# Meeting Summary

# Reasoning Process

Below is my step-by-step reasoning process for analyzing the provided transcript from the User Requirements Specification (URS) meeting. After this, I’ll present the structured summary based on the required format.

### Step-by-Step Reasoning Process:

1. **Review the Transcript**: I first read through the entire transcript to understand the context. It’s a discussion about developing a dashboard for “tunai” (cash), involving data analysis, mapping, and feedback. Key topics include data sources (e.g., BC add, BKKPM), dashboard features, potential issues with data matching, and integration with systems like VRM and SAP. Participants include Speaker A, B, C, and references to others like Puan Sabariah and Puan Hashimah. The meeting focuses on requirements for software development, such as functional features, data handling, and performance considerations.
2. **Identify Key Sections**: Using the specified structure, I mapped the transcript content:
   * **Meeting Overview**: Extract date (not explicitly stated, but inferred as recent), participants, and objectives from the opening and closing remarks.
   * **Functional Requirements**: Look for specific features like dashboard creation, data filtering, matching, and sampling. Assign unique identifiers (e.g., FR-01) and note priorities (e.g., critical for data accuracy). Check for dependencies, such as reliance on external systems.
   * **Non-Functional Requirements**: Identify aspects like security (data protection), performance (processing time for large datasets), usability (user-friendly interfaces), and compatibility (with SAP or VRM). Assign identifiers (e.g., NFR-01).
   * **Business Rules & Constraints**: Note regulatory requirements (e.g., VRM compliance), organizational policies (e.g., using 2024 data), and technical limitations (e.g., data size constraints).
   * **Data Requirements**: Extract mentions of data structures (e.g., columns like SG text, unique keys), integration points (e.g., with BC add), and validation rules (e.g., inner joins for matching).
   * **Open Questions & Decisions Required**: Flag unclear items (e.g., data matching ambiguities) and conflicts (e.g., discrepancies between systems).
   * **Next Steps & Action Items**: Identify action items (e.g., sharing documents), responsibilities (e.g., Speaker A for development), and deadlines (e.g., implied short-term).
3. **Organize and Prioritize Information**:
   * Use bullet points for clarity and preserve exact terminology (e.g., “tunai”, “VRM”, “inner join”).
   * Highlight ambiguities or problematic areas (e.g., data matching issues that could lead to inaccuracies).
   * Assign identifiers systematically: FR-01 for functional, NFR-01 for non-functional.
   * Ensure priorities are noted where mentioned (e.g., critical for core features).
   * Cross-check for dependencies (e.g., dashboard development depends on data from multiple sources).
4. **Handle Edge Cases**:
   * The transcript is in Malay with some English terms; I’ll translate key terms for clarity in the summary while preserving original phrasing in quotes.
   * Some requirements are implied (e.g., performance expectations from discussions on processing time), so I’ll infer based on context.
   * Ambiguities: Mark items like unresolved data matching as potentially problematic for implementation.
   * Completeness: Cover all sections, even if some have limited content (e.g., no explicit deadlines, so note as “not specified”).
5. **Format the Summary**:
   * Structure exactly as instructed: Start with Meeting Overview, then proceed sequentially.
   * Use bullet points for lists.
   * End with any highlights of issues.

Now, I’ll generate the final structured summary.

# Meeting Summary

### Summary of URS Meeting Analysis

#### 1. Meeting Overview

* **Date and Participants**: The meeting date is not explicitly stated, but it appears to be a recent session based on references to 2024 data. Participants include Speaker A (leading the discussion on dashboard development), Speaker B (providing feedback and introductions, e.g., to Puan Sabariah and Puan Hashimah), Speaker C, and other stakeholders like TKNK and VRM representatives.
* **Main Objectives of the Discussion**: The session focused on reviewing and refining a dashboard for “tunai” (cash), discussing data development from 2024 (initially for three months), gathering feedback on analyses, and addressing potential enhancements. The goal was to “level up” the dashboard based on data analysis, ensure accurate mapping, and prepare for full-year data integration.

#### 2. Functional Requirements

* **FR-01**: Develop a dashboard for tunai transactions, including profiling by categories like liability, equity, expenses, assets, and revenue, with trends for three months initially (Priority: Critical, as it forms the core of the discussion; Dependency: Relies on data from BC add and BKKPM for mapping).
* **FR-02**: Implement data mapping and matching using unique keys (e.g., combination of document number, Fcomp, and date) to identify tunai transactions in detail tables (Priority: High; Dependency: Depends on inner joins and SG text column for accuracy, which seems ambiguous in implementation).
* **FR-03**: Add filtering and visualization features, such as by expenditure type, agency, or account, with drill-down capabilities (e.g., from summary to transaction details) (Priority: Medium; Dependency: Requires complete data integration).
* **FR-04**: Enable comparisons (e.g., year-over-year, monthly trends) and sampling for audits, including materiality analysis and random sampling based on percentages (Priority: High; Dependency: Needs full dataset, potentially problematic if data is incomplete).
* **FR-05**: Generate reports for specific analyses, like outliers in accounts or budget variances (Priority: Medium; Dependency: Relies on external data sources like VRM).

#### 3. Non-Functional Requirements

* **NFR-01**: Performance expectations – Reduce processing time for large datasets (e.g., three months of data is “quite besar” and needs optimization for full-year data) (Potential issue: Implementation may be problematic if data volume causes delays).
* **NFR-02**: Usability considerations – Dashboard should be user-friendly with filters, drill-downs, and options for time periods (e.g., monthly or yearly comparisons), as discussed for auditors (e.g., “don’t use technical terms too much”).
* **NFR-03**: Security requirements – Ensure secure handling of sensitive data, such as financial transactions from systems like SAP and VRM (Potential issue: Ambiguous dependencies on external systems could introduce vulnerabilities).
* **NFR-04**: Compatibility needs – Ensure compatibility with systems like SAP, BC add, and VRM for data integration (e.g., matching SG text across tables).

#### 4. Business Rules & Constraints

* **Regulatory Requirements**: Compliance with VRM (e.g., requirements for risk 17 and account profiling), as mentioned in discussions with TKNK; also, adherence to audit processes for tunai and accrual data.
* **Organization-Specific Policies**: Use 2024 data initially for three months to speed up development, then expand to full-year data; policies on data mapping and inner joins to relate tunai with accrual transactions.
* **Technical Limitations Discussed**: Processing constraints for large datasets (e.g., “data tiga bulan pun dah quite besar”), potential issues with data matching (e.g., unmapped transactions), and reliance on manual verification for some elements (e.g., “ada yang tak matching”).

#### 5. Data Requirements

* **Data Structures Mentioned**: Structures include columns like SG text (combination of document number, Fcomp, and date), unique keys for matching, and tables from BKKPM (header data) and BC add (detail and tunai data).
* **Integration Points with Other Systems**: Integration with SAP for audit data, VRM for requirements, and systems like BC add and BKKPM for full data sets; also, potential mapping with external sources for complete transactions.
* **Data Validation Rules**: Use inner joins and matching based on SG text for validation; rules for identifying tunai transactions (e.g., “intersect antara dua table”) and handling unmapped items (e.g., “transaksi yang tak di mapping kan lagi”).

#### 6. Open Questions & Decisions Required

* **Items Requiring Further Clarification**: How to handle unmapped transactions (e.g., “ada yang tak matching” in data joins, potentially problematic for accuracy); confirmation on data sources for memorandum items (e.g., “adakah semua dana akaun memorandum ni transaksi dia ada dalam accrual?”); and details on VRM requirements (e.g., “VRM tu VRM nombor-nombor minta maaf lah sebab kita tak ada”).
* **Conflicting Viewpoints Needing Resolution**: Discrepancies between systems (e.g., “content sama tapi data tu akan berbeza” between tunai and accrual); debates on data classification (e.g., whether to use “liability” and “equity” for tunai, as it might not align with audit needs).

#### 7. Next Steps & Action Items

* **Action Items and Responsibilities**: Speaker A to share VRM documents and refine dashboard development (e.g., “boleh tak Rashid share VRM?”); Speaker B to provide feedback and verify data mappings; all participants to review and update analyses based on three months of data (Responsible: Speaker A and team).
* **Deadlines**: Not explicitly mentioned, but implied as short-term (e.g., “sebulan tu mungkin kita boleh keluarkan” for testing, and “tiga minggu” for dashboard refinements); next meeting or updates expected soon, such as sharing slides by Monday.