



# Pertemuan 1 : Perancangan CDM & PDM

---

Sistem Manajemen basis data

M Najamuddin Dwi Miharja, S.Kom, M.Kom

Teknik Informatika

Fakultas Teknik

Universitas Pelita Bangsa 2020





# Tujuan

- Memberikan gambaran umum rekayasa software
- Mengenalkan Power Designer
- Menyusun Input requirement
- Membuat Data Flow Diagram (DFD)
- Membuat Entity Relational Diagram (ERD)



# Pengertian

## PENGERTIAN CDM

*Conceptual Data Model* atau biasa disebut CDM. CDM memodelkan struktur logis dari keseluruhan aplikasi data, tidak tergantung pada software atau pertimbangan model struktur data. CDM yang valid dapat dikonversi ke PDM.

- CDM dalam penerapannya dapat di samakan dengan ERD yang fungsinya memang sama yaitu memodelkan struktur logik dari basis data.
- CDM dipakai untuk menggambarkan secara detail struktur basis data dalam bentuk logik.
- CDM terdiri dari objek yang tidak diimplementasikan secara langsung kedalam basis data yang sesungguhnya.



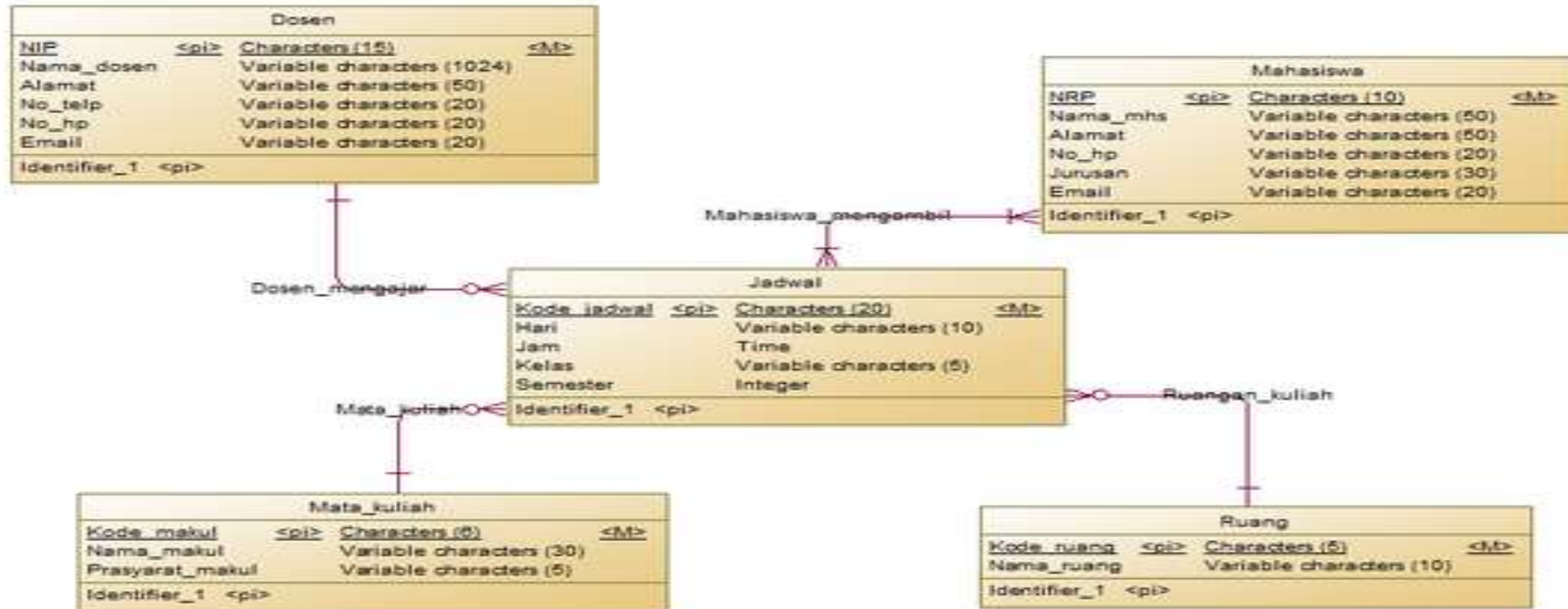
## LANGKAH – LANGKAH UNTUK MEMBUAT CDM

- Pahami terlebih dahulu inti permasalahan dari kasus yang diberikan,
- Tentukan entity apa saja yang terlibat.
- Tentukan atribut-atribut data untuk setiap entity berikut tipe datanya.
- Tentukan hubungan/keterkaitan antar tiap entity berikut kardinalitasnya.
- Modelkan Entity dan Relationship
- Cek kebenaran model
- Perbaiki setiap error dan warning



## CDM

### Conseptual Data Model Aplikasi FRS Online





# PDM

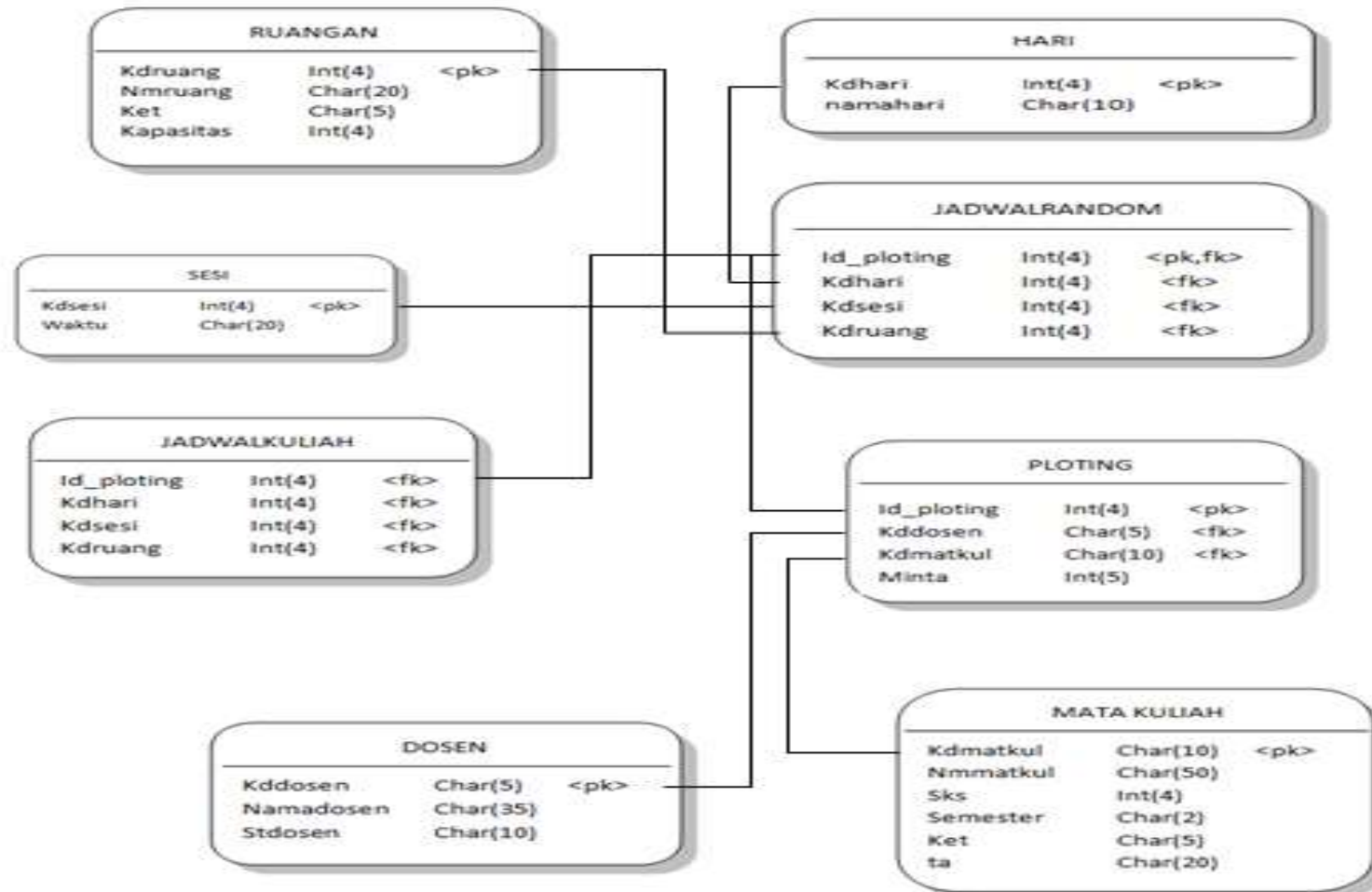
## PENGERTIAN PDM

- Physical Data Model atau yang biasa disebut PDM. PDM merupakan representasi fisik dari database yang akan dibuat dengan mempertimbangkan DBMS yang akan digunakan. PDM dapat dihasilkan (di-generate) dari CDM yang valid.
- PDM dalam penerapannya dapat di samakan dengan Skema Relasi yang fungsinya adalah memodelkan struktur fisik dari suatu basis data.
- Merupakan gambaran secara detail suatu basis data dalam bentuk fisik.
- PDM memperlihatkan struktur penyimpanan data yang benar pada basis data yang digunakan sesungguhnya.



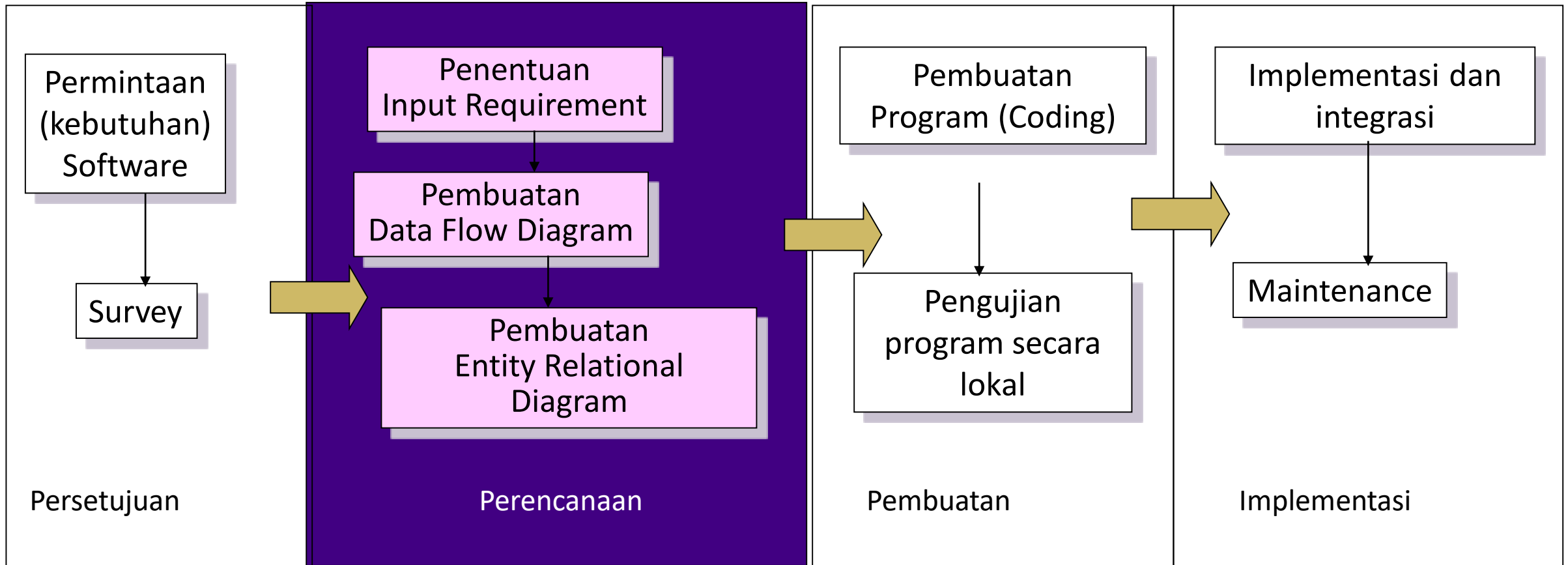


## PDM





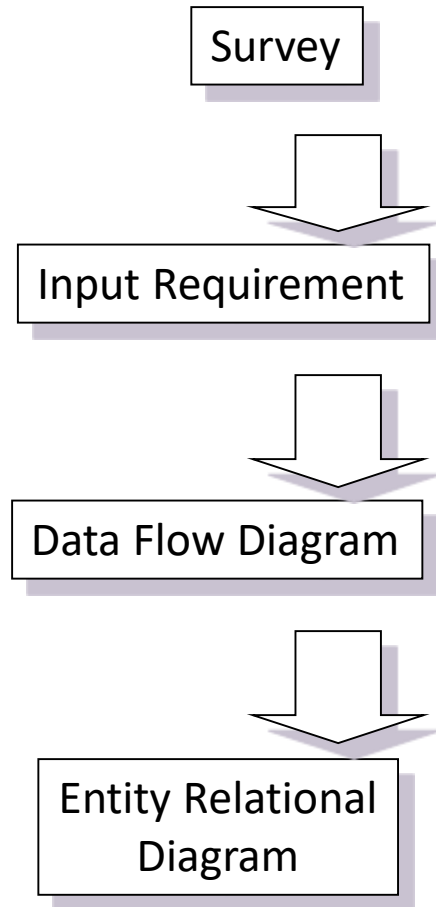
# Gambaran Umum Pengembangan Software







# Gambaran Umum Perencanaan Software



Laporan dan informasi yang diinginkan, data yang tersedia, dan aturan-aturan pokok dari sistem

Menentukan tujuan, pemakai, bidang-bidang yang terkait, input, output dan gambaran proses secara global

Membuat diagram yang menggambarkan aliran data berdasarkan aturan-aturan pokok dan kebijakan dari sebuah sistem. DFD ini menggambarkan proses manajemen yang berlaku.

Membuat diagram yang menggambarkan sistem database yang digunakan.



# Power Designer

- Power Designer adalah program yang digunakan untuk membuat rancangan program aplikasi dengan konsep software engineering.
- Power Designer terdiri dari 4 macam program yaitu : Process Analyst, Data Architect, dan Application Modeler
- Dari program-program yang disediakan dapat dilihat bahwa perancangan software dilakukan dengan pembuatan DFD dan ERD. Setelah itu baru program aplikasi dibuat.



# **Pembuatan Model Perancangan Software Dengan Power Designer**

- Membuat DFD
- Membuat ERD dalam CDM (Conceptual Data Model)
- Mengenerate PDM (Physical Data Model) dari CDM yang sudah dibuat
- Mengenerate Database
- Mengenerate Program Aplikasi



# INPUT REQUIREMENT (1)

- Menjelaskan tujuan.

Tujuan dari sistem informasi nilai adalah memperoleh program aplikasi yang dapat secara cepat dan akurat digunakan untuk keperluan pengisian nilai, pembuatan laporan dan informasi yang dapat diakses secara mudah, serta penentuan perangkat keras dan perangkat lunak yang akan digunakan

- Menentukan pemakai program

Pemakai program ini adalah :

- (1) Administrasi jurusan
- (2) Dosen
- (3) Mahasiswa



# INPUT REQUIREMENT (2)

- Menentukan Output

Output dari sistem informasi nilai adalah laporan-laporan mengenai nilai mahasiswa, baik berupa transkrip nilai, hasil evaluasi kelas, rekapitulasi nilai.

- Menentukan Input

Input dari sistem nilai ini adalah data nilai itu sendiri, dan data-data acuan yang berupa data mahasiswa, data mata kuliah, data jurusan, data dosen dan data beban mengajar.

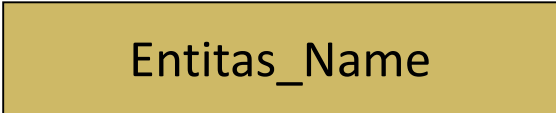
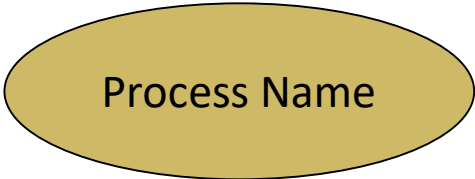
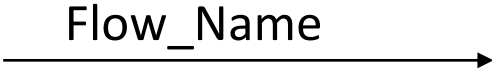



# Data Flow Diagram

- Menggambarkan proses manajemen dari sebuah sistem, yang digambarkan dengan diagram aliran dari data.
- Mempunyai 4 simbol dasar sebagai elemen-elemen sistem yaitu entitas, flow, process, data stored.
- Setiap elemen mempunyai properti yang harus diisi,



# Simbol-Simbol Dasar Dalam Data Flow Diagram

|   |  |
|---|--|
|    | Entitas, menyatakan subyek yang bekerja pada sistem              |
|    | Proses   |
|    | Flow, yang menggambarkan aliran data                             |
|  | Data Store, yang melambangkan tabel dan view yang akan digunakan |





# Pengisian Properti Pada Komponen

- Sorot komponen yang akan diberi property
- Double-click mouse atau tekan Enter pada keyboard maka akan muncul tampilan pengisian properti.
- Setiap komponen mempunyai properti yang berbeda.

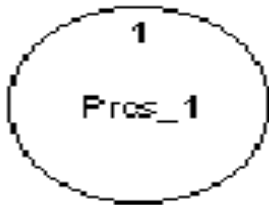


# Contoh DFD Untuk Sistem Informasi Nilai Mahasiswa

- Terdapat 3 pihak yang berkepentingan dalam sistem informasi ini yaitu mahasiswa, dosen dan administrasi jurusan. DFD Level 0 menyatakan tugas dan hak serang user dari sistem informasi ini,
- Setiap proses yang perlu dijelaskan diperlukan level yang lebih tinggi yang merupakan subproses dari proses yang ada.



# Properti Pada Process



**Process Properties**

Definition | **Specification** | Description | Annotation

Name:

Code:

Label:

Number:

Lowest level: ☐

Rules

OK Cancel Apply Help

- Properti yang perlu diisi adalah Name, Code, Label dan Number sebagai identifikasi proses.
- Tanda sama dengan diclick ketika nilai dari kotak isian code disamakan dengan Name dan sebaliknya.
- Lowest level diisi ketika proses tersebut dianggap tidak mempunyai subproses lagi.



# Properti Pada Entitas



**External Entity Properties**

Definition | Description | Annotation

Name:

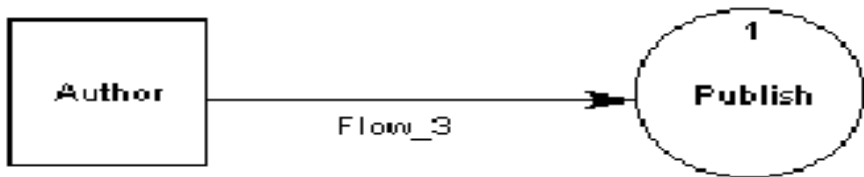
Code:

Label:

- Properti yang perlu diisi adalah Name, Code dan Label sebagai identifikasi dari entitas.
- Deskripsi juga perlu dituliskan untuk menyatakan tugas dari masing-masing entitas.



# Properti Pada Flow



- Properti yang perlu diisi adalah Name, Code, Label sebagai identifikasi Flow.
- Direction digunakan untuk menyatakan arah dari flow.
- Data Item digunakan bila pada flow tersebut dilewatkan data elektronik. Pada flow yang berhubungan dengan data store, data item harus diisi.

The screenshot shows the 'Data Flow Properties' dialog box with the following fields and options:

- Definition** tab is selected.
- Name:** Manuscript proposal
- Code:** MANUSCRIPT\_PROPOSAL
- Label:** (empty)
- Type:** Logical
- Direction:** (indicated by a right-pointing arrow)
- Operations:** C ☐ R ☐ U ☐ D ☐
- Buttons:** Rules, OK, Cancel, Apply, Help



# Properti Untuk Data Store



**Data Store Properties**

Definition | Data Items | Description | Annotation

Name: Manuscripts pending

Code: MANUSCRIPTS\_PENDING

Label:

Number: 1

Is Entity: ☐

Rules

OK Cancel Apply Help

- Properti yang perlu diisi adalah Name, Code, Label dan Number sebagai identifikasi data store.
- Is Entity digunakan bila data store ini dianggap sebagai satu tabel pada ERD.
- Catatan: Setiap Flow yang masuk atau keluar dari Data Store harus mempunyai atribut



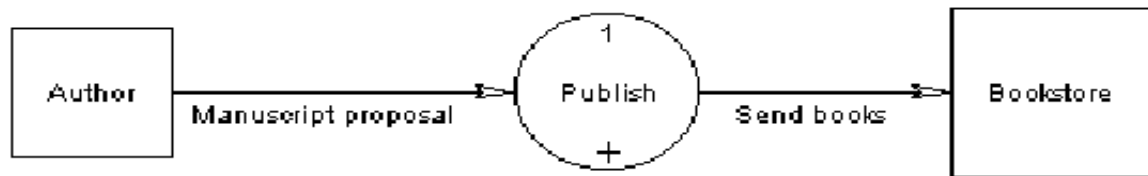
# Cara membuat DFD Dengan Power Designer

- Buka program [**Process Analysis**]
- Lakukan penggambaran DFD dengan drap-drop komponen-komponen pada component toolbox.
- Untuk memberikan property pada setiap komponen dapat dilakukan dengan double-click mouse.
- Untuk proses dekomposisi (menaikkan level karena ada subproses) dapat dilakukan dengan click kanan mouse pada setiap komponen yang akan dikomposisi, lalu pilih [**Decomposition**].
- Setelah DFD digambar maka tekan F4 atau Check Model untuk mengetahui apakah hasil rancangan sudah benar.

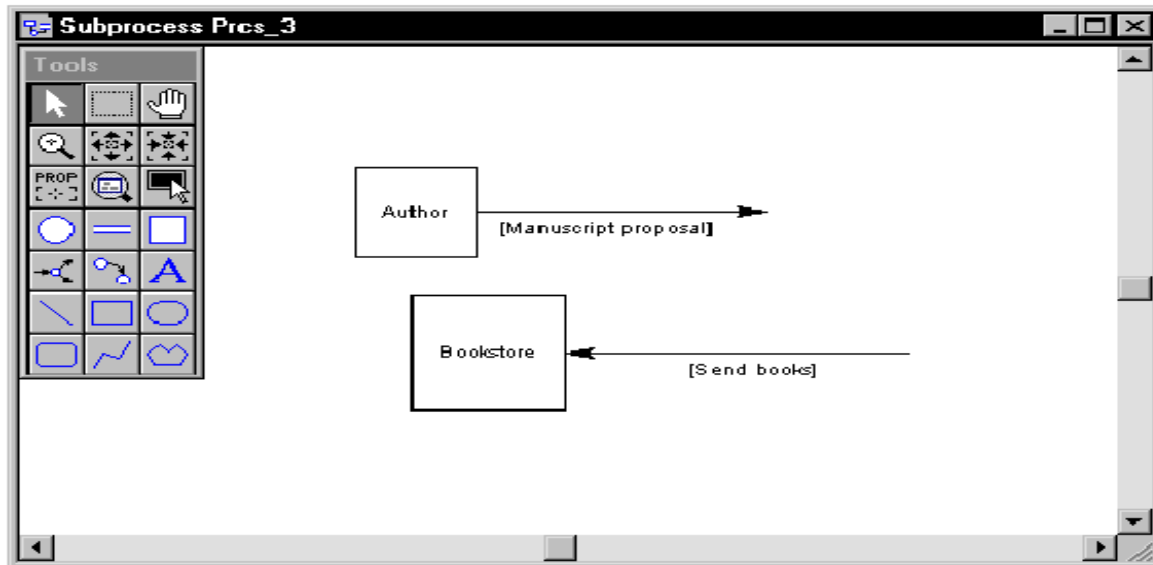




# Pembuatan Process Decomposition



Dekomposisi



- Untuk membuat subprocess dengan decomposition, click kanan pada komponen proses yang dianggap punya subprocess. Lalu pilih dekomposisi.
- Setelah itu baru dilakukan pembuatan DFD pada level yang lebih tinggi.



# Dokumentasi Dari DFD

- Hal penting yang tidak boleh dilupakan adalah pembuatan dokumentasi DFD. Hal ini akan membuat programmer lebih terarah dalam membuat program, sedangkan analyst yang lain dapat membaca sistem dan mungkin bisa memperbaiki.
- Pembuatan dokumentasi pada power designer dapat dilakukan dengan cepat tanpa perlu capek-capek mengetik.
- Untuk membuat dokumentasi ini dapat dilakukan dengan memilih [**Create Report**] pada menu **File**.
- Kemudian pilih salah satu model dari report yang akan dibuat, apakah complete, standard atau list.
- Hasil dari report ini berupa file RTF.

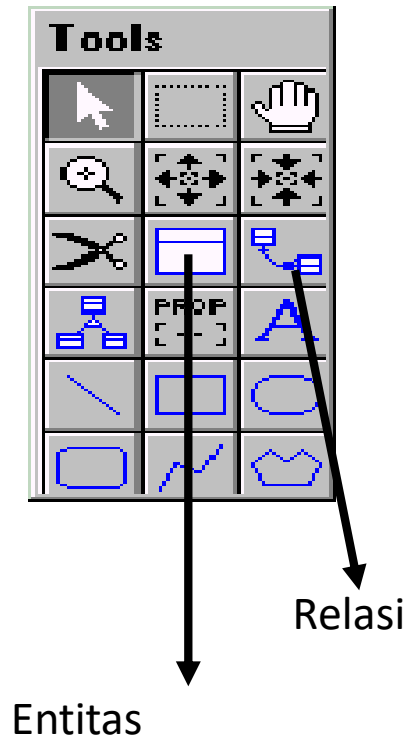


# Entity Relational Diagram

- ERD adalah diagram yang digunakan untuk menggambarkan sistem database yang digunakan.
- ERD terdiri dari tabel ,view, dan relasi.
- Pada Power Designer terdapat 2 macam ERD yaitu : Conceptual Data Model (CDM) dan Physical Data Model (PDM).
- Pembuatan ERD dilakukan dengan menggunakan CDM, sedangkan PDM bisa digenerate secara langsung oleh Power Designer. Tetapi PDM ini adalah model yang menggambarkan bentuk fisik dari data.



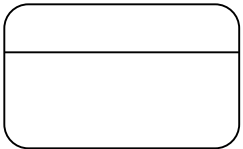
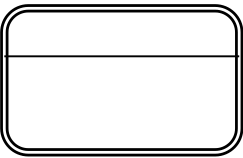

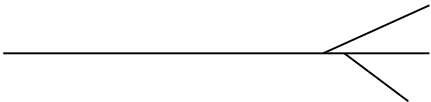
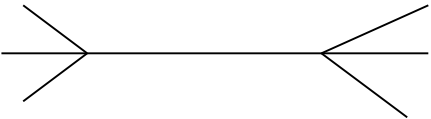
# Membuat ERD



- Untuk membuat ERD buka program [**Data Architect**]
- Pilih New atau Load CDM tergantung apakah mau membuat baru atau memanggil CDM yang pernah dibuat.
- Gambarkan CDM dengan cara drag-drop pada komponen-komponen yang ada.
- Ada 3 macam simbol yang digunakan pada CDM ini yaitu entitas yang berupa tabel dan view, dan relasinya.
- Setelah selesai, untuk menguji kebenarannya dapat dilakukan check model dengan menekan F4



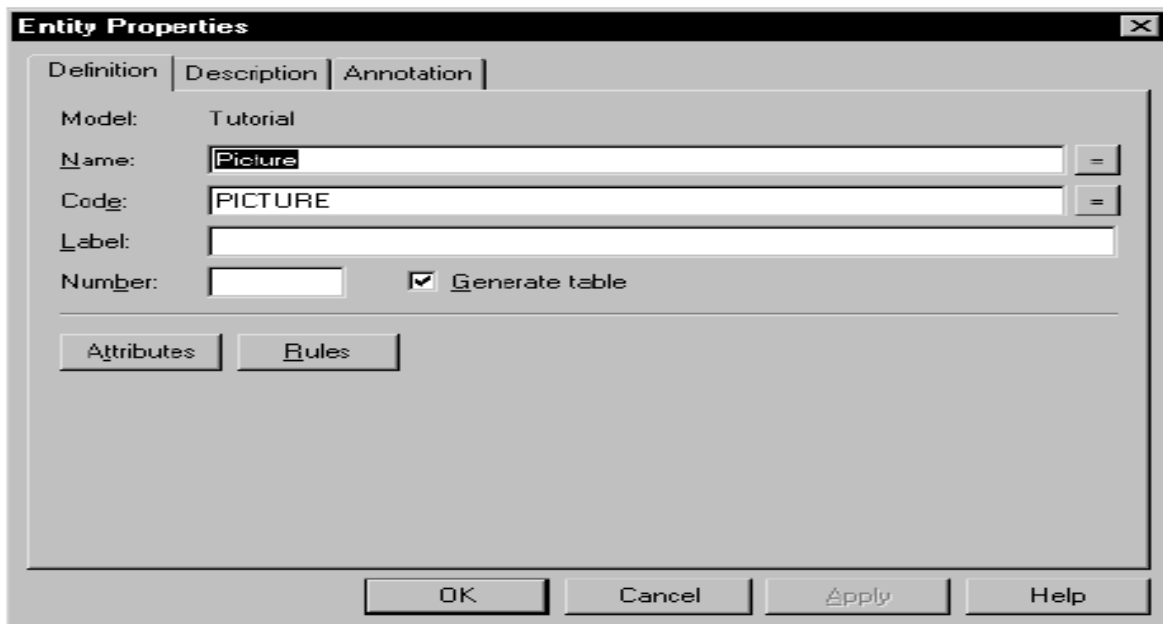
# Simbol-Symbol Dasar CDM

|   |                    |
|---|--------------------|
|    | Tabel              |
|    | View               |
|    | Relasi one to one  |
|   | Relasi one to many |
|  | Relasi many to one |



# Properti Pada Entitas

|        |
|--------|
| Ent_84 |
|        |



The image shows a screenshot of the 'Entity Properties' dialog box in a database design software. The dialog has three tabs: 'Definition', 'Description', and 'Annotation'. The 'Definition' tab is active. It contains the following fields and controls:

- Model:** Tutorial
- Name:** Picture (with a '=' button)
- Code:** PICTURE (with a '=' button)
- Label:** (empty text box)
- Number:** (empty text box)
- ☒ **Generate table**
- Buttons:** Attributes, Rules
- Footer Buttons:** OK, Cancel, Apply, Help

- Properti yang perlu diisi adalah Name, Code, Label dan Number sebagai identifikasi Entitas
- Hal lain yang harus diisi adalah atribut, yaitu field-field yang ada pada entitas tersebut.



# Pengisian Atribut Pada Entitas

Nama field

Nama variabel untuk field

Tipe data dari field

|   | Name | Code | Data Type | I                        | M                        | D                                   | N                        |
|---|------|------|-----------|--------------------------|--------------------------|-------------------------------------|--------------------------|
| → |      | =    |           | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
|   |      |      |           |                          |                          |                                     |                          |
|   |      |      |           |                          |                          |                                     |                          |
|   |      |      |           |                          |                          |                                     |                          |
|   |      |      |           |                          |                          |                                     |                          |
|   |      |      |           |                          |                          |                                     |                          |
|   |      |      |           |                          |                          |                                     |                          |
|   |      |      |           |                          |                          |                                     |                          |
|   |      |      |           |                          |                          |                                     |                          |
|   |      |      |           |                          |                          |                                     |                          |

Insert Delete Add...

Label:

Domain: (None) ...

☐ Identifier ☐ Mandatory ☒ Display

Check Describe Annotate OK Cancel Help

Primary Key





# Properti Pada Relasi

- Identifikasi Relasi diisi dari Name, Code dan Label.
- Jenis relasi dipilih dari Cardinality
- Apakah data yang direlasikan boleh tidak punya teman atau tidak dilakukan pada mandatory



# Dokumentasi Dari CDM/PDM

- Hal penting yang tidak boleh dilupakan adalah pembuatan dokumentasi CDM/PDM. Hal ini akan membuat programmer lebih terarah dalam membuat program, sedangkan analyst yang lain dapat membaca sistem dan mungkin bisa memperbaiki.
- Pembuatan dokumentasi pada power designer dapat dilakukan dengan cepat tanpa perlu capek-capek mengetik.
- Untuk membuat dokumentasi ini dapat dilakukan dengan memilih [**Create Report**] pada menu **File**.
- Kemudian pilih salah satu model dari report yang akan dibuat, apakah complete, standard atau list.
- Hasil dari report ini berupa file RTF.



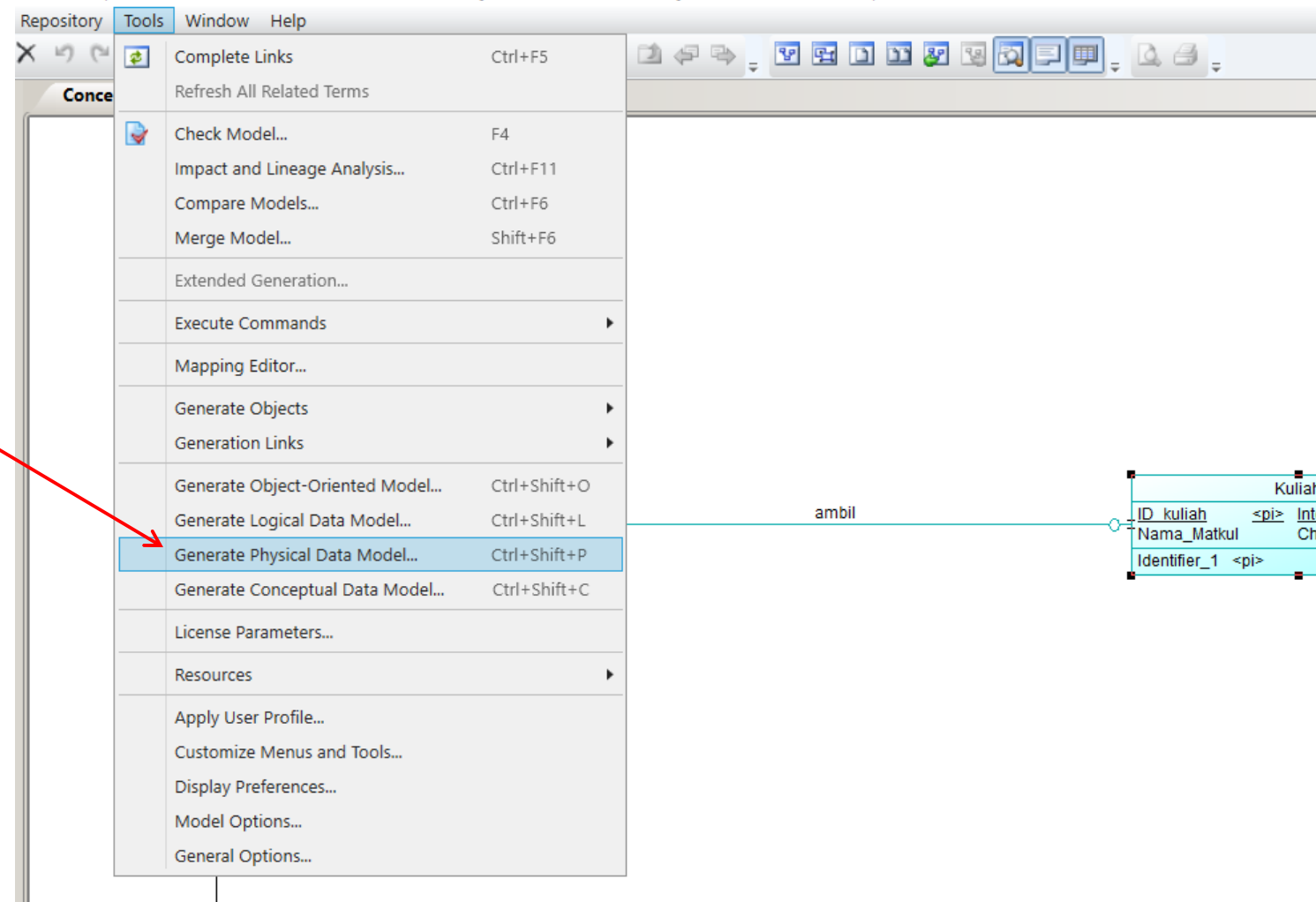
# Membuat PDM

- Untuk membuat PDM dapat dilakukan dengan membuat secara langsung dengan memilih New PDM, atau generate CDM yang sudah dibuat. Cara kedua adalah cara yang disarankan.
- Sebelum generate CDM menjadi PDM pastikan dahulu bahwa CDM tersebut benar, kemudian pilih [**Generate PDM**] pada menu yang ada di layout atas.
- Anda akan ditanyakan DBMS yang akan digunakan dalam program aplikasi yang akan dibangun apa. Dalam contoh ini digunakan DBMS Ms. Access dan program aplikasinya Visual Basic.



# Membuat PDM

el\_1, ConceptualDataModel\_1 - D:\lecture\silabus\2. Ganjil 20-21\6. sistem manajemen basis data\ConceptualDataModel\_1.cdm]





# Membangkitkan Database

- Bila PDM sudah dibuat, membuat database dengan power designer adalah sesuatu yang sangat mudah karena database dibangkitkan berdasarkan PDM.
- Power Designer 6.0 mengenal 48 macam DBMS.
- Hasil pembuatan database ini berupa tabel-tabel dan relasinya yang secara langsung pada DBMS, dengan dilengkapi script sebagai dokumentasi. Bagi programmer yang menyukai script hal ini lebih menyenangkan.

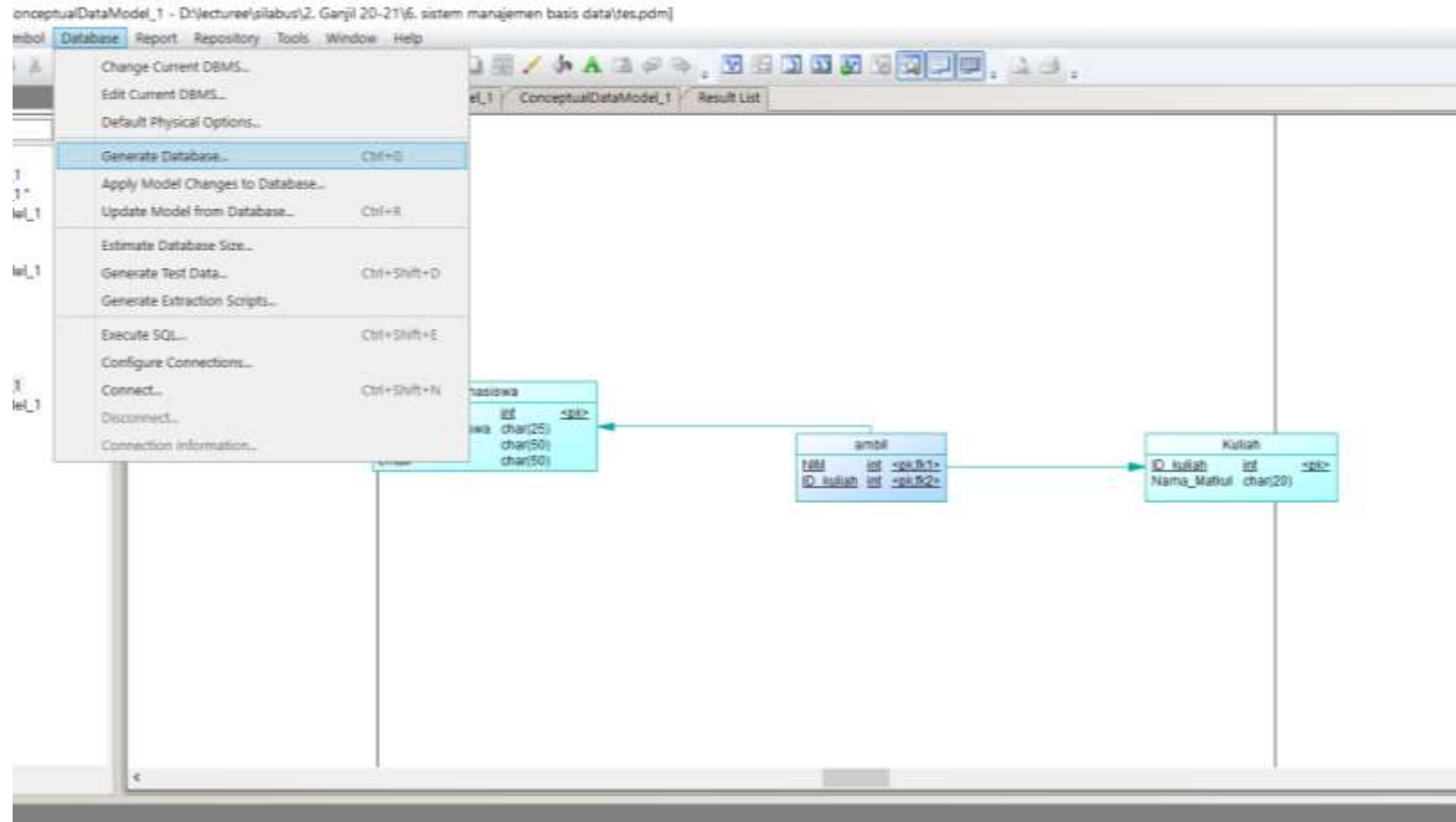


# Membangkitkan Database Dengan Power Designer

- Buka PDM yang sudah dibuat.
- Untuk memilih DBMS yang digunakan pilih Database → [Change Target Database], pada pembuatan PDM hal ini sebenarnya sudah dilakukan tetapi mungkin ingin menggunakan database yang lain, ini bisa dilakukan.
- Untuk melakukan koneksi pada DBMS, pilih Database → Connect.
- Untuk mengenerate database dilakukan dengan memilih Database → Generate Database
- Setelah database digenerate, cek langsung pada DBMSnya.



# Generate Database







# Terimakasih

---

M Najamuddin Dwi Miharja, S.Kom, M.Kom

085730304025 – [gmail@najamuddindwi.com](mailto:gmail@najamuddindwi.com)