21] V. C. Andrade, R. D. Gomes, S. Reinehr, C. O. D. A. Freitas, and A. Malucelli, "Privacy by Design and Software Engineering," in *Proceedings of the XXI Brazilian Symposium on Software Quality*, New York, NY, USA: ACM, Nov. 2022, pp. 1–10. doi: 10.1145/3571473.3571480.

[22] S. A. de Chaves and F. Barreto Vavassori Benitti, "Privacy by Design in Software Engineering: An update of a Systematic Mapping Study," in *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*, New York, NY, USA: ACM, Mar. 2023, pp. 1362–1369. doi: 10.1145/3555776.3577626.

[23] C. Del-Real, E. De Busser, and B. van den Berg, "A systematic literature review of security and privacy by design principles, norms, and strategies for digital technologies," *International Review of Law, Computers & Technology*, vol. 39, no. 3, pp. 374–405, 2025, doi: 10.1080/13600869.2025.2457227.

[24] M. Hatamian, S. Wairimu, N. Momen, and L. Fritsch, "A privacy and security analysis of early-deployed COVID-19 contact tracing Android apps," *Empir. Softw. Eng.*, vol. 26, no. 3, p. 36, 2021, doi: 10.1007/s10664-020-09934-4.

[25] A. E. Waldman, "Data Protection by Design? A Critique of Article 25 of the GDPR," *Yale Journal of Law & Technology*, 2021, [Online]. Available: https://ssrn.com/abstract=3773143

[26] R. Alt, "Digital Transformation in the Restaurant Industry: Current Developments and Implications," *Journal of Smart Tourism*, vol. 1, no. 1, pp. 69–74, 2021, doi: 10.52255/smarttourism.2021.1.1.9.

[27] T. Sze and G. Gunasekara, "A Privacy Trojan Horse? Consumer Loyalty Programmes in the Grocery Sector and Data Privacy," *SSRN Electronic Journal*, 2025, doi: 10.2139/ssrn.5628590.

[28] E. Blancaflor, I. Z. L. Delfin, A. Molate, A. I. Penafiel, A. Villaluz, and A. K. Balan, "Ethical Use of Geolocation Data: Privacy Concerns in Food Delivery Applications in the Philippines," in *Proceedings of the 2024 7th International Conference on Information Management and Management Science*, New York, NY, USA: ACM, Aug. 2024, pp. 346–353. doi: 10.1145/3695652.3695683.

[29] C. Mulligan, G. Gillis, L. Remedios, C. Parsons, L. Vergeer, and M. Potvin Kent, "Children's digital privacy on fast-food and dine-in restaurant mobile applications," *PLOS Digital Health*, vol. 4, no. 2, p. e0000723, 2025, doi: 10.1371/journal.pdig.0000723.

[30] Z. Q. Ferdyan Putri and Y. Efawati, "Exploring the Impact of Customer Data Security on Consumer Trust in Gojek's Digital Services," *International Journal Administration, Business & Organization*, vol. 6, no. 1, pp. 136–145, 2025, doi: 10.61242/ijabo.25.332.

[31] A. R. Fadillah and A. Primajaya, "Analisis Keamanan Aplikasi Cafe Berbasis Android Menggunakan Mobile Security Framework (MobSF)," *Jurnal Ilmiah Wahana Pendidikan*, 2026, [Online]. Available: http://www.jurnal.peneliti.net/index.php/JIWP/article/view/12754

[32] S. Koch, M. Karl, R. Kirchner, M. Wessels, A. Paschke, and M. Johns, "The Impact of Default Mobile SDK Usage on Privacy and Data Protection," *Proceedings on Privacy Enhancing Technologies*, vol. 2025, no. 1, pp. 808–823, 2025, doi: 10.56553/popets-2025-0042.

[33] M. Khedkar, A. Kumar Mondal, and E. Bodden, "A study of privacy-related data collected by Android apps," *Automated Software Engineering*, vol. 33, no. 2, p. 45, 2026, doi: 10.1007/s10515-025-00589-3.

[34] Republik Indonesia, *Undang-Undang Republik Indonesia Nomor 27 Tahun 2022 tentang Pelindungan Data Pribadi*. Indonesia, 2022.

[35] F. Albirra, M. J. Soesapto, L. Agata, A. M. Pribadi, Istijanto, and Lydia Apriliani, "The Factors Influencing Online Food Delivery Usage Intention on Semi-Endemic Period," *TIJAB (The International Journal of Applied Business)*, vol. 7, no. 2, pp. 134–151, 2023, doi: 10.20473/tijab.v7.I2.2023.44104.

[36] D. S. Guaman, J. M. Del Alamo, and J. C. Caiza, "GDPR Compliance Assessment for Cross-Border Personal Data Transfers in Android Apps," *IEEE Access*, vol. 9, pp. 15961–15982, 2021, doi: 10.1109/ACCESS.2021.3053130.

[37] D. S. Guamán, D. Rodriguez, J. M. del Alamo, and J. Such, "Automated GDPR compliance assessment for cross-border personal data transfers in android applications," *Comput. Secur.*, vol. 130, p. 103262, 2023, doi: 10.1016/j.cose.2023.103262.

[38] T. Fedynyshyn, O. Mykhaylova, and I. Opirskyy, "Security Implications of Mobile Development Frameworks: Findings from Static Analysis of Android Apps," in *2024 IEEE 17th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET)*, IEEE, Oct. 2024, pp. 444–448. doi: 10.1109/TCSET64720.2024.10755684.

[39] C. Tila, "Leading Official Restaurant Apps in Indonesia in 2023, by Downloads (in 1,000s)," Apr. 2024. [Online]. Available: https://www.statista.com/statistics/1461199/indonesia-leading-official-restaurant-apps-by-downloads/

[40] V. Kouliaridis, G. Karopoulos, and G. Kambourakis, "Assessing the Security and Privacy of Android Official ID Wallet Apps," *Information*, vol. 14, no. 8, p. 457, 2023, doi: 10.3390/info14080457.

APPENDIX A. FULL UU PDP ARTICLE MAPPING

This appendix reproduces the complete Section 3.3 mapping between MobSF PbD subprocesses and UU PDP (Law No. 27 of 2022) articles, including the exact subprocess description and the verbatim, PDF-verified article text underlying each row of Table 2.

Table A1. Full indicative association of MobSF subprocesses with relevant UU PDP articles (with verbatim quotations)

MobSF Subprocess (PbD)	UU PDP Article	MobSF Sub-Process Quote / UU PDP Article Quote (Verbatim, PDF-Verified)
1.2.1 Identifying Trackers	Article 37	**MobSF Sub-Process Quote:** "Identifies tracker libraries (tracker SDKs) embedded in the APK code using the Exodus Privacy / ETIP database to detect the presence of third-party trackers." **UU PDP Article Quote:** "The Data Controller must supervise every party involved in the Processing of Personal Data under the Data Controller's control."
1.2.4 Determining Vulnerable Call	Article 35 point a	**MobSF Sub-Process Quote:** "Determines potentially vulnerable code calls based on known security vulnerability patterns, such as the use of weak encryption algorithms or insecure APIs." **UU PDP Article Quote:** "...to protect and ensure the security of the Personal Data it processes, by: a. preparing and implementing technical operational

Idealis: Indonesia Journal Information System
Volume 9, Nomor 2, Juli 2026
ISSN 2684-7280 (*online*)
Halaman 302-313
available online at http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index

MobSF Subprocess (PbD)	UU PDP Article	MobSF Sub-Process Quote / UU PDP Article Quote (Verbatim, PDF-Verified)
		measures to protect Personal Data from disruptions to Personal Data processing that are contrary to statutory provisions; ..."
1.1.1 Checking Permissions	Article 27	**MobSF Sub-Process Quote:** "Checks the list of permissions requested by the APK in AndroidManifest.xml to identify permissions that are dangerous or excessive relative to the required standard." **UU PDP Article Quote:** "The Data Controller must process Personal Data in a limited and specific, lawful, and transparent manner."
1.1.3 Checking Configuration	Article 39 paragraphs (1)-(2)	**MobSF Sub-Process Quote:** "Checks manifest configurations such as the debuggable flag, allowBackup, exported components, and network security policy settings (Network Security Config)." **UU PDP Article Quote:** "(1) The Data Controller must prevent unauthorized access to Personal Data. (2) The prevention referred to in paragraph (1) is carried out by building a security system for Personal Data processed and/or used to process Personal Data using electronic systems in a reliable, secure, and accountable manner."
1.2.3 Extracting Methods	Article 34 paragraph (2) point d	**MobSF Sub-Process Quote:** "Extracts the list of methods and functions called in the code (method extraction) as material for vulnerability analysis and further data flow mapping." **UU PDP Article Quote:** "...the processing of Personal Data for systematic evaluation, scoring, or monitoring activities of the Personal Data Subject;"
2.1 Searching Method Calls	Article 16 paragraph (1)	**MobSF Sub-Process Quote:** "Searches for and maps all method calls in the codebase to trace the data flow path from entry point (source) to exit point (sink)." **UU PDP Article Quote:** "The Processing of Personal Data includes: a. acquisition and collection; b. processing and analysis; c. storage; d. correction and updating; e. display, announcement, transfer, dissemination, or disclosure; and/or f. erasure or destruction."
2.2 Generating Call Graphs	Article 31	**MobSF Sub-Process Quote:** "Builds a call graph to visualize the code execution flow and dependencies between components, so that data leakage paths can be identified." **UU PDP Article Quote:** "The Data Controller must record all Personal Data Processing activities."
2.3 Taint Analysis	Article 38	**MobSF Sub-Process Quote:** "Analyzes the flow of sensitive data from source to sink using taint analysis to detect leaks of user privacy data sent to third parties or stored without encryption." **UU PDP Article Quote:** "The Data Controller must protect Personal Data from unauthorized processing."
1.2.2 Matching Sensitive Data	Article 21 paragraph (1)	**MobSF Sub-Process Quote:** "Matches patterns of sensitive data, such as API keys, hardcoded credentials, tokens, and personal data, in the source code using regular expression (regex) matching." **UU PDP Article Quote:** "...the Data Controller must provide Information regarding: a. the legality of the Personal Data processing; b. the purpose of the Personal Data processing; c. the type and relevance of the Personal Data to be processed; d. the retention period of documents containing Personal Data; e. details of the Information collected; f. the period of Personal Data processing; and g. the rights of the Personal Data Subject."
1.1.2 Checking Protection Level	Article 9 & Article 22 paragraph (1)	**MobSF Sub-Process Quote:** "Checks the protection level (normal, dangerous, signature) of each requested permission to assess the access risk granted to the application." **UU PDP Article Quote:** "The Personal Data Subject has the right to withdraw consent for the processing of their Personal Data previously given to

Idealis: Indonesia Journal Information System
Volume 9, Nomor 2, Juli 2026
ISSN 2684-7280 (*online*)
Halaman 302-313
available online at http://jom.fti.budiluhur.ac.id/index.php/IDEALIS/index

MobSF Subprocess (PbD)	UU PDP Article	MobSF Sub-Process Quote / UU PDP Article Quote (Verbatim, PDF-Verified)
		the Data Controller. Consent for Personal Data processing is given through written or recorded consent."

APPENDIX B. TRACKER SDK CATEGORIES BY APPLICATION

This appendix presents the categorisation of tracker SDK instances identified across the seven QSR applications discussed in Section 3.4, grouped by primary function (analytics/crash-reporting versus advertising/profiling) and cross-referenced to the anonymised application labels (A–G) used throughout this manuscript. No brand-specific identities are disclosed.

Table B1. Tracker SDK categories and application presence (anonymised labels A–G)

Tracker SDK	**Category**	**Applications Present (Anonymized)**
Google Firebase Analytics	Analytics & Crash-Reporting	Present across most applications*
Google Crashlytics	Analytics & Crash-Reporting	Present across most applications*
Branch	Analytics & Crash-Reporting	Present across most applications*
Sentry	Analytics & Crash-Reporting	Present across most applications*
AppsFlyer	Advertising & Profiling	D, E
Facebook Analytics	Advertising & Profiling	A, E, G
Facebook Login	Advertising & Profiling	A, B, D, E, G
mParticle	Advertising & Profiling	A
CleverTap	Advertising & Profiling	C

**Exact per-application distribution for these four analytics/crash-reporting SDKs was not individually itemised in the underlying MobSF extraction summarised in Section 3.4; only the aggregate presence pattern across the sample is reported here, to avoid overstating precision beyond what the source data support.*