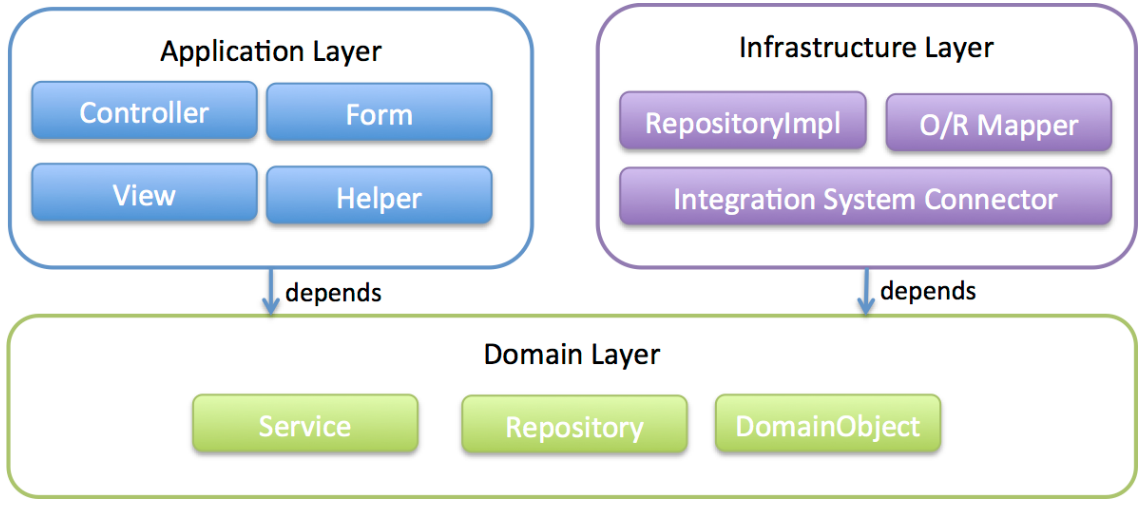
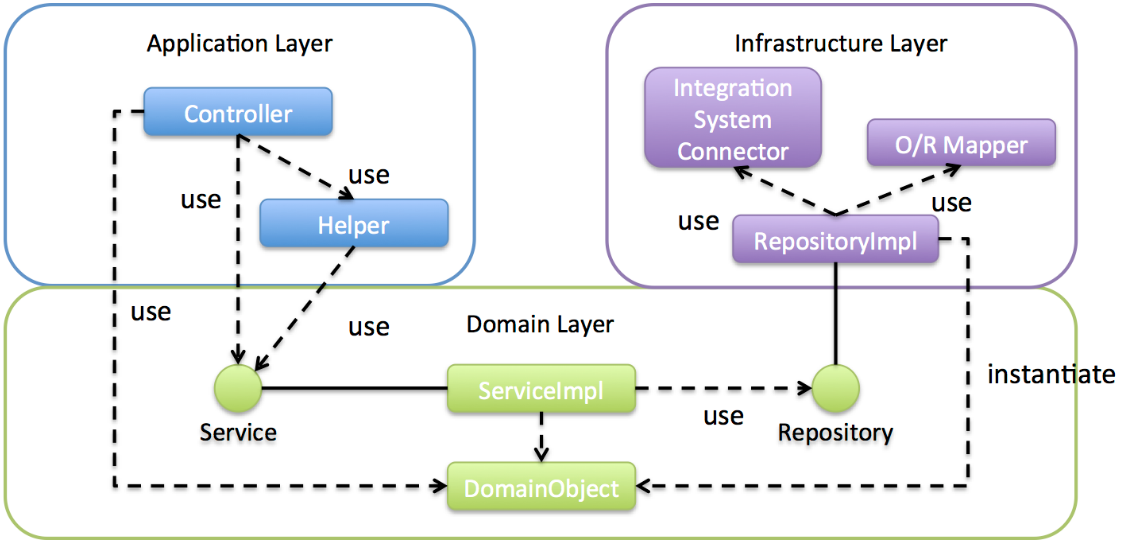
**APPLICATION STRUCTURE**

1. **Application Layering**



1. **Dependency Between Layer**



1. **Processing And Data Flow**

Timeline

Description automatically generated

| **No.** | **Description** |
| --- | --- |
| 1 | Controller menerima Request. |
| 2 | (Optional) Controller memanggil Helper dan mengubah informasi Form menjadi DomainObject atau DTO. |
| 3 | Controller memanggil Service dengan memakai DomainObject atau DTO. |
| 4 | Service memanggil Repository dan mengeksekusi business logic. |
| 5 | Repository memanggil O/R Mapper dan persists DomainObject atau DTO. |
| 6 | (Implementation dependency) O/R Mapper menyimpan informasi DomainObject atau DTO ke DB. |
| 7 | Service mengembalikan DomainObject atau DTO yang merupakan hasil dari eksekusi dari business logic ke Controller. |
| 8 | (Optional) Controller memanggil Helper dan mengubah DomainObject atau DTO menjadi Form. |
| 9 | Controller mengembalikan nama View sebagai tujuan transisi. |
| 10 | View menampilkan Response. |

source: <https://terasolunaorg.github.io/guideline/5.4.1.RELEASE/en/Overview/ApplicationLayering.html>

**PROJECT STRUCTURE**

1. **[projectName]-domain**



| No. | Details |
| --- | --- |
| (1) | Package to store configuration elements of domain layer. |
| (2) | Package to store DomainObjects classes. |
| (3) | Package to store Repository interfaces.  Create a separate package for each Entity. If there are associated Entities to the main entity, then Repository interfaces of associated Entities must also be placed in the same package as main Entity. (For example, Order and OrderLine). If DTO(holds such as search criteria) is also required, it too must be placed in this package.  RepositoryImpl belongs to Infrastructure layer; however, there is no problem in keeping it in this project. In case of using different data stores or existance of multiple persistence platforms, RepositoryImpl class must be kept in separate project or separate package so that implementation related details are concealed. |
| (4) | Package to store Service classes.  Package must be created based on Entity Model or other functional unit. Interface and Implementation class must be kept at the same level of package. If input/output classes are also required, then they must be placed in this package. |
| (5) | Bean definition for CodeList. |
| (6) | Bean definition pertaining to domain layer. |
| (7) | Bean definition pertaining to infrastructure layer. |

1. **[projectName]-web**

A picture containing timeline

Description automatically generated

| No, | Details |
| --- | --- |
| (1) | Package to store configuration elements of application layer. |
| (2) | Bean definited related to the entire application. |
| (3) | Define the properties to be used in the application. |
| (4) | Bean definitions related to Spring MVC. |
| (5) | Bean definitions related to Spring Security |
| (6) | Define the messages and other contents to be used for screen display (internationalization). |
| (7) | Store static resources(css、js、image, etc) |
| (8) | Store View(jsp) files. |
| (9) | Servlet deployment definitions |

1. **[projectName]-env**

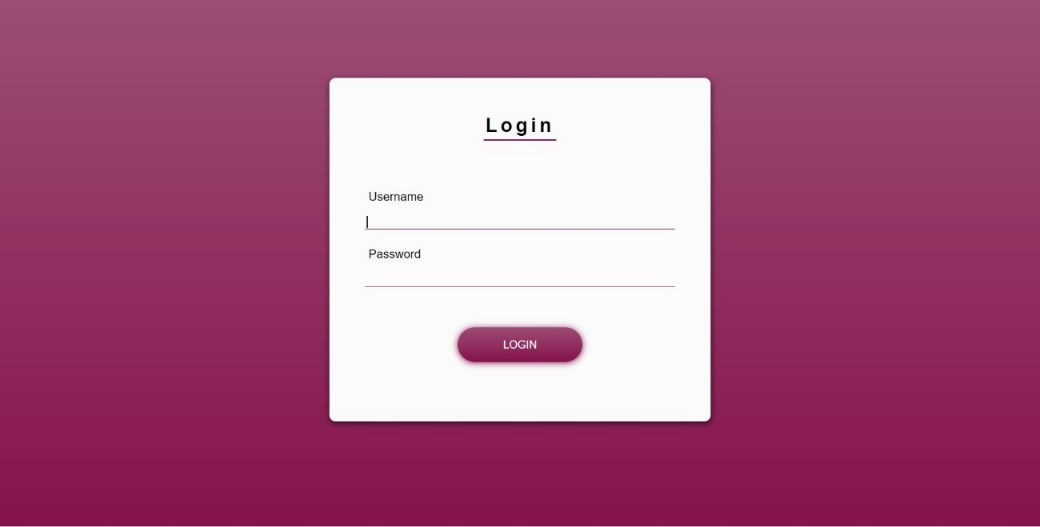
Text

Description automatically generated

| No. | Details |
| --- | --- |
| (1) | Directory to define configurations depends on the environment for all environments. |
| (2) | Directory to define configurations depeands on each environment.  The directory name is used as the name to identify the environment. |
| (3) | Directory to define configurations depeands on each environment.  The sub directory structure and files are same as (4). |
| (4) | Directory to define configurations depeands on the local development environment. |
| (5) | Bean definitions that depend on the local development environment (like dataSource etc). |
| (6) | Property definitions which depend on the local development environment. |
| (7) | Log output definitions which depend on the local development environment. |

**IMPLEMENTATIONS & RESULT**

**1. Login**



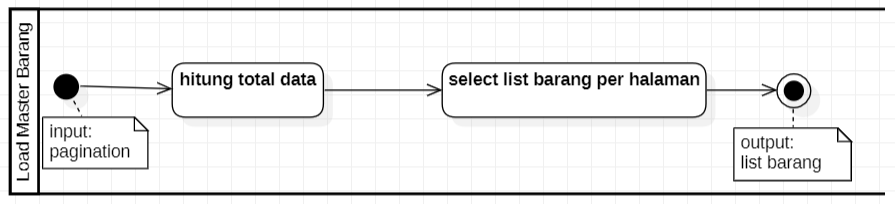
**2. Master Barang**

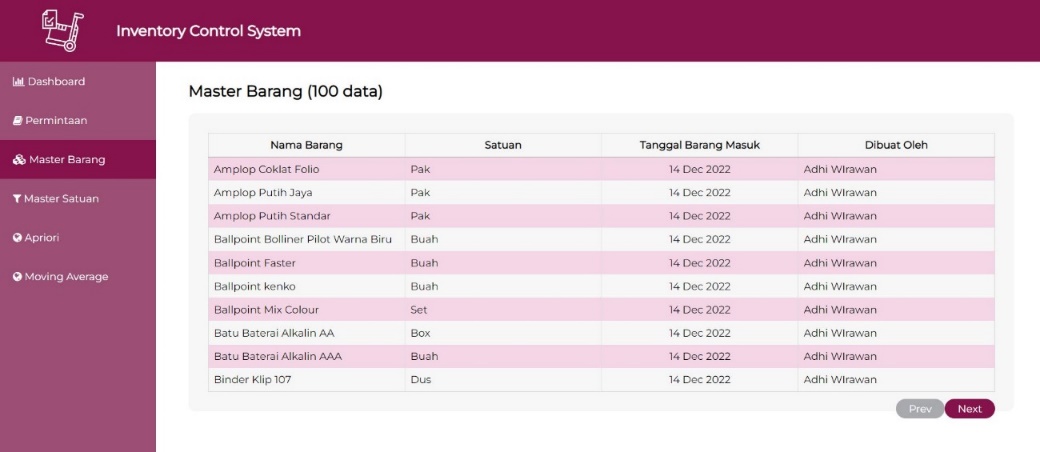
url : ${pageContext.request.contextPath}/item/load

controller : ItemController.java

method : load()

service : ItemLoadServiceImpl.java

proses : 

screen : 

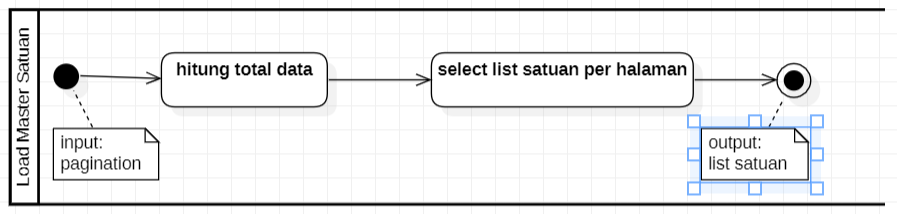
**3. Master Satuan**

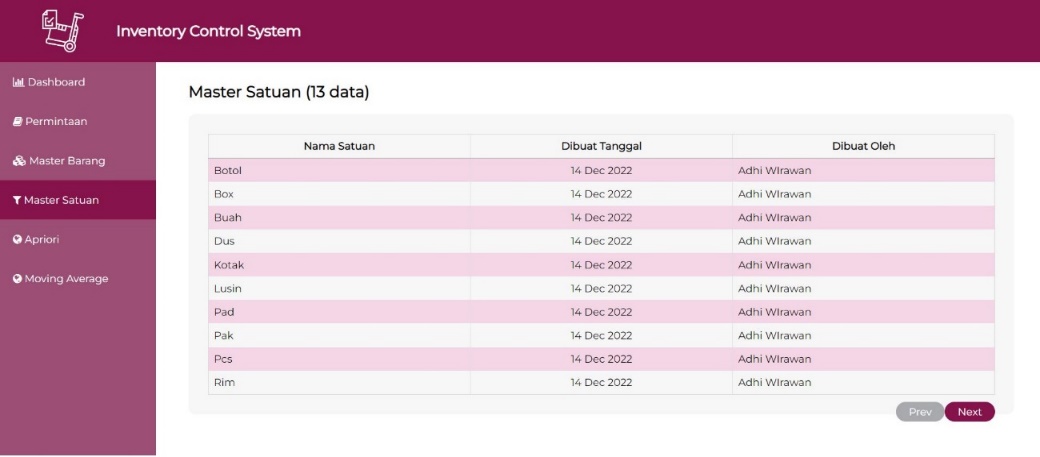
url : ${pageContext.request.contextPath}/unit/load

controller : UnitController.java

method : load()

service : UnitLoadServiceImpl.java

proses : 

screen : 

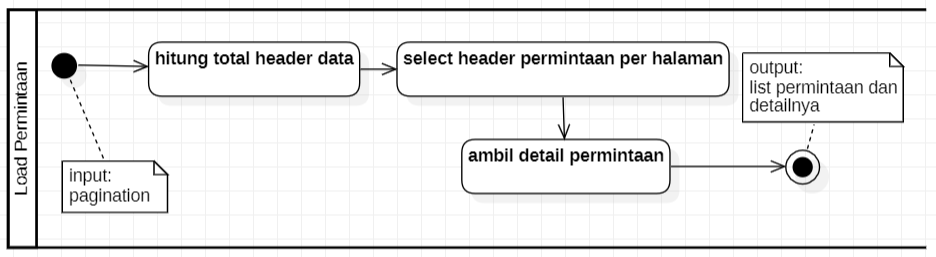
**4. Permintaan**

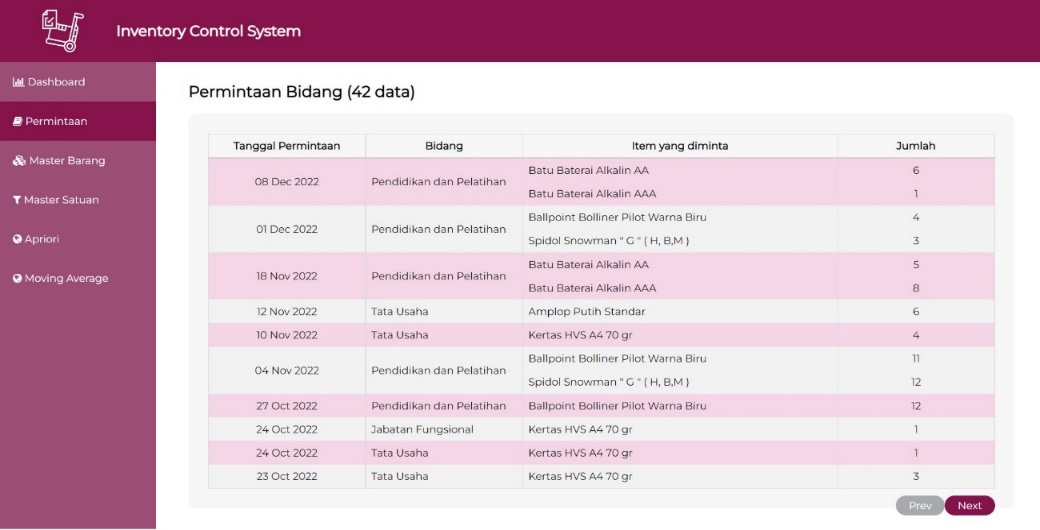
url : ${pageContext.request.contextPath}/request/load

controller : RequestController.java

method : load()

service : RequestLoadServiceImpl.java

proses : 

screen : 

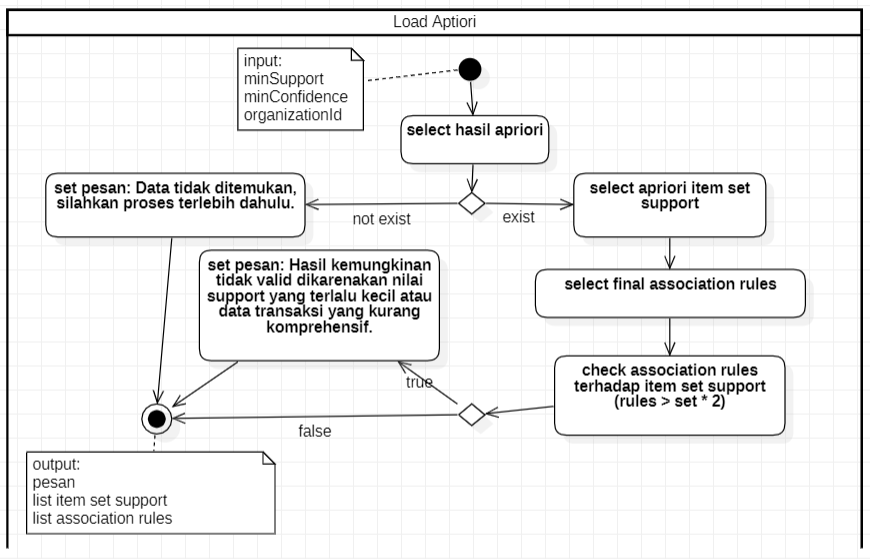
**5.Apriori**

url : ${pageContext.request.contextPath}/apriori/load

controller : AprioriController.java

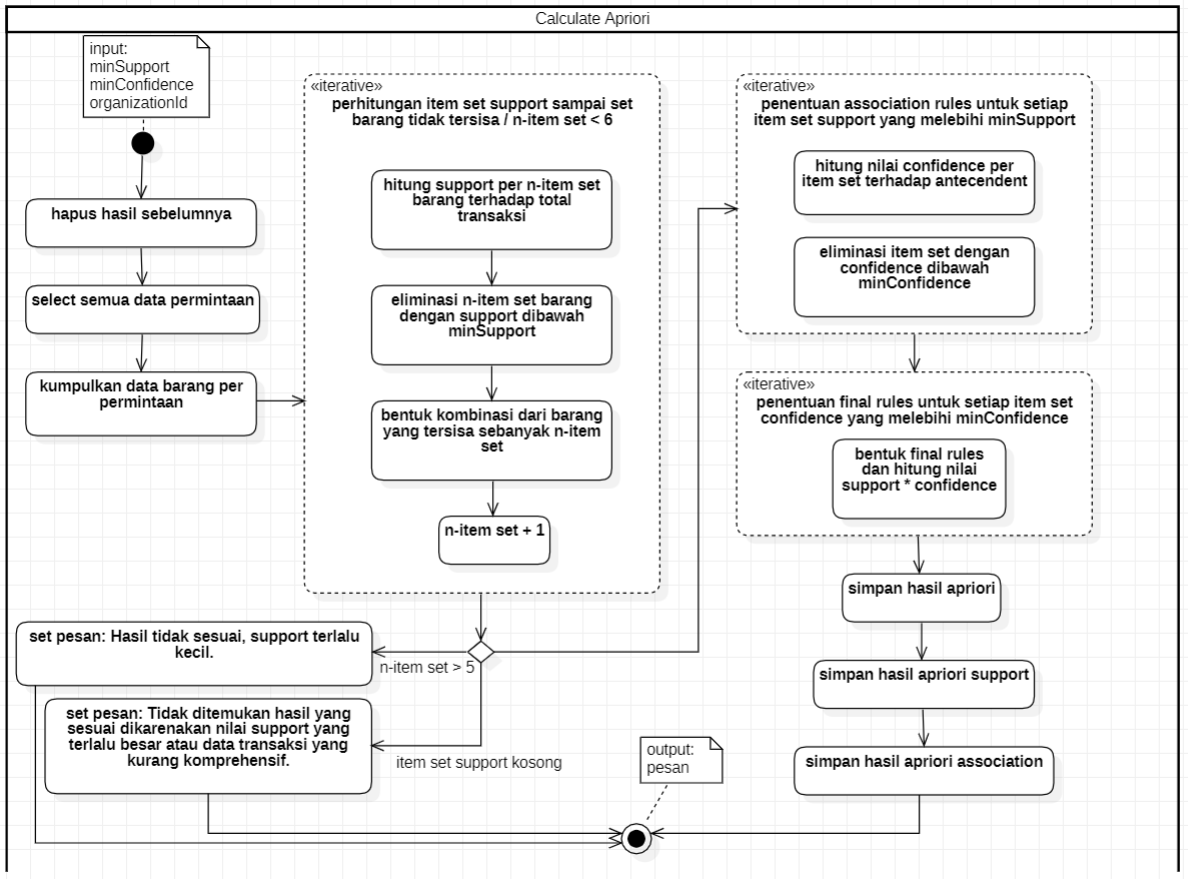
method : load()

service : AprioriLoadServiceImpl.java

proses : 

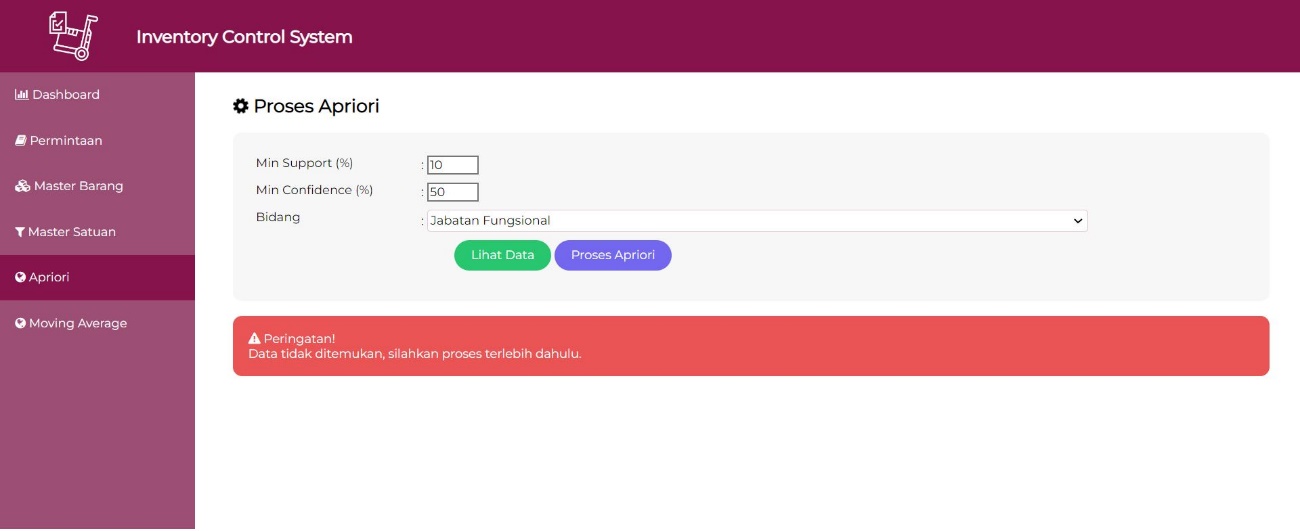
method : processCalculate()

service : AprioriCalculateServiceImpl.java

proses : 

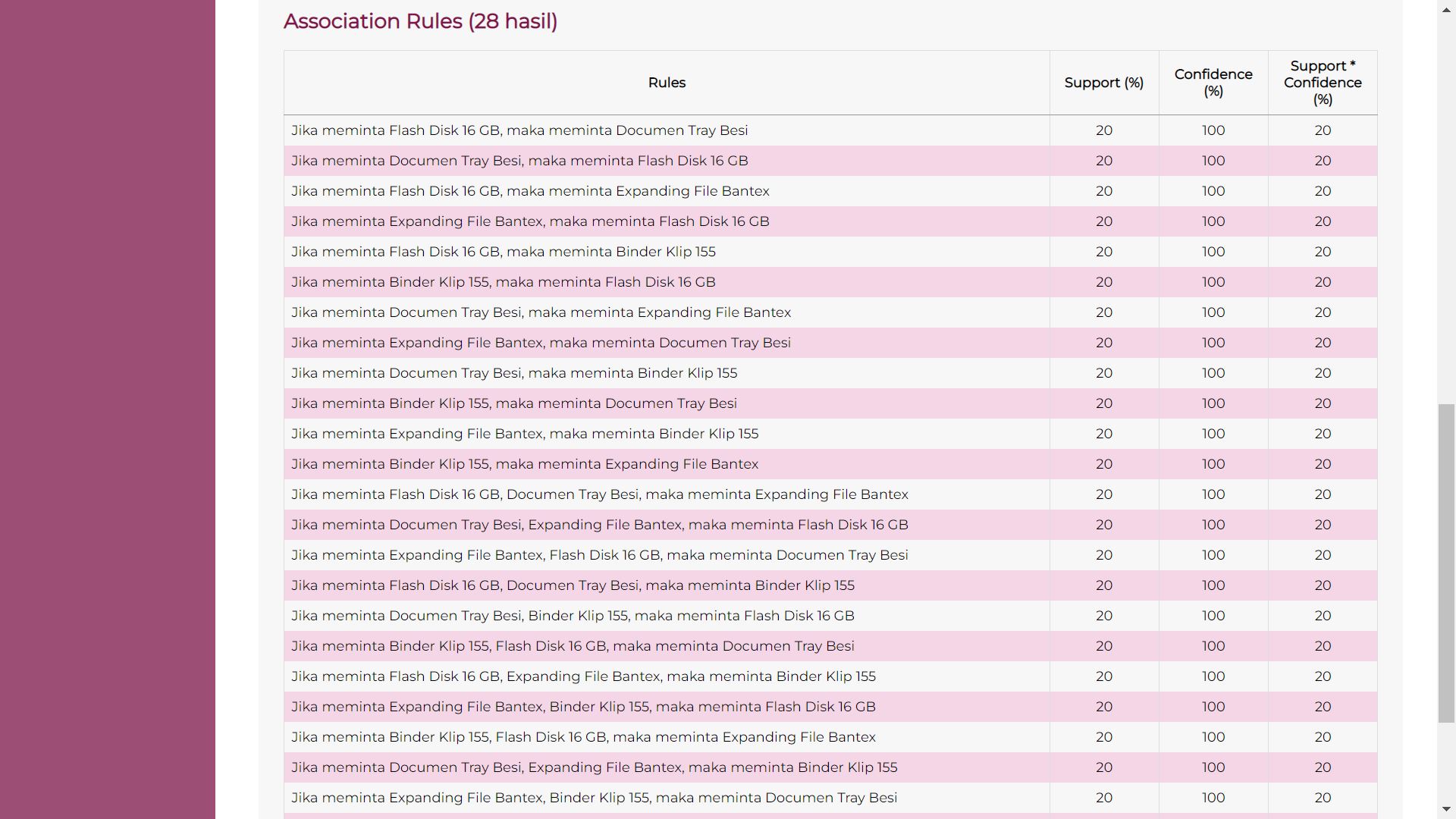
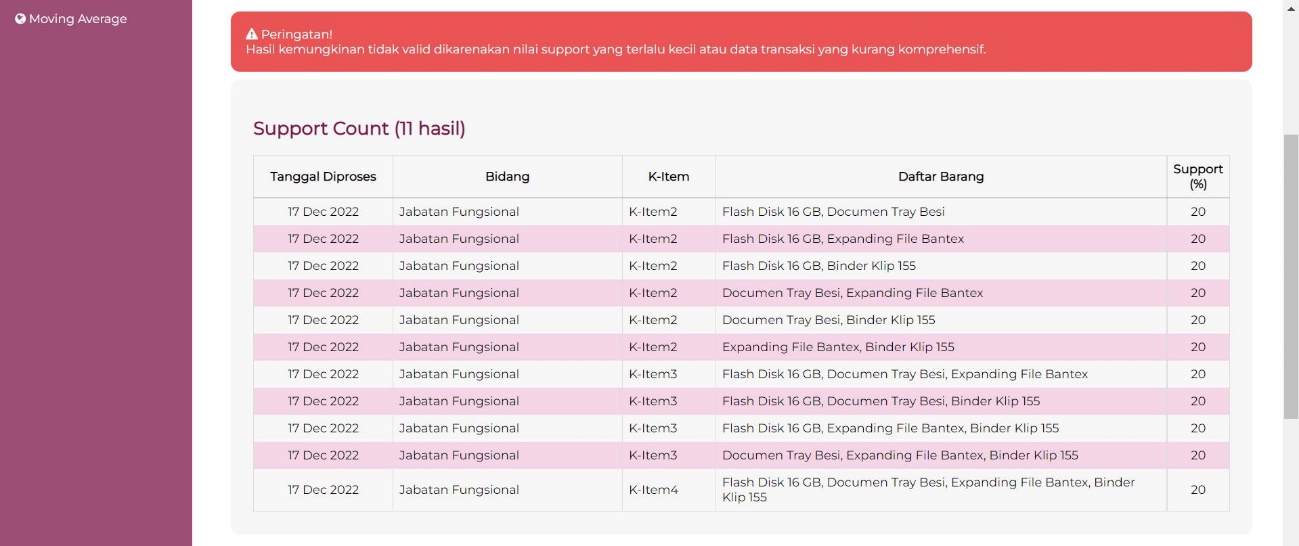
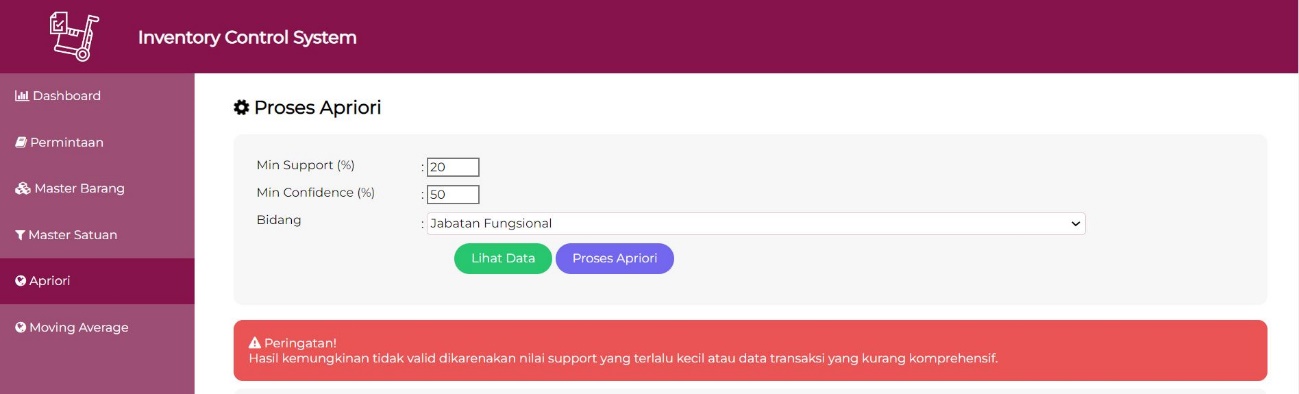
screen :

**a. Lihat data, data tidak ditemukan.**

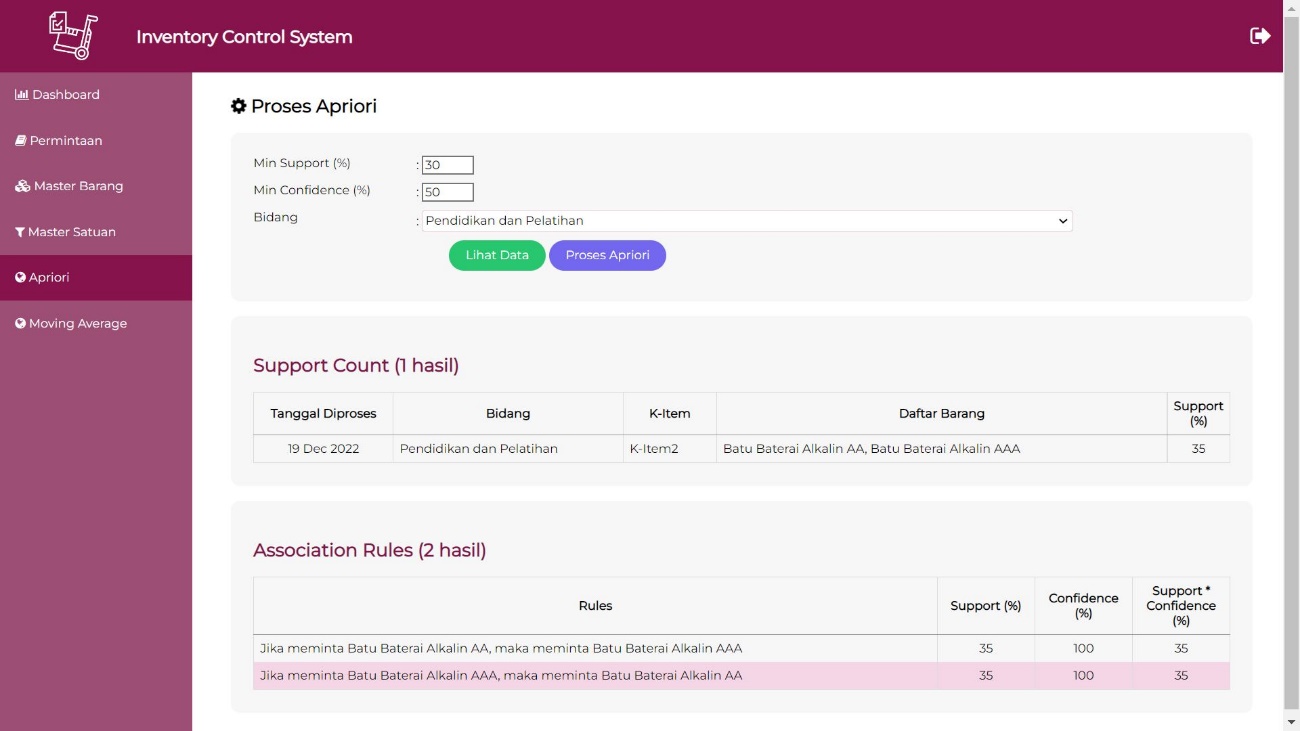


**b. Lihat data, hasil tidak komprehensif karena data permintaan sedikit, variasi barang sangat kurang,**

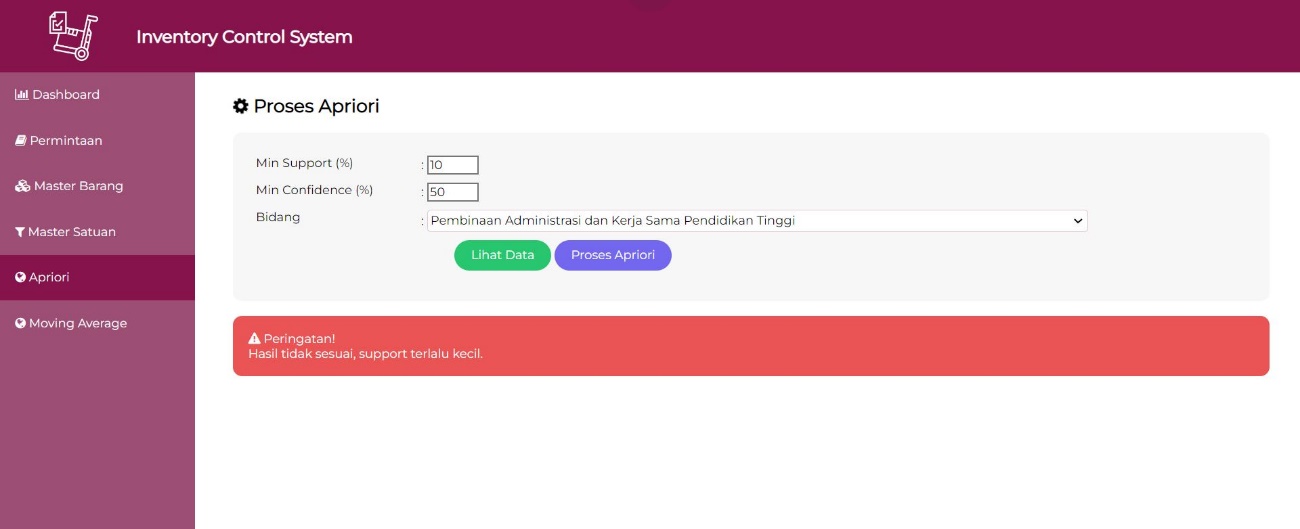
**dan dilakukan proses dengan nilai support yang kecil.**



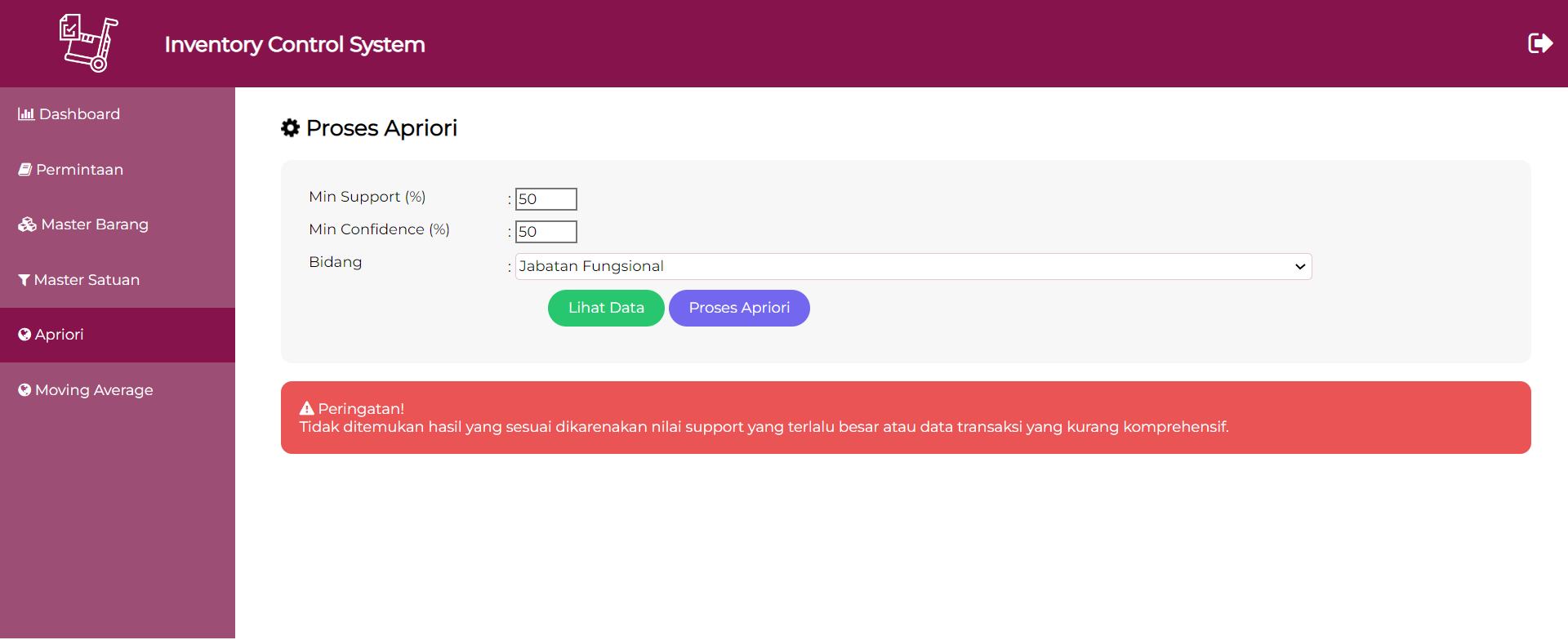
**c. Proses Apriori, normal.**



**d. Proses Apriori, gagal karena support terlalu kecil, data permintaan sedikit tetapi barang yang diminta sangat banyak.**



**e. Proses Apriori, gagal karena support terlalu besar, data permintaan tidak mencukupi.**



**5. Moving Average**

url : ${pageContext.request.contextPath}/movingaverage/load

controller : MovingAverageController.java

method : load()

service : MovingAverageLoadServiceImpl.java

proses : Diagram

Description automatically generated

method : processCalculate()

service : MovingAverageCalculateServiceImpl.java

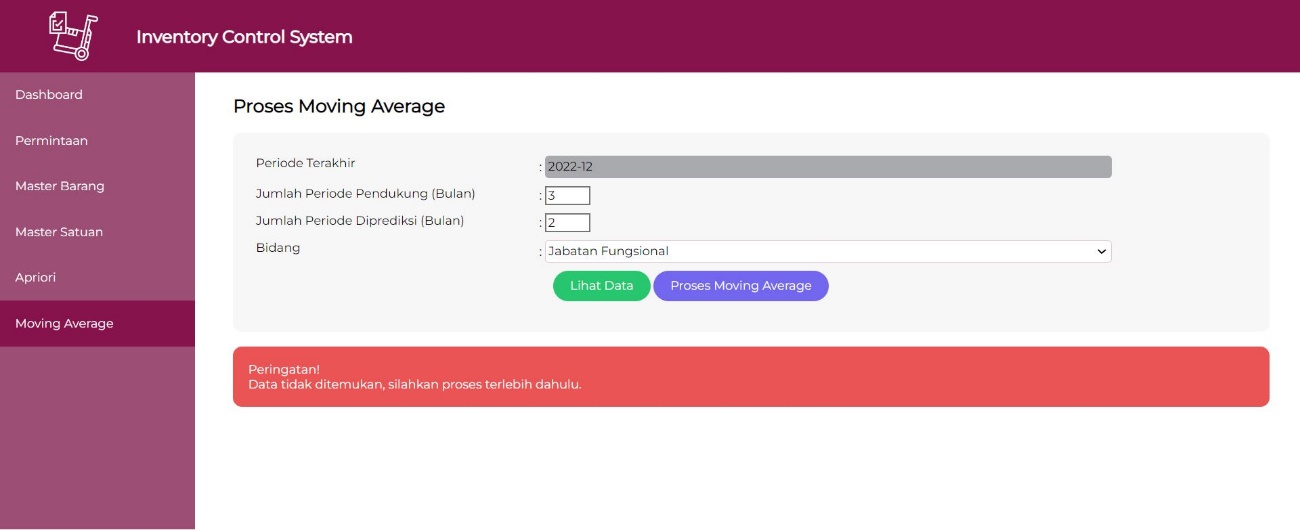
proses :

Diagram

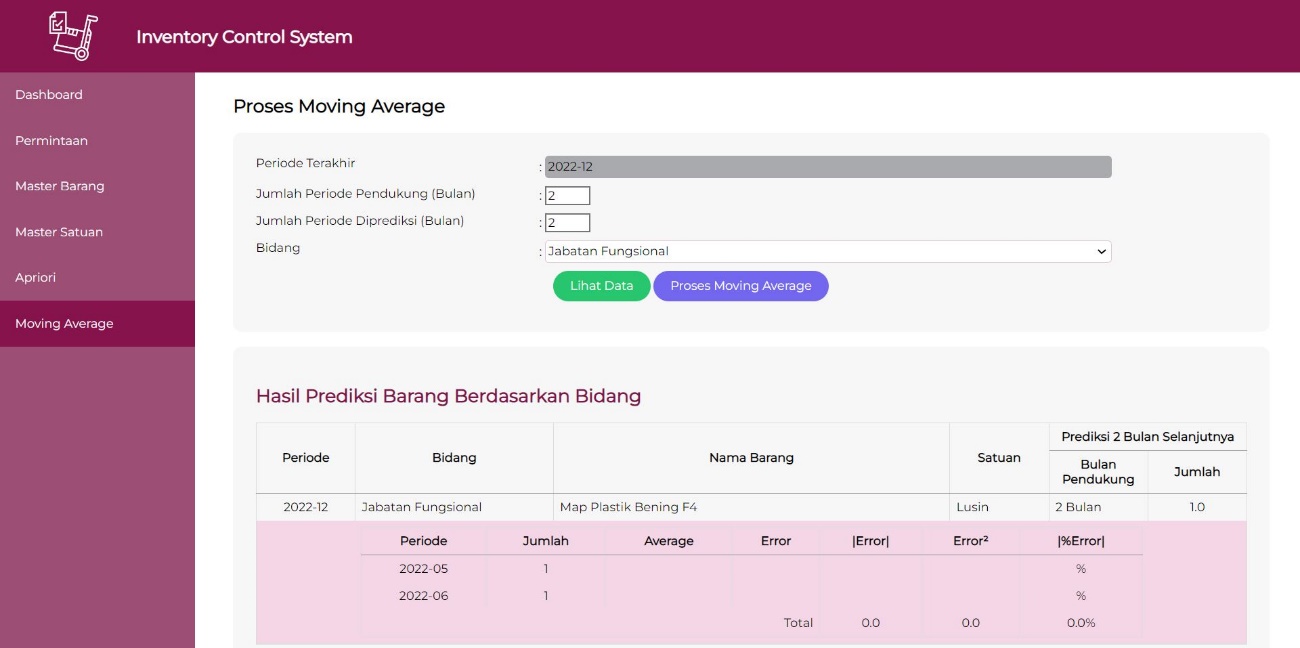
Description automatically generated

screen :

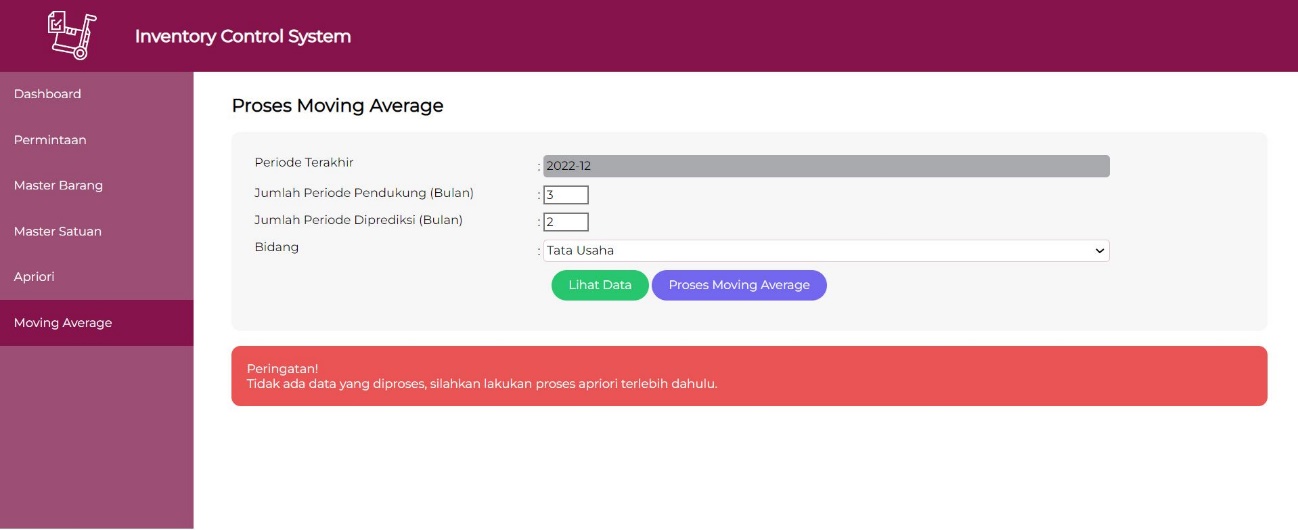
**a. Lihat Data, data tidak ditemukan.**



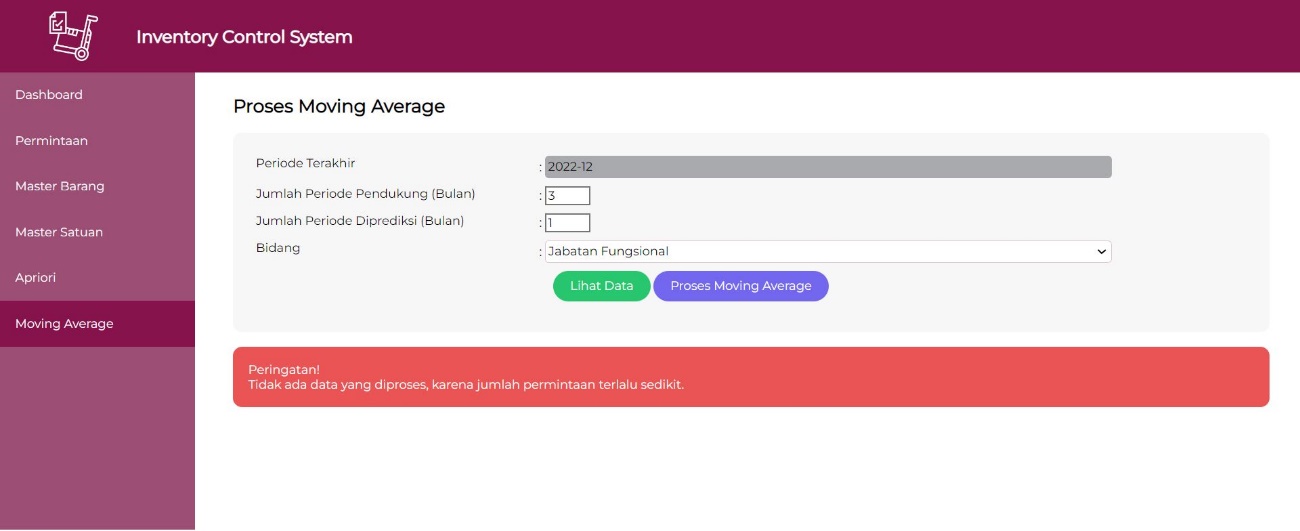
**b. Proses Moving Average, normal.**



**c. Proses Moving Average, gagal karena data barang tidak ditemukan karena proses apriori belum dilakukan terhadap data permintaan.**



**d. Proses Moving Average, gagal karena tidak ada data yang diproses total permintaan per periode lebih kecil dibanding periode pendukung.**



**e. Proses Moving Average, sebagian data tidak diproses karena total permintaan yang sedikit.**

