Masalah pada restrukturisasi

- Masalah restrukturisasi :
  - Kehilangan documentasi
  - Kebutuhan akan komputasi yang tinggi
- Restrukturisasi tidak bisa membantu pada sistem yang memiliki kelemahan modularisainya yaitu komponen-komponen yg terkait terseber di seluruh code.

Modularitas Program

- Proses re-organisasi suatu program sehingga program yang berkaitan terkumpul menjadi satu module.
- Biasanya dilakukan secara manual pada inspeksi program dan re-organisasi

Tipe-tipe Modul

- Abstraksi Data
  - Abstract data type untuk pengelompokkan data structures dan operasinya
- Hardware modules
  - Fungsi yang diperliukan untuk interface dg hardware (driver).
- Functional modules
  - Module terdiri dari fungsi-fungsi yang memiliki tugas yang saling terkait.
- Process support modules
  - Modules yang berfungsi mendukung proses bisnis

Recovering data abstractions

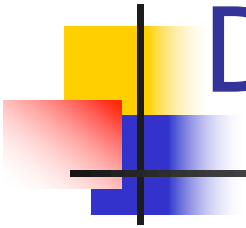
- Many legacy systems use shared tables and global data to save memory space
- Causes problems because changes have a wide impact in the system
- Shared global data may be converted to objects or ADTs
  - Analyse common data areas to identify logical abstractions
  - Create an ADT or object for these abstractions
  - Use a browser to find all data references and replace with reference to the data abstraction

Data re-engineering

- Involves analysing and reorganising the data structures (and sometimes the data values) in a program
- May be part of the process of migrating from a file-based system to a DBMS-based system or changing from one DBMS to another
- Objective is to create a managed data environment

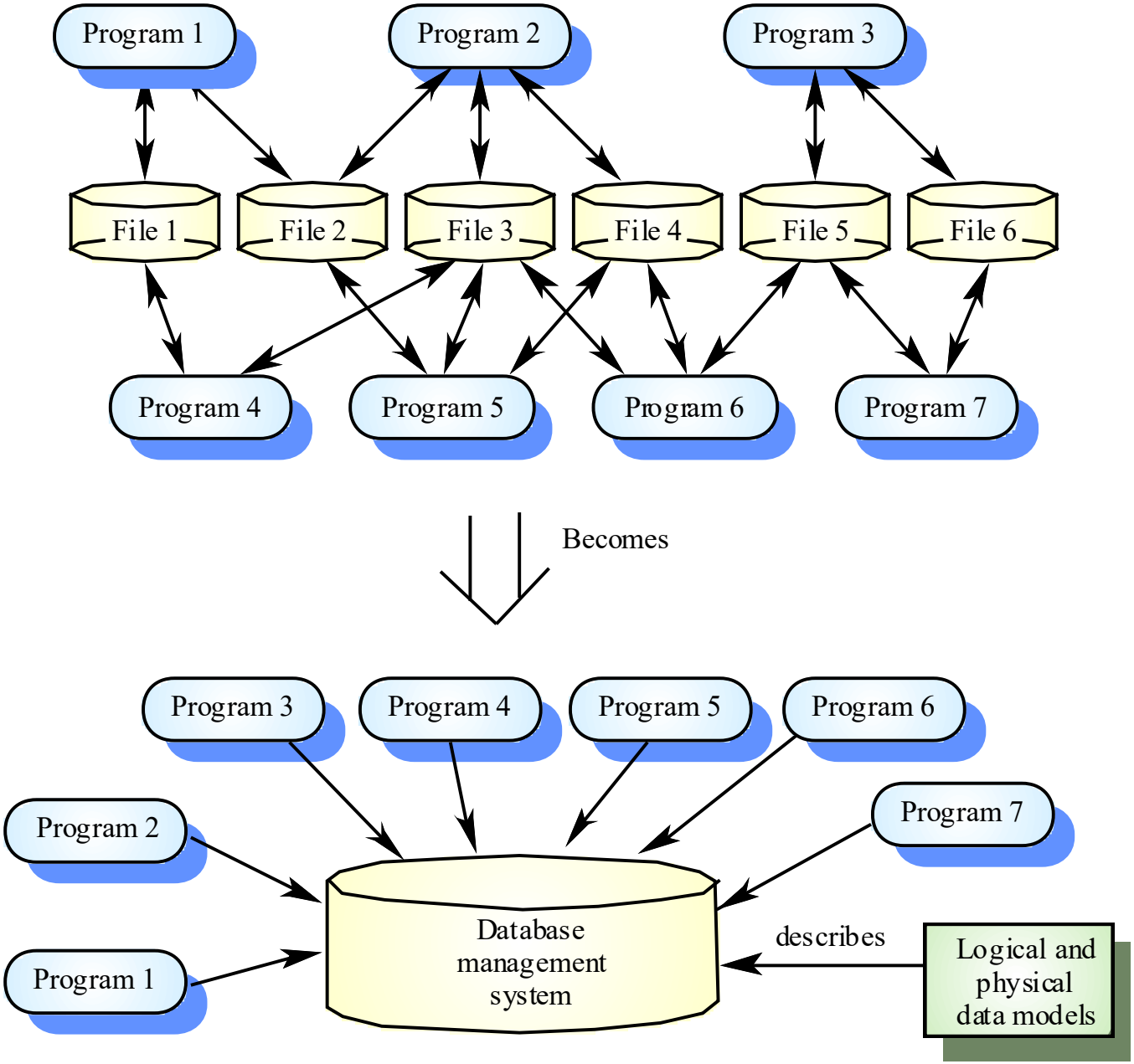
Pendekatan untuk data re-engineering

Approach	Description
Data cleanup	The data records and values are analysed to improve their quality. Duplicates are removed, redundant information is deleted and a consistent format applied to all records. This should not normally require any associated program changes.
Data extension	In this case, the data and associated programs are re-engineered to remove limits on the data processing. This may require changes to programs to increase field lengths, modify upper limits on the tables, etc. The data itself may then have to be rewritten and cleaned up to reflect the program changes.
Data migration	In this case, data is moved into the control of a modern database management system. The data may be stored in separate files or may be managed by an older type of DBMS.



Data problems

- End-users want data on their desktop machines rather than in a file system. They need to be able to download this data from a DBMS
- Systems may have to process much more data than was originally intended by their designers
- Redundant data may be stored in different formats in different places in the system



Data migration

Data problems

- Data naming problems
  - Names may be hard to understand. The same data may have different names in different programs
- Field length problems
  - The same item may be assigned different lengths in different programs
- Record organisation problems
  - Records representing the same entity may be organised differently in different programs
- Hard-coded literals
- No data dictionary

Data value inconsistencies

Data inconsistency	Description
Inconsistent default values	Different programs assign different default values to the same logical data items. This causes problems for programs other than those that created the data. The problem is compounded when missing values are assigned a default value that is valid. The missing data cannot then be discovered.
Inconsistent units	The same information is represented in different units in different programs. For example, in the US or the UK, weight data may be represented in pounds in older programs but in kilograms in more recent systems. A major problem of this type has arisen in Europe with the introduction of a single European currency. Legacy systems have been written to deal with national currency units and data has to be converted to euros.
Inconsistent validation rules	Different programs apply different data validation rules. Data written by one program may be rejected by another. This is a particular problem for archival data which may not have been updated in line with changes to data validation rules.
Inconsistent representation semantics	Programs assume some meaning in the way items are represented. For example, some programs may assume that upper-case text means an address. Programs may use different conventions and may therefore reject data which is semantically valid.
Inconsistent handling of negative values	Some programs reject negative values for entities which must always be positive. Others, however, may accept these as negative values or fail to recognise them as negative and convert them to a positive value.

Data conversion

- Data re-engineering may involve changing the data structure organisation without changing the data values
- Data value conversion is very expensive. Special-purpose programs have to be written to carry out the conversion

The data re-engineering process