

MODUL PRAKTIKUM 1 PENGENALAN JOB PADA PENTAHO

PRAKTIKUM DATA WAREHOUSE DAN BUSINESS INTELLIGENT

1. *Dasar Teori*

Pentaho Data Integration (PDI) atau Kettle adalah software dari Pentaho yang dapat digunakan untuk proses ETL (Extraction, Transformation dan Loading). PDI dapat digunakan untuk migrasi data, membersihkan data, loading dari file ke database atau sebaliknya dalam volume besar. PDI menyediakan graphical user interface dan drag-drop komponen yang memudahkan user.

Elemen utama dari PDI adalah Transformation dan Job. Transformation adalah sekumpulan instruksi untuk merubah input menjadi output yang diinginkan (input-proses-output). Sedangkan Job adalah kumpulan instruksi untuk menjalankan transformasi. Ada tiga komponen dalam PDI: Spoon, Pan dan Kitchen. Spoon adalah user interface untuk membuat Job dan Transformation. Pan adalah tools yang berfungsi membaca, merubah dan menulis data. Sedangkan Kitchen adalah program yang mengeksekusi job.

2. *Instalasi*

STEP-1: DOWNLOAD PENTAHO DATA INTEGRATION (PDI/KETTLE) SOFTWARE

Langkah pertama adalah mengunduh PDI Community Edition dari <https://sourceforge.net/projects/pentaho/>. Versi terbaru adalah 9.1. File-file tersebut akan diunduh sebagai file zip dengan nama "pdi-ce-9.1.0.0-324.zip".

STEP-2: EKSTRAKSI FILE ZIP

Klik kanan file tersebut dan pilih 'Extract Here' jika Anda ingin mengekstraknya di folder unduhan. Jika Anda ingin memilih folder lain, klik kanan dan pilih opsi 'Ekstrak File ...' dan berikan jalur folder tujuan. Nama default dari folder yang diekstrak adalah "data-integration".

Langkah ketiga bisa dilewati jika aplikasi Spoon.bat sudah bisa langsung di buka.

STEP-3: CEK VERSI JAVA (JRE DAN JDK) SERTA MENGATUR ENVIRONMENT VARIABLES

Untuk langkah selanjutnya jika belum menginstall JRE dan JDK bisa dilihat pada link dibawah ini untuk mengatur Environment Variables dan mendownload versi JRE dan JDK terbaru:

<https://www.hitachivantara.com/en-us/pdf/white-paper/pentaho-community-edition-installation-guide-for-windows-whitepaper.pdf>

Versi JRE terbaru : <https://java.com/en/download/>

Versi JDK terbaru : <https://www.oracle.com/java/technologies/javase-jdk15-downloads.html>

Alternatif lain buat download JDK dan JRE : <https://www.techspot.com/downloads/5198-java-jre.html>

STP-4 JALANKAN SPOON.BAT

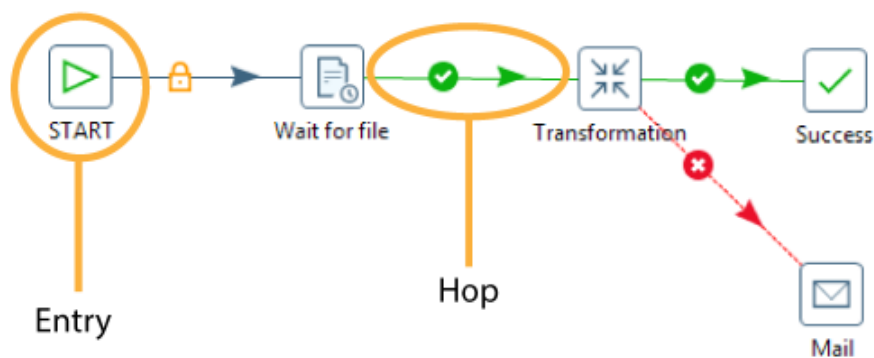
Jalankan aplikasi spoon.bat pada folder "data-integration" yang telah di ekstraksi tadi.

3. Pengenalan Job pada Pentaho

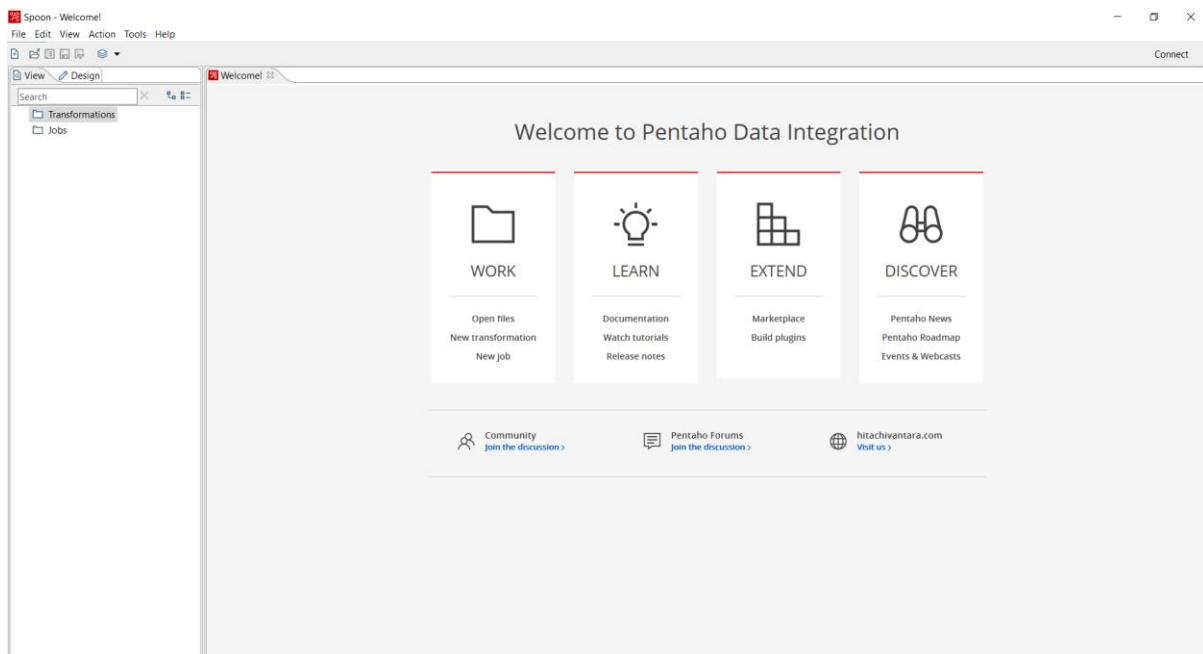
Jobs digunakan untuk mengoordinasikan aktivitas ETL seperti:

- Menentukan aliran dan dependensi untuk urutan transformasi apa yang harus dijalankan
- Mempersiapkan eksekusi dengan memeriksa ketentuan seperti, "Apakah file sumber saya tersedia?" atau "Apakah tabel ada?"
- Melakukan operasi database
- Manajemen File seperti memposting atau mengambil file menggunakan FTP, menyalin file dan menghapus file
- Mengirim pemberitahuan berhasil atau gagal melalui email

Jobs terdiri dari job hops, entries, dan job settings.



Gambar Struktur dari Job



Gambar Tampilan Awal Aplikasi Spoon pada PDI

MEMBUAT JOB BARU

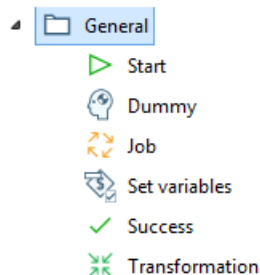
Langkah membuat Job baru:

- Pada halaman awal, di kotak “WORK”, klik “New job”, atau
- pada menu “File → New → Job”, atau
- dengan shortcut “CTRL-ALT-N”

MENAMPILKAN PESAN PADA JOB

1. Langkah Memulai Job:

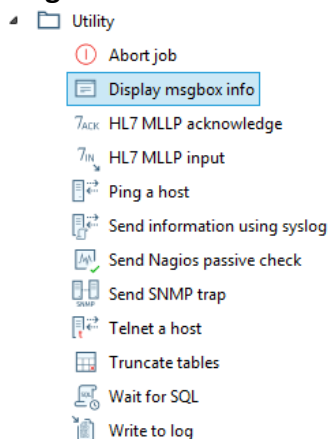
- a. Klik tab **Design**. Node yang berisi entri Job muncul.
- b. Perluas node **General**; dan pilih entri **Start**.
- c. Seret entri **Start** ke ruang kerja (kanvas) di sebelah kanan.
- d. Entri **Start** pekerjaan menentukan di mana eksekusi akan dimulai.



Gambar Tampilan Node General

2. Untuk menampilkan messagebox

- a. Pilih pada Node Utility → **display msgbox info**
- b. Seret entri **display msgbox info** ke kanvas



Gambar Tampilan Node Utility

3. Buat hop antara entri **Start** dan **Display msgbox info** dengan cara:

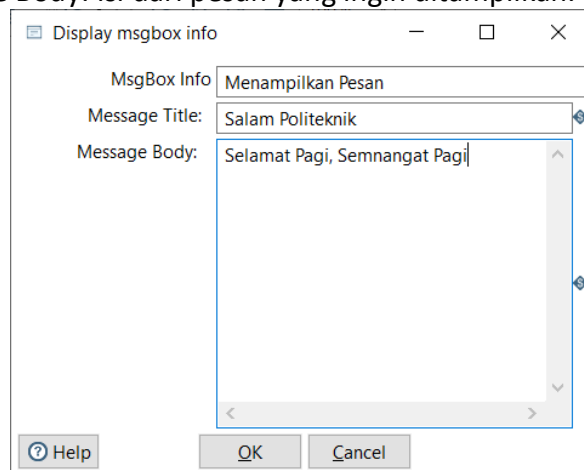
- a. Tekan **SHIFT** key.
- b. Klik entri **START** dan tarik mouse untuk menampilkan garis ke entri **display msgbox info**.
- c. Lepaskan **SHIFT** key.

4. Tampilan setelah dihubungkan.



Gambar Tampilan dua entri yang dihubungkan


5. Double-klik entri **Display msgbox info** untuk menampilkan dialog box.
6. Pada dialog box:
 - a. Pada field MsgBox Info: kita bisa mengganti nama tampilan entri pada kanvas.
 - b. Message Title: Judul dari Pesan.
 - c. Message Body: Isi dari pesan yang ingin ditampilkan.



Gambar Dialog box "Display msgbox info"

TUGAS:

Pada Message Body isi dengan NAMA_NIM_KELAS.

7. Menampilkan hasil dari Job
 - a. Pada kanvas bagian atas, klik ikon **Run** .
 - b. Jika muncul dialog box, klik **Run**.

8. Tampilan akhir

HASIL TAMPILAN DIISI OLEH MAHASISWA SESUAI PERINTAH TUGAS

MENAMPILKAN PESAN DENGAN MEMAKAI VARIABLE

1. Jika memakai variabel kita harus menambahkan entri **SET VARIABLES** pada node **General** diantara entri **START** dan entri **display message box**, lalu buat hop untuk menghubungkannya. "Jangan lupa menghapus hop antara **START** dan **display message box** dengan cara klik kanan garis penghubung dan pilih **delete**".



Gambar Tampilan tiga entri yang dihubungkan

2. Double-Klik entri set **VARIABLES**,

- a. Field **Job entry name**: untuk mengubah tampilan nama entri/entri
- b. Pada **Properties file** dan **Settings**, biarkan default.
- c. Pada **Variables**:

Variable name untuk nama variabel yang ditentukan, dan **value** adalah nilai dari variabel tersebut serta **Variable scope type**: untuk menentukan ruang lingkup variabel, opsi yang memungkinkan adalah:

- Valid in the Java virtual machine: seluruh mesin virtual akan mengetahui tentang variabel ini.
- Valid in the current job: variabel hanya valid atau dapat diakses pada job saat ini aktif.
- Valid in the parent job: variabel hanya valid di parent job.
- Valid in the root job: variabel ini valid di root job dan semua job turunan serta transformasi.

Set variables

Job entry name: Set variables

Properties file

Name of properties

Variable scope: Valid in the Java Virtual Machine

Settings

Variable substitution? ☒

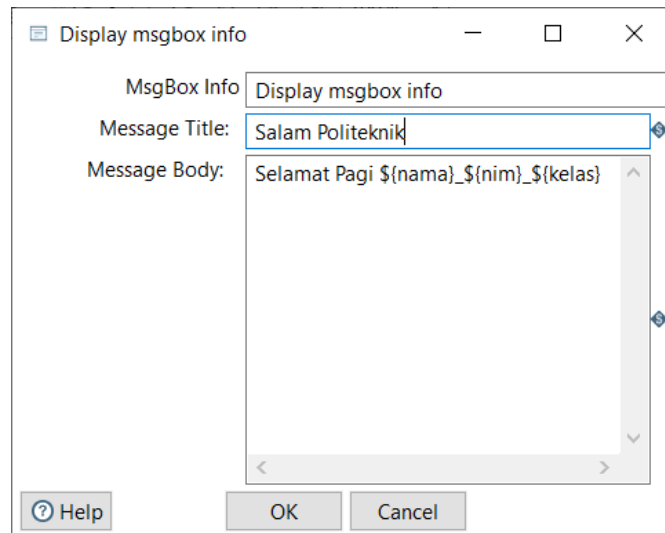
Variables :

#	Variable name	Value	Variable scope type
1	nama	Renovita Edelani	Valid in the current job
2	nim	2110131048	Valid in the current job
3	kelas	4 D4 IT B	Valid in the current job

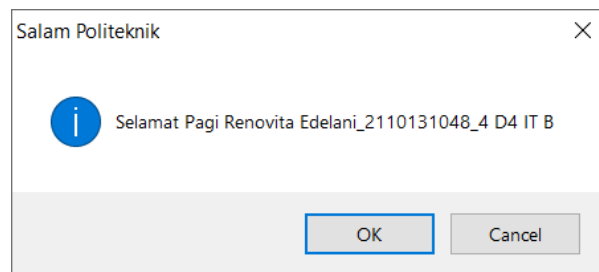
Help OK Cancel

Gambar Dialog box set variables dengan menentukan inialisasi variabel yang akan digunakan

3. Double-Klik entri **display msgbox info**, pada message body gunakan variabel yang sudah kita inialisasi dengan format **\${NAMA_VARIABEL}**



Gambar Dialog box Display msgbox info




Gambar Contoh hasil tampilan akhir menggunakan variabel

TUGAS:

Pada Message Body isi dengan variabel NAMA_NIM_KELAS.

4. Menampilkan hasil dari Job

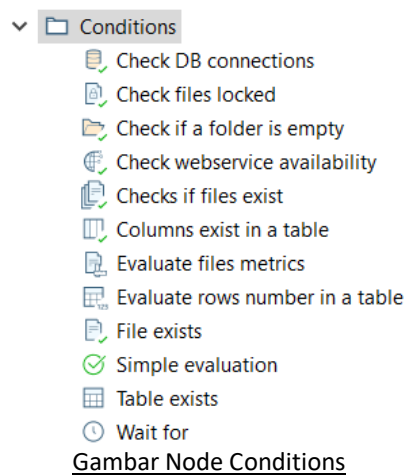
- a. Pada kanvas bagian atas, klik ikon **Run** .
- b. Jika muncul dialog box, klik **Run**.

5. Hasil setelah di run.

HASIL TAMPILAN DIISI OLEH MAHASISWA SESUAI PERINTAH TUGAS

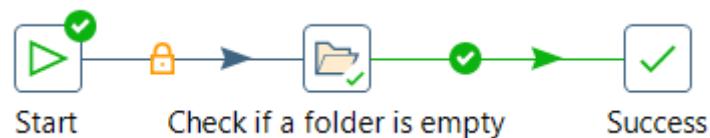
MENGECEK FOLDER KOSONG

1. Untuk mengecek folder kosong kita memakai entri **check if a folder is empty** pada Node **Conditions**.



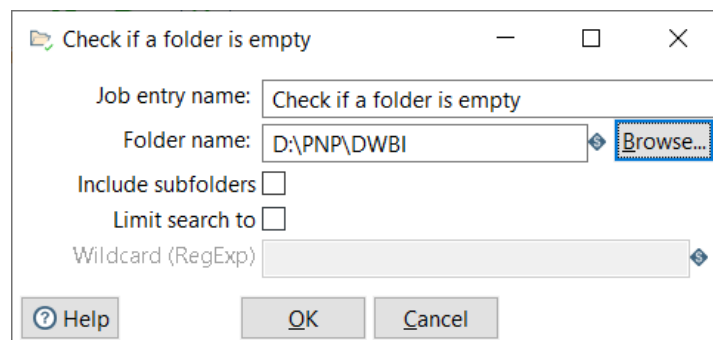
Gambar Node Conditions

2. Langkahnya sama dengan percobaan sebelumnya, kita membutuhkan entri **START** untuk memulai job dan diakhiri dengan entri **SUCCESS** yang terletak pada Node **General** untuk mengetahui proses telah berhasil atau gagal. Buat hop yang menghubungkan antara entri **START**, entri **check folder is empty** dan entri **SUCCESS**.



Gambar Job untuk mengecek folder kosong atau tidak

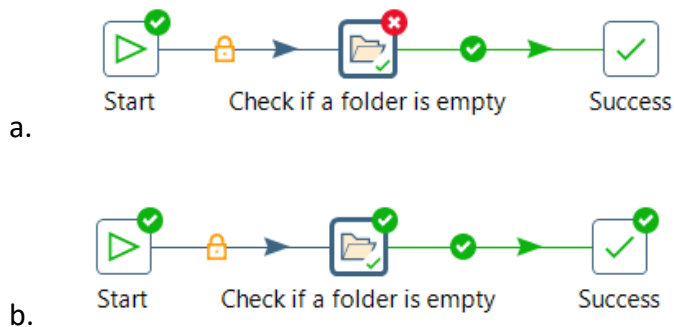
3. Selanjutnya kita akan mencari folder mana yang akan dicek. Double-Klik entri **check if a folder is empty** lalu pilih lokasi folder dengan meng-klik tombol **Browse**.
 - a. Field Job entry name: untuk mengubah tampilan nama entri/entri
 - b. Field Folder Name: lokasi folder yang dicek apakah kosong atau berisi
 - c. Checkbox Include Sub Folder: subfolder pada folder pencarian juga dicek.
 - d. Checkbox limit search to: Membatasi pencarian file hanya untuk file dengan karakter pengganti tertentu (ekspresi reguler). misalnya: * \. txt \$



Gambar dialog box dari entri Check if a folder is empty




4. Jalankan job.

TUGAS: JELASKAN DAN ANALISA PERBEDAAN KEDUA GAMBAR a dan GAMBAR b DIBAWAH INI

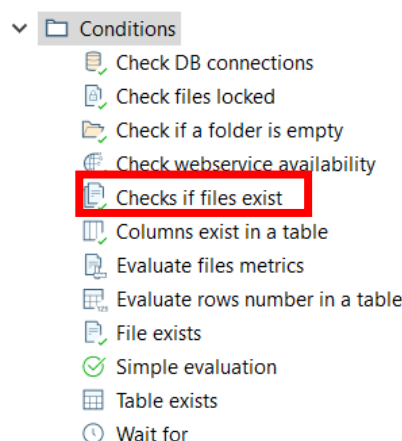


MENGECEK KEBERADAAN FILE/FOLDER

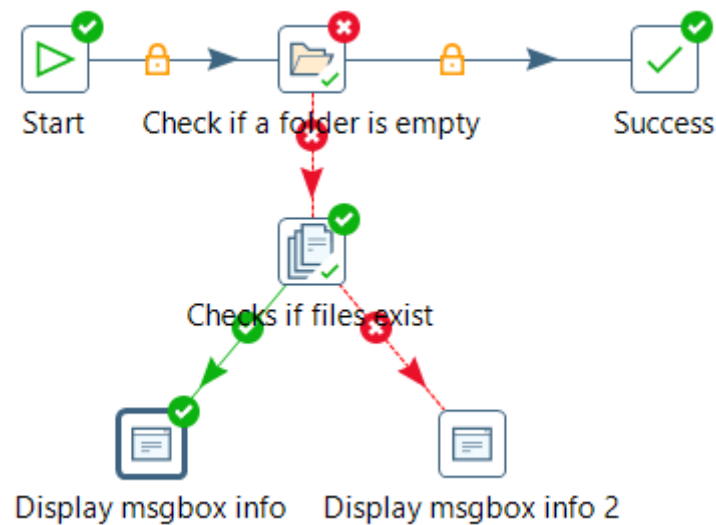
Sebelum ke percobaan selanjutnya, kita harus mengetahui perbedaan dari hop yang menghubungkan antar entry/entri. Hop link ke entri job, berdasarkan hasil entri job sebelumnya, dan menentukan langkah selanjutnya.

-  : Unconditional → entri Job berikutnya akan dieksekusi tanpa memperhatikan dari hasil entri Job sebelumnya.
-  : result is true → jika hasil eksekusi entri sebelumnya bernilai benar atau true maka dilanjutkan dengan entri terhubung dengan hop ini.
-  : result is false → jika hasil eksekusi entri sebelumnya bernilai salah atau false maka dilanjutkan dengan entri yang terhubung dengan hop ini.

1. Untuk mengecek apakah file ada atau tidaknya, kita menggunakan entri **check if files exist** pada Node **Conditions**.



2. Buat lah struktur Job seperti gambar dibawah ini.



- Pada entri **Check if a folder is empty**

Check if a folder is empty

Job entry name: Check if a folder is empty

Folder name: D:\PNP\DWBI Browse...

Include subfolders ☐

Limit search to ☐

Wildcard (RegExp)

Help OK Cancel

(folder yang saya pilih ada isinya)

- Pada Entri **Check if Files Exist**

Checks if files exist

Job entry name: Checks if files exist

File/folder name: Add File... Folder...

Files/Folders:

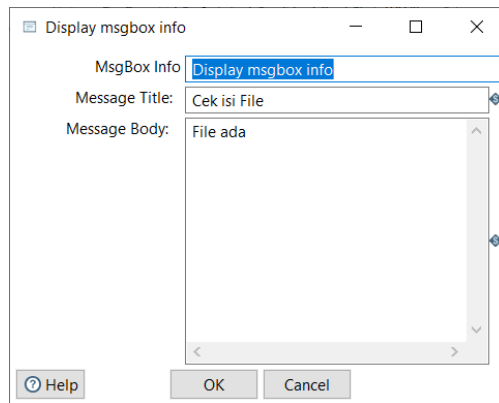
#	File/Folder
1	D:\PNP\DWBI\Praktikum 1.kjb

Delete Edit

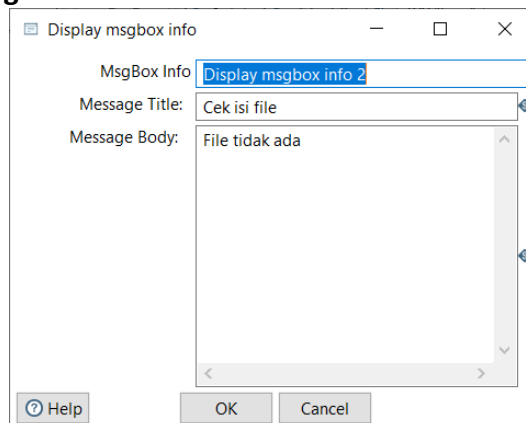
Help OK Cancel

- Tombol **File** digunakan untuk memilih file yang akan dicek ada atau tidaknya.
- Tombol **Folder** digunakan untuk memilih folder yang akan dicek ada atau tidaknya.
- Tombol **Add** diklik untuk menambahkan ke daftar file/folder yang akan dicek.

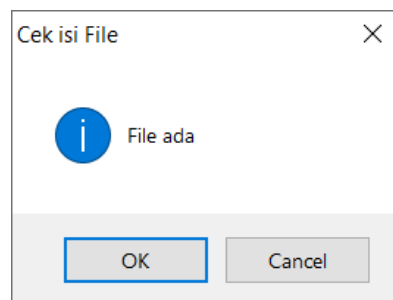
- Pada Entri **Display msgbox info**



- Pada Entri **Display messagebox info 2**

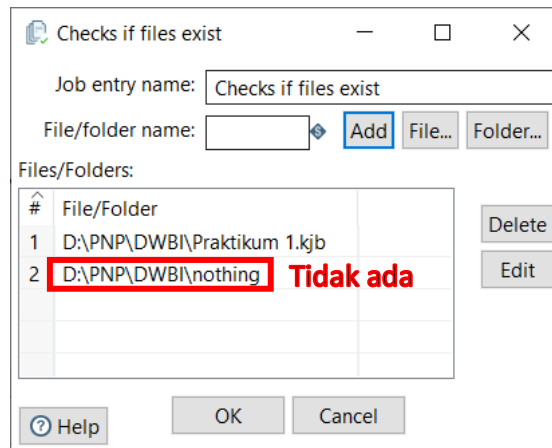


3. Hasil dari Job yang dijalankan

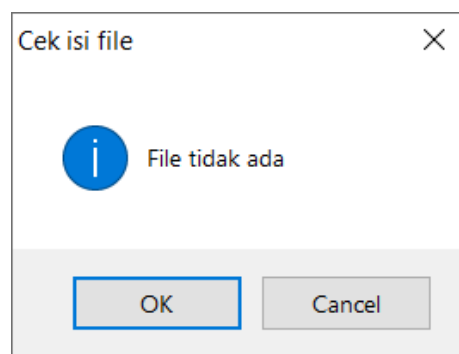


TUGAS: JELASKAN DAN ANALISA PROSES APA YANG TERJADI PADA PERCOBAAN MENGECEK KEBERADAAN FILE/FOLDER.

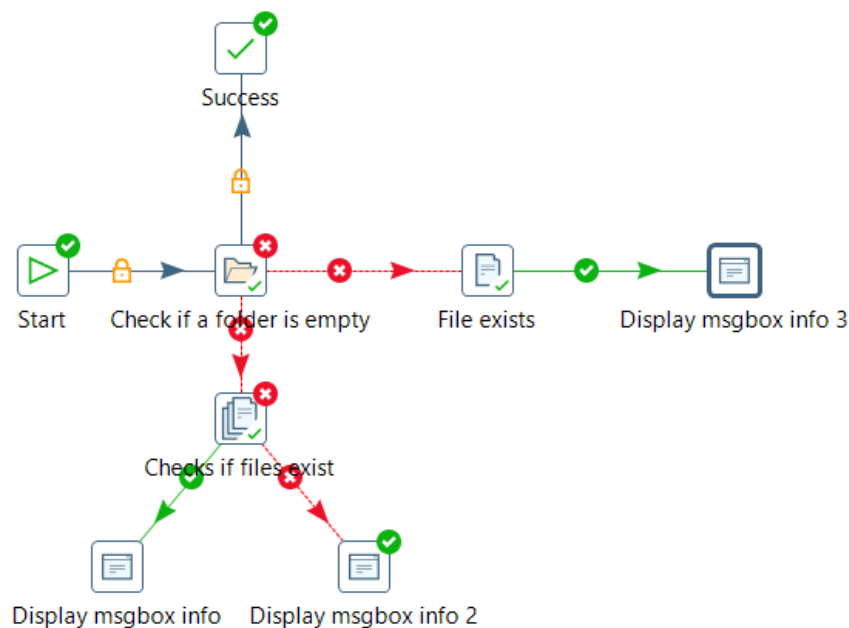
- Percobaan A, dimana file/folder yang dicek tidak ada salah satunya pada komputer.
 - Pada Entri **Check if Files Exist**



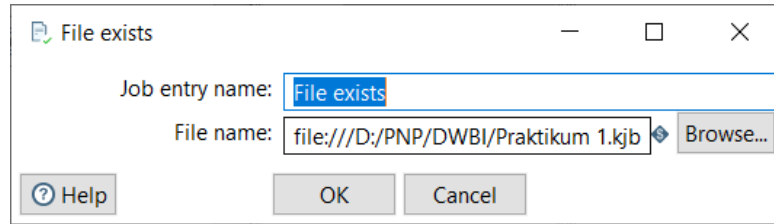
- Hasil Percobaan A



- Karena kondisi atau aturan yang dipakai adalah AND pada entri **Checks if files exist**, maka akan bernilai false jika salah satu file tidak ada.
5. Percobaan B, dengan menambahkan entri **file exist** yang berada di Node **Conditions** pada Percobaan A.
- Tambahkan entri **file exist** dan **Display msgbox info 3** pada struktur jobs sebelumnya sehingga seperti gambar dibawah ini.

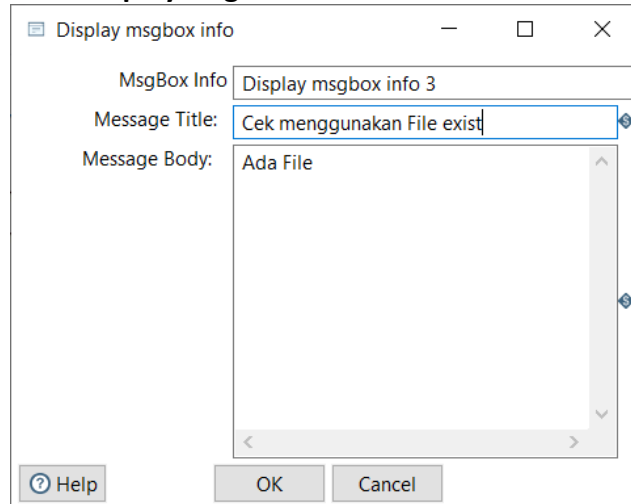


- Pada entri **File exist**

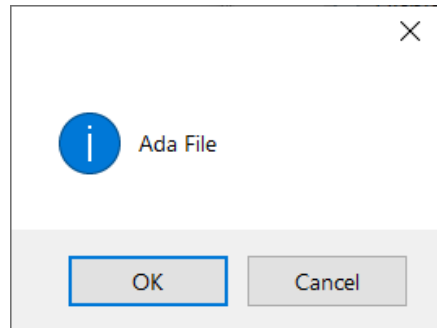


Pada tombol browse pilih file yang ada pada komputer.

- Pada **Display msgbox info 3**



- Hasil Percobaan B



TUGAS: ALUR DARI JOB PADA PERCOBAAN B MENGHASILKAN TAMPILAN PADA DISPLAY MSGBOX INFO 3, JELASKAN KENAPA BISA SEPERTI ITU.

4. TUGAS

1. Lakukan semua percobaan yang telah dijelaskan pada modul diatas dan buat laporannya. **Pada tahap Instalasi tidak perlu dibuat laporannya.**
2. Dalam laporan sesuaikan tampilan screenshoot dari hasil praktikum sesuai perintah yang di tugaskan **(tulisan tugas yang di cetak tebal).**
3. Berikan analisa dan kesimpulan percobaan yang telah dilakukan.

FORMAT LAPORAN, terdiri dari:

- A. Cover
- B. Dasar Teori
- C. Langkah dan Hasil Percobaan
- D. Analisa dan Kesimpulan Percobaan
- E. Referensi