



Pusat Pendidikan dan Pelatihan APU PPT

# **TEKNIK PENGOLAHAN DATA LANJUTAN**

PUSAT PENDIDIKAN DAN PELATIHAN APU PPT

PUSAT PELAPORAN DAN ANALISIS TRANSAKSI KEUANGAN

2017

# **Teknik Pengolahan Data Lanjutan**

Dikeluarkan oleh Pusat Pendidikan dan Pelatihan APU PPT PPATK.

Edisi Ke- 1 : 2017

Penyusun : Muhamad Agung Arif Wicaksono  
Pereviu : Ardhiyan Dwiyoenanto dan Maryanto  
Editor : Tania Rianti Kamalia

Pusdiklat APU PPT  
Jl. Tapos Raya No.82, Cimpaeun, Tapos, Kota Depok, Jawa Barat 16459  
Telp. :  
Fax. :  
Email : [pusdiklat@ppatk.go.id](mailto:pusdiklat@ppatk.go.id)  
Website : [pusdiklat.ppatk.go.id](http://pusdiklat.ppatk.go.id)

**Dilarang keras mengutip, menjiplak atau menggandakan sebagian atau seluruh isi modul ini serta memperjualbelikan tanpa izin tertulis dari Pusat Pendidikan dan Pelatihan APU PPT PPATK.**

## KATA PENGANTAR

Puji syukur kehadiran Allah SWT yang karena kesempatan-Nya kami dapat menyelesaikan modul tentang "Teknik Pengolahan Data Lanjutan". Modul ini di buat sebagai panduan pengajar dalam melaksanakan pelatihan Audit Khusus dan Kepatuhan Anti Pencucian Uang dan Pendanaan Terorisme Lanjutan.

Modul ini diharapkan dapat membantu dan memandu pengajar dalam memberikan pelatihan terkait materi tersebut. Modul bersifat dinamis dan dapat disesuaikan dengan perkembangan yang ada. Oleh karena itu, masukan dan kritik terhadap modul ini sangat diharapkan untuk perbaikan dan kemajuan modul ini dalam memandu pengajar memberikan materi ini.

Demikian yang dapat kami sampaikan, semoga modul ini dapat bermanfaat dan dapat digunakan oleh para pengajar sehingga dapat dengan optimal tersampaikan materi yang akan diajarkan.

Depok, Desember 2017  
Kepala Pusdiklat APU PPT – PPATK

Yusup Darmaputra  
197706152002121007

## DAFTAR ISI

KATA PENGANTAR.....	iii
DAFTAR ISI.....	iv
DAFTAR TABEL.....	iv
DAFTAR GAMBAR .....	vi
I. PENDAHULUAN .....	1
A. Latar Belakang .....	1
B. Deskripsi Singkat .....	1
C. Manfaat Modul .....	1
D. Tujuan Pembelajaran .....	1
E. Metode Pembelajaran .....	1
F. Materi Pokok dan Sub Materi Pokok .....	1
G. Petunjuk Belajar .....	2
II. TEKNIK PENGOLAHAN DATA LANJUTAN.....	3
A. Pengolahan Data Audit Menggunakan ACL.....	3
III. PENUTUP .....	19
A. Rangkuman.....	19
B. Evaluasi.....	19
C. Kunci Jawaban .....	21
DAFTAR PUSTAKA .....	22

## DAFTAR TABEL

Tabel 1. <i>Bit and Character function</i> .....	12
Tabel 2. <i>Conversion function</i> .....	12
Tabel 3. <i>Date and Time function</i> .....	13
Tabel 4. <i>Financial function</i> .....	14
Tabel 5. <i>Logical function</i> .....	15
Tabel 6. <i>Math function</i> .....	15
Tabel 6. <i>Math function</i> .....	16
Tabel 4. <i>Miscellaneous function</i> .....	17

## DAFTAR GAMBAR

Gambar 1. Membuat <i>Project</i> .....	3
Gambar 2. <i>Importing</i> dan <i>formatting</i> data audit (1).....	3
Gambar 3. <i>Importing</i> dan <i>formatting</i> data audit (2).....	4
Gambar 4. <i>Importing</i> dan <i>formatting</i> data audit (3).....	4
Gambar 5. <i>Importing</i> dan <i>formatting</i> data audit (4).....	4
Gambar 6. <i>Importing</i> dan <i>formatting</i> data audit (5).....	5
Gambar 7. <i>Importing</i> dan <i>formatting</i> data audit (6).....	5
Gambar 8. <i>Importing</i> dan <i>formatting</i> data audit (7).....	6
Gambar 9. <i>Importing</i> dan <i>formatting</i> data audit (8).....	6
Gambar 10. <i>Importing</i> dan <i>formatting</i> data audit (9).....	7
Gambar 11. <i>Importing</i> dan <i>formatting</i> data audit (10).....	7

## **I. PENDAHULUAN**

### **A. Latar Belakang**

Adanya kebutuhan pengetahuan pegawai PPATK terkait teknik pengolahan data audit lanjutan untuk tujuan menilai kepatuhan Pihak Pelapor dalam melaporkan Transaksi Keuangan Mencurigakan, Transaksi Keuangan Tunai, Transaksi Keuangan dari dan ke Luar Negeri dan Laporan Transaksi.

### **B. Deskripsi Singkat**

Program diklat ini bertujuan untuk menguraikan cara mengolah data yang lebih kompleks untuk mendapatkan tujuan audit yang ingin dicapai.

### **C. Manfaat Modul**

Sebagai sumber rujukan dalam mempelajari dan memahami teknik pengelolaan data audit lanjutan.

### **D. Tujuan Pembelajaran**

#### **1. Kompetensi Dasar**

Peserta diklat diharapkan mampu melakukan pengolahan data yang lebih kompleks.

#### **2. Indikator Keberhasilan**

Setelah mempelajari modul ini, peserta diharapkan mampu melakukan pengolahan data yang lebih kompleks.

### **E. Metode Pembelajaran**

Penyampaian materi diklat menggunakan metode sebagai berikut:

1. Ceramah;
2. Tanya Jawab;
3. Curah Pendapat;
4. Diskusi;
5. Simulasi; dan
6. Praktek.

### **F. Materi Pokok dan Sub Materi Pokok**

Materi pokok untuk mata diklat Teknik “Pengolahan Data Lanjutan” yaitu penggunaan *ACL (Audit Command Language)* untuk pengolahan data audit yang besar dan kompleks.

### **G. Petunjuk Belajar**

Kami harapkan seluruh peserta diklat mengikuti langkah-langkah di bawah ini agar proses pembelajaran dapat berjalan dengan lancar dan tujuan pembelajaran tercapai dengan baik:

1. Bacalah secara teliti dan pahami tujuan pembelajaran yang tertulis pada setiap awal bab;
1. Pelajari setiap bab secara berurutan, mulai dari Bab I Pendahuluan hingga Bab Penutup;
2. Kerjakan secara sungguh-sungguh dan tuntas setiap evaluasi pada setiap akhir bab;
3. Keberhasilan proses pembelajaran dalam mata diklat ini tergantung pada kesungguhan Anda. Untuk itu, belajarlah secara mandiri atau berkelompok;
4. Anda disarankan mempelajari bahan-bahan dari sumber lain seperti yang tertera pada Daftar Pustaka pada akhir modul ini, dan jangan segan-segan bertanya kepada widyaiswara atau teman yang telah memahami tentang mata diklat ini.

Baiklah, selamat belajar! Semoga Anda sukses menerapkan pengetahuan dan keterampilan yang diuraikan dalam mata diklat ini dalam melaksanakan tugas sehari-hari anda di kantor.



## II. TEKNIK PENGOLAHAN DATA LANJUTAN

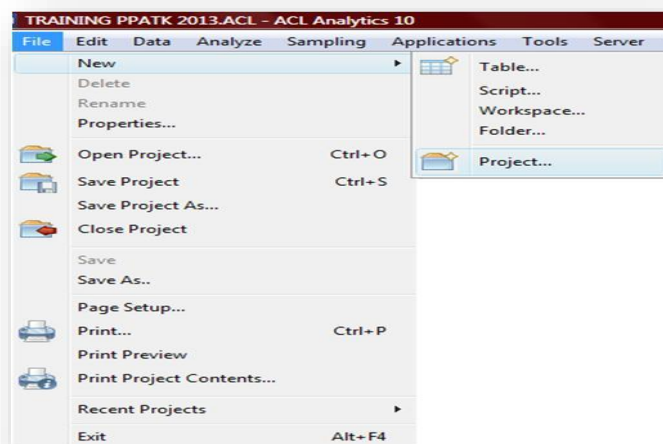
### Indikator Keberhasilan:

Peserta dapat memahami pengolahan data yang lebih kompleks.

### A. Pengolahan Data Audit Menggunakan ACL

#### 1. Membuat *Project*

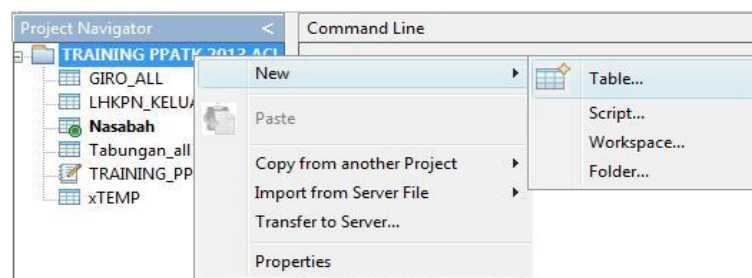
*File --> New --> Project --> Beri nama Project dan tentukan lokasi penyimpanan --> Save*



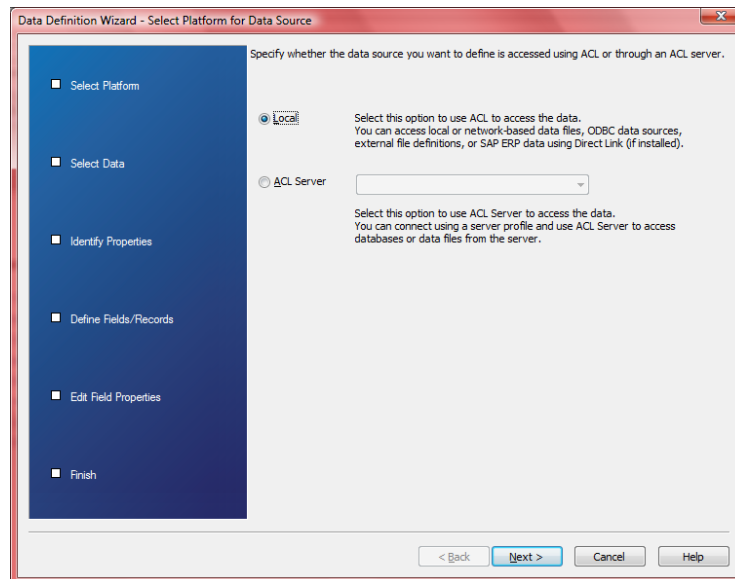
Gambar 1. Membuat *Project*

#### 2. *Importing* dan *formatting* Data Audit

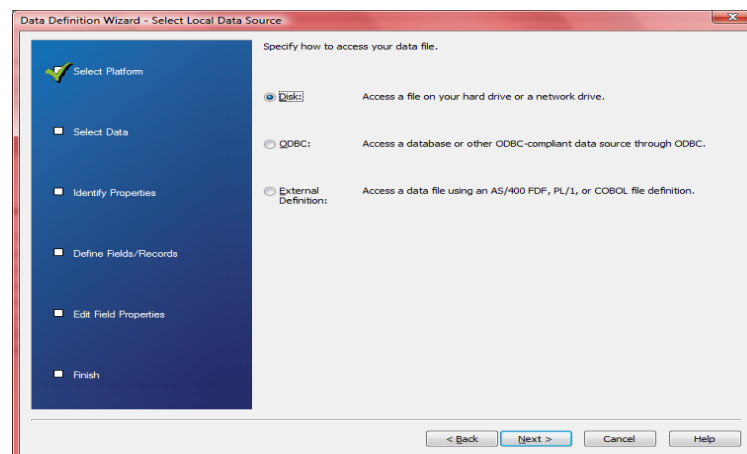
*File --> New --> Table --> Local Next --> Disc Next --> Pilih File yang akan di-import --> open --> pilih tipe file yang akan di-import --> Pilih tabel yang akan di-import --> pilih format data yang sesuai dari masing-masing kolom --> pilih lokasi penyimpanan --> Finish.*



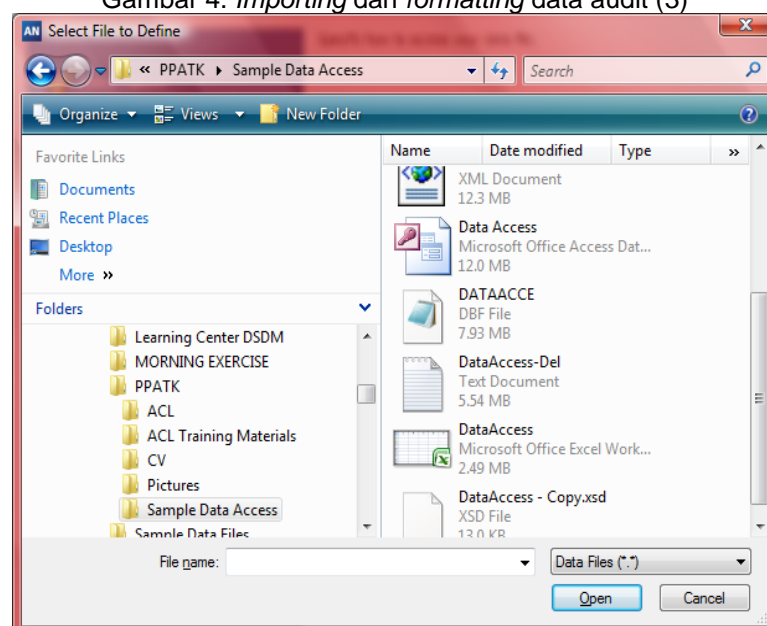
Gambar 2. *Importing* dan *formatting* data audit (1)



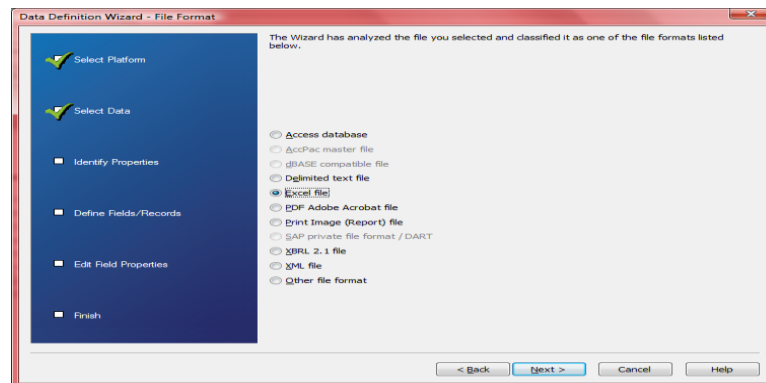
Gambar 3. *Importing dan formatting data audit (2)*



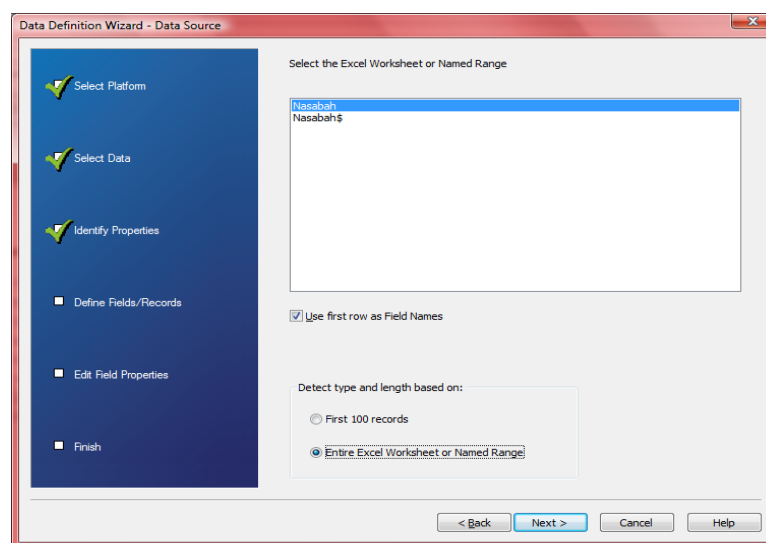
Gambar 4. *Importing dan formatting data audit (3)*



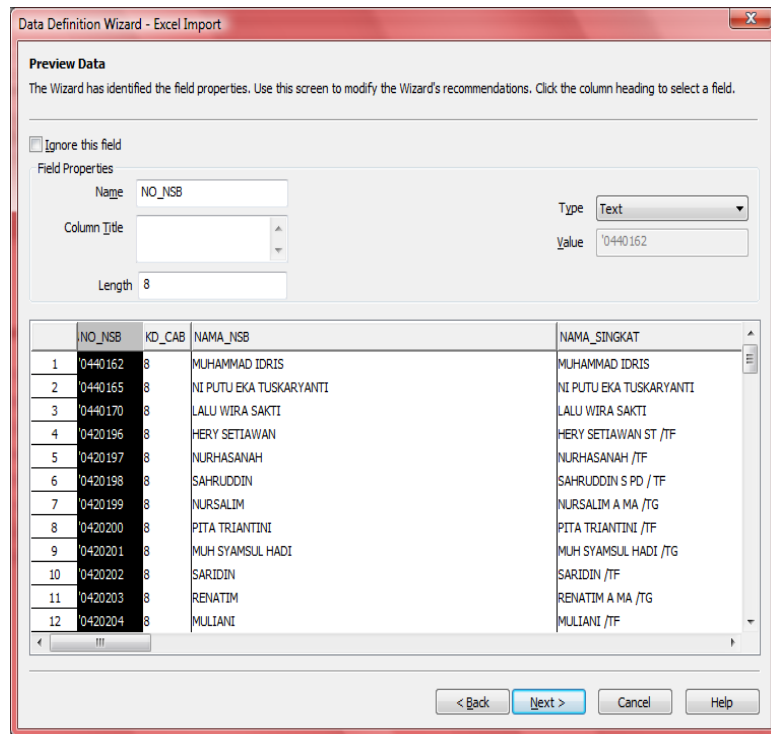
Gambar 5. *Importing dan formatting data audit (4)*



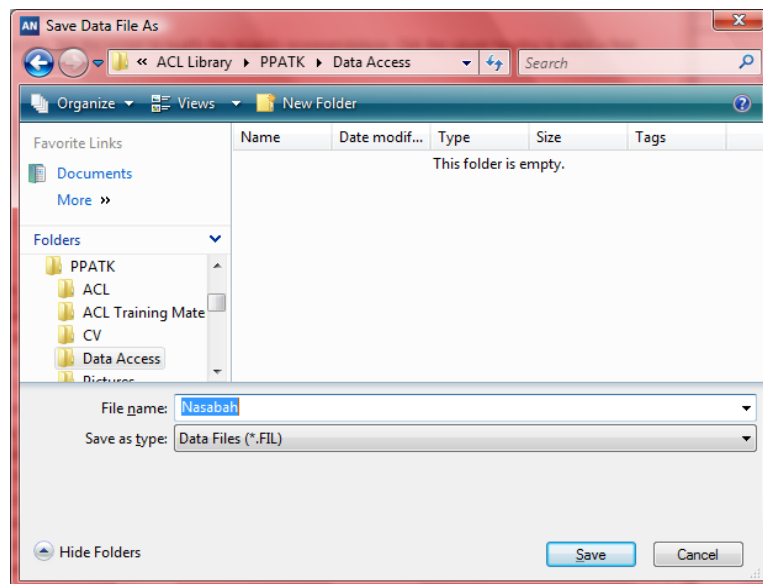
Gambar 6. *Importing dan formatting data audit (5)*



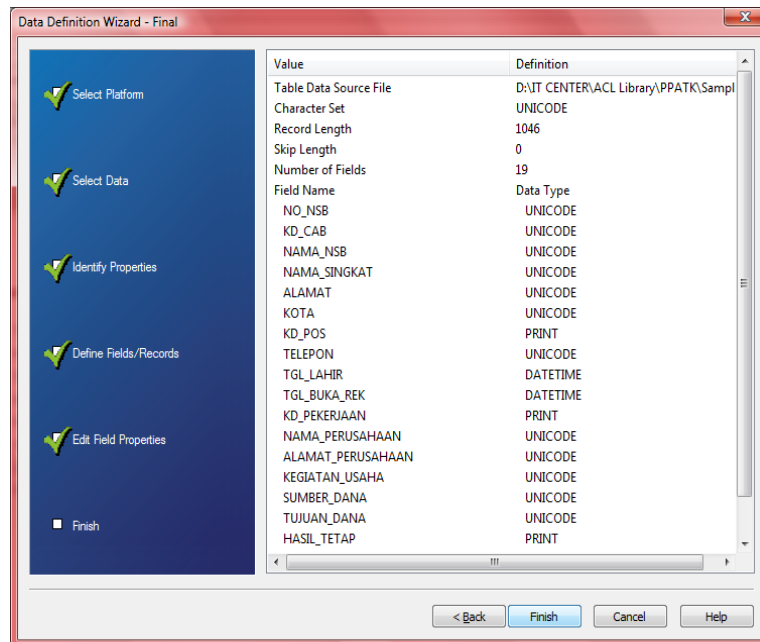
Gambar 7. *Importing dan formatting data audit (6)*



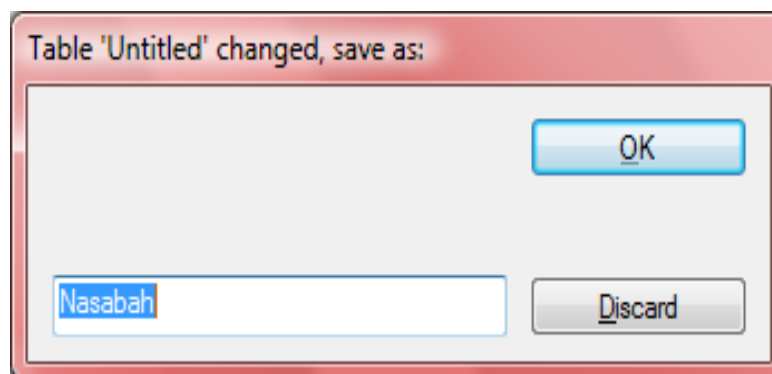
Gambar 8. Importing dan formatting data audit (7)



Gambar 9. Importing dan formatting data audit (8)



Gambar 10. *Importing dan formatting data audit (9)*



Gambar 11. *Importing dan formatting data audit (10)*

### 3. Pengecekan Integritas Data Audit

Menganalisa data berdasarkan *field*, untuk memastikan bahwa data konsisten dengan jenis *field* yang dimaksud. Untuk memastikan tidak adanya *error* pada setiap *record* dan *field*-nya.

Buka tabel --> Data --> *Verify* --> Pilih *field* yang akan di cek --> Ok

### 4. Memahami Data Audit

#### a) *Count*

Perintah untuk menghitung banyaknya *record* (baris) dalam suatu *file*.

Buka tabel --> *Analyze* --> *Count record* --> Ok.

#### b) *Total*

Perintah untuk menampilkan total satu atau lebih *field* (kolom) numerik dalam suatu *file*.

Buka tabel --> *Analyze* --> *Total Field* --> Pilih *Field* yang akan di total --> Ok.

c) *Statistics*

Perintah untuk memberikan gambaran statistik *field-field* numerik. Informasi yang disediakan antara lain jumlah *record*, total, rata-rata, *range* antara nilai tertinggi dan terendah, nilai tertinggi/terendah (*highest/lowest*).

Buka tabel --> *Analyze* --> *Statistical* --> *Statistics* --> Pilih *statistics on* --> Ok.

d) *Profile*

Perintah ini memberikan ringkasan informasi statistik dari satu atau beberapa *field* numerik pada suatu file input yaitu nilai total, nilai absolut, nilai minimum, nilai maksimum. Hasil perintah *profile* akan digunakan menjadi nilai *default* perintah *STRATIFY*, *HISTOGRAM* DAN *SAMPLE*.

Buka tabel --> *Analyze* --> *profile*--> memilih *profile field* --> Ok.

e) *Sequance*

Perintah untuk menentukan atau memeriksa apakah *key field* dalam suatu *input file definition* (IFD) berada dalam susunan *sequential* (keterurutan data)

Buka tabel --> *Analyze* --> *Examine sequance* --> memilih *sequance on* --> Ok.

f) *Duplicate*

Perintah untuk menemukan adanya duplikasi record dan mencari apakah suatu field tertentu berisi nilai yang unik (*Completeness*).

Buka tabel --> *Analyze* --> *look for duplicate* --> memilih *duplicate on* --> Ok.

g) *Gaps*

Perintah untuk memverifikasi bahwa seluruh *record* dalam suatu *sequence* telah lengkap dan untuk mendeteksi apakah *key field* dalam *file* berisi *gaps* dalam pengurutannya (*Uniqueness*).

Buka tabel --> *Analyze* --> *look for gaps* --> memilih *gaps on* --> Ok.

## 5. Analisa Data

### a) *Extract*

1. Untuk mencari *record* dan memisahkannya ke file lain.

Buka data yang akan di extract --> *Data* --> *Extract Data* --> Pilih *Field* yang akan di *extract* --> tentukan syarat atau kriteria data yang mau diambil --> To "pilih lokasi penyimpanan" --> Ok.

2. Untuk menggabungkan data dari beberapa file

Buka data yang akan di gabung --> *Data* --> *Extract Data* --> Pilih *Field* yang akan di gabung --> *more "append to existing file"* --> To "pilih file tujuan penggabungan" --> Ok.

### b) *Export*

Digunakan untuk membuat *file* dengan format lain selain ACL (. *FIL*) dari data yang ada pada suatu input *file*.

Buka tabel --> *Data* --> *Export to other application*--> Pilih *Field* yang akan di *export* --> pilih *export As* --> Pilih *export with field names* --> To "menentukan tempat penyimpanan" --> Ok.

### c) *Sort*

Untuk mengurutkan data berdasarkan satu atau beberapa field tertentu baik secara *ascending* (awal ke akhir) maupun *descending* (akhir ke awal). Menghasilkan file baru (input file dan data file).

Buka tabel --> *Data* --> *Sort record* --> *sort on* --> Ok.

### d) *Index*

Tidak menghasilkan file data baru melainkan file indeks yang berisi *pointer* ke *file* data yang bersangkutan serta disusunurut sesuai kriteria yang digunakan pada saat indeks dibuat. Dapat mengurutkan *record* atas dasar beberapa *field* sekaligus.

### e) *Classify*

Mengelompokkan *record* berdasarkan kategori tertentu yang diwakili *field* jenis karakter. Batas maksimal kategori yang dapat dibuat dengan perintah *Classify* sebanyak 10.000 kelompok atau lebih sedikit tergantung pada memori RAM yang dimiliki.

Buka tabel --> *Analyze*--> *classify* --> *classify on* --> Ok.

### f) *Summarize*

Fungsi *Summarize* sama dengan *Classify* yaitu digunakan untuk mengelompokkan *record* berdasarkan kategori tertentu yang diwakili *field* jenis *character*. Perbedaan dengan *Classify*, fungsi *Summarize* bisa menggunakan lebih dari satu *key field* dengan tipe *Character* (ASCII) maupun *Date*.

Buka tabel --> *Analyze*--> *Summarize*--> pilih *summarize on* --> pilih sub total field --> pilih *other field*--> *output* --> *name* --> tentukan tempat penyimpanan --> Ok.

g) *Stratify*

Memungkinkan untuk membuat pengelompokan *record* berdasarkan *range* nilai tertentu.

Buka tabel --> *Analyze*--> *Stratify*--> pilih *stratify on* --> pilih sub total *field* --> Ok.

h) *Age*

Mengelompokkan data *record* berdasarkan umur suatu transaksi yang didapat dari selisih data *field* berjenis *date* tertentu dengan data jenis *date* yang lain. Anda dapat mengurangi dengan suatu tanggal tertentu (*cut-off date*), jika tidak ACL akan mengambil tanggal komputer. Perintah *Age* akan mengelompokkan *record* sekaligus membuat sub total *field numeric* lainnya. Perintah *Age* akan membuat *field* baru dengan nama *Count* yang berisi jumlah *record* pada kelompok yang sama.

Buka tabel --> *Analyze*--> *Age* --> pilih *age on* --> pilih sub total *field*--> Ok.

i) *Join*

Menggabungkan 2 buah tabel yang mempunyai hubungan relational menjadi satu *file* output baru.

Syarat-syarat melakukan *JOIN*:

1. *Primary File* dan *Secondary File* harus berasal dari file proyek yang sama;
2. Terdapat field penghubung yang sama diantara kedua file tersebut;
3. *Field* penghubung berjenis *character* dan memiliki panjang yang sama;
4. *Secondary File* harus tersusun secara urut (*ascending*) secara fisik berdasarkan field penghubungnya;
5. Apabila hubungan bersifat *one-to-many*, maka *secondary file* harus diringkas terlebih dahulu; dan



6. Apabila *Primary File* belum tersusunurut berdasarkan field penghubungnya, maka aktifkan status *presort* yang terdapat pada jendela perintah *JOIN* untuk memaksa ACL melakukan penyortiran terlebih dahulu sebelum melakukan penggabungan;

Buka tabel -->Data-->*Join Table* --> Pilih *primary key* --> pilih *secondary table* --> pilih *secondary key* --> To "lokasi penyimpanan" --> Ok.

j) *Relation*

*Relation* merupakan *statement* yang menyatakan bahwa *file input* tempat *statement* tersebut disimpan memiliki hubungan relational dengan *file input* lain. *JOIN* menghubungkan tabel secara bertahap, maka *Relation* dapat mengikat hubungan satu *file* induk dengan banyak *file* input lainnya sekaligus.

Syarat membuat *Relation*:

1. *File* induk dan *file* pasangan berasal dari *file* proyek yang sama;
2. Satu *file* induk boleh mempunyai banyak pasangan;
3. Terdapat *field* penghubung yang sama antara *field* induk dengan *field* pasangan;
4. Kedua *field* penghubung harus berjenis *character* dan memiliki panjang yang sama; dan
5. *File* pasangan harus memiliki *file* indeks berdasarkan *field* penghubung dengan *file* induk.

Buka tabel -->Data-->*Relate Table* --> klik kanan *add table* --> tarik garis pada tabel satu ke tabel dua.

k) *Merge*

Menggabungkan data secara bertahap, setiap 2 tabel digabungkan terlebih dahulu, kemudian hasilnya digabungkan dengan tabel ke-3, dst.

Buka tabel -->Data-->*Merge Table* --> Pilih *Primary key* --> pilih *secondary table* --> pilih *secondary key* --> To "lokasi penyimpanan" --> Ok.

## 6. *Function*

### a) *Bit and Character functions*

Tabel 1. *Bit and Character function*

ASCII( )	Returns the ASCII character code for a specified character.
BIT( )	Returns the binary representation for the specified byte position in the current record.
BYTE( )	Returns the character stored in the specified byte position in the current record.
CHR( )	Returns the character associated with the specified ASCII character code.
DBYTE( )	Returns the Unicode character located at the specified byte position in a record.
DHEX( )	Converts a Unicode string to a hexadecimal string.
DIGIT( )	Returns the upper or lower digit of a specified Packed data type byte.
HEX( )	Returns a hexadecimal string representing the contents of the specified value.
HTOU( )	Converts a hexadecimal string to a Unicode string. Abbreviation for "Hexadecimal to Unicode".
MASK( )	Returns the individual bits from the first byte of a character expression.
SHIFT( )	Returns a single character string with the bits in the first character of the input value shifted to the left or right.

### b) Conversion functions

Tabel 2. *Conversion function*

BINTOSTR( )	Returns Unicode character data converted from ZONED or EBCDIC character data. Abbreviation for "Binary to String".
DEC( )	Returns a numeric expression with the specified number of decimal places.
EBCDIC( )	Returns a string that has been converted to EBCDIC character encoding.
PACKED( )	Returns numeric data converted to the Packed data type.
STRING( )	Converts a numeric value to a character string.
UNSIGNED( )	Returns numeric data converted to the Unsigned data type.
VALUE( )	Converts a character string to a numeric value.
ZONED( )	Converts a numeric value to the Zoned data type.

### c) Date and Time functions

Tabel 3. *Date and Time function*

AGE( )	Returns the age, in days, of a specified date compared to a specified cutoff date, or the current operating system date.
CDOW( )	Returns the name of the day of the week for a specified date or datetime. Abbreviation for "Character Day of Week".
CMOY( )	Returns the name of the month of the year for a specified date or datetime. Abbreviation for "Character Month of Year".
CTOD( )	Converts a character or numeric date value to a date. Can also extract the date from a character or numeric datetime value and return it as a date. Abbreviation for "Character to Date".
CTODT( )	Converts a character or numeric datetime value to a datetime. Abbreviation for "Character to Datetime".
CTOT( )	Converts a character or numeric time value to a time. Can also extract the time from a character or numeric datetime value and return it as a time. Abbreviation for "Character to Time".
DATE( )	Extracts the date from a specified date or datetime and returns it as a character string. Can also return the current operating system date.
DATETIME( )	Converts a datetime to a character string. Can also return the current operating system datetime.
DAY( )	Extracts the day of the month from a specified date or datetime and returns it as a numeric value.
DOW( )	Returns a numeric value (1 to 7) representing the day of the week for a specified date or datetime. Abbreviation for "Day of Week".
DTOU( )	Converts a date to a Unicode string, which allows dates in a numeric format to be displayed in various languages. Abbreviation for "Date to Unicode".
EOMONTH( )	Returns the date of the last day of the month that is the specified number of months before or after a specified date.
GOMONTH( )	Returns the date that is the specified number of months before or after a specified date.
HOURL( )	Extracts the hour from a specified time or datetime and returns it as a numeric value.
MINUTE( )	Extracts the minutes from a specified time or datetime and returns it as a numeric value.
MONTH( )	Extracts the month from a specified date or datetime and returns it as a numeric value.
NOW( )	Returns the current operating system time as a Datetime data type.

SECOND( )	<i>Extracts the seconds from a specified time or datetime and returns it as a numeric value.</i>
STOD( )	<i>Converts a serial date – that is, a date expressed as an integer – to a date value. Abbreviation for “Serial to Date”.</i>
STODT( )	<i>Converts a serial datetime – that is, a datetime expressed as an integer, and a fractional portion of 24 hours – to a datetime value. Abbreviation for “Serial to Datetime”.</i>
STOT( )	<i>Converts a serial time – that is, a time expressed as a fractional portion of 24 hours, with 24 hours equaling 1 – to a time value. Abbreviation for “Serial to Time”.</i>
TIME( )	<i>Extracts the time from a specified time or datetime and returns it as a character string. Can also return the current operating system time.</i>
TODAY( )	<i>Returns the current operating system date as a Datetime data type.</i>
UTOD( )	<i>Converts a Unicode string to a date, which allows dates in various languages to be displayed in a numeric format. Abbreviation for “Unicode to Date”.</i>
YEAR( )	<i>Extracts the year from a specified date or datetime and returns it as a numeric value.</i>

*d) Financial functions*

Tabel 4. Financial function

CUMIPMT( )	<i>Returns the amount of interest paid on a loan during the specified period.</i>
CUMPRINC( )	<i>Returns the amount paid against the principal of a loan for the specified period.</i>
EFFECTIVE( )	<i>Returns the effective annual interest rate on a loan.</i>
FVANNUITY( )	<i>Returns the future value of a series of payments, which represents the sum of the payments plus interest.</i>
FVLUMPSUM( )	<i>Returns the future value of a lump sum after a number of periods.</i>
FVSCHEDULE( )	<i>Returns the future value of an investment calculated at a variable compound rate.</i>
IPMT( )	<i>Returns the size of the interest payment on a loan for a specified period.</i>
NOMINAL( )	<i>Returns the annual nominal interest rate.</i>
NPER( )	<i>Returns the number of periods required to pay off a loan.</i>
PMT( )	<i>Returns the amount of the periodic payment required to pay off a loan.</i>

PPMT( )	Returns the amount of the principal repaid on a loan for a specified period of the term.
PVANNUIITY( )	Returns the present value of a series of payments.
PVLUMPSUM( )	Returns the present value of a lump sum due in a number of periods.
RATE( )	Returns the interest rate per period.

#### e) Logical functions

Tabel 5. Logical function

BETWEEN( )	Returns a logical value indicating whether the specified value falls within a range.
FIND( )	Returns a logical value indicating whether the specified string was found in a particular field, or in the record.
ISBLANK( )	Returns a logical value indicating whether the input value is blank.
ISDEFINED( )	Returns a logical value indicating whether the specified field exists.
ISFUZZYDUP( )	Returns a logical value indicating whether a string is a fuzzy duplicate of a comparison string.
MAP( )	Returns a logical value indicating if a string matches a specified format.
MATCH( )	Returns a logical value indicating if a value matches any of the values it was compared against.
SOUNDSLIKE( )	Returns a logical value indicating whether a string phonetically matches a comparison string.
TEST( )	Returns a logical value indicating whether a specified string was found at a specific byte position in a record.
VERIFY( )	Returns a logical value indicating whether the data in a physical data field is valid.

#### f) Math functions

Tabel 6. Math function

ABS( )	Returns the absolute value of a numeric expression. The absolute value of a number is the number without its sign.
EXP( )	Returns the exponential value (base 10) of a numeric expression with a specified number of decimal places.
INT( )	Returns the integer value of a numeric expression or field value.

LOG( )	Returns the logarithm (base 10) of a numeric expression or field value with a specified number of decimal places.
MAXIMUM( )	Returns the greater value of two numeric values.
MINIMUM( )	Returns the lesser value of two numeric values.
MOD( )	Returns the remainder from dividing two numbers.
RAND( )	Returns a random number between zero and the value of a numeric expression.
ROOT( )	Returns the square root of a numeric expression.
ROUND( )	Returns a rounded whole number for a numeric value.

#### g) String functions

Tabel 7. Math function

ALLTRIM( )	Returns a new string containing a copy of the specified string with leading and trailing spaces removed.
AT( )	Returns a number specifying where a particular occurrence of a search string starts within a character string.
BLANKS( )	Returns a string containing a specified number of blank spaces.
CLEAN( )	Returns a string where the first invalid character, and all subsequent characters, are replaced with blanks.
EXCLUDE( )	Returns a string containing a copy of the specified string with the excluded characters removed.
INCLUDE( )	Returns a string with the specified included characters, and with all other characters removed.
INSERT( )	Returns the original string contents with a specified string inserted in a specified location.
LAST( )	Returns a string containing a specified number of characters from the right side of a string.
LENGTH( )	Returns the length of a string.
LEVDIST( )	Returns the Levenshtein Distance between two specified strings, which is a measurement of how much the two strings differ.
LOWER( )	Returns a string with alphabetic characters converted to lowercase.
LTRIM( )	Returns a string with leading spaces removed from the input value.
OCCURS( )	Returns a count of the number of times a substring occurs in a specified character string.
OMIT( )	Returns a string with one or more specified substrings removed.
PROPER( )	Returns a string with the first character of each word is set to uppercase and the rest are lowercase.



REMOVE( )	Returns a string with all characters removed, except for a specified list of allowed characters.
REPEAT( )	Returns a string that repeats a substring a specified number of times.
REPLACE( )	Replaces all instances of a specified character string with a new character string.
REVERSE( )	Returns a string with the characters in reverse order.
RJUSTIFY( )	Returns a right-justified string the same length as a specified string, with any trailing spaces moved to the left of the string.
SOUNDEX( )	Returns the soundex code for the specified string, which can be used for phonetic comparisons with other strings.
SPLIT( )	Returns a string containing the text between two separators in the specified string.
SUBSTR( )	Returns a string containing only the specified substring from the input value.
TRANSFORM( )	Reverses the display order of bi-directional text within a specified string.
TRIM( )	Returns a new string containing a copy of the specified string with no trailing spaces.
UPPER( )	Returns a string with alphabetic characters converted to uppercase.

#### *h) Miscellaneous functions*

Tabel 8. *Miscellaneous function*

ILESIZE( ) F	Returns the size of the specified file in bytes.
FREQUENCY( )	Returns the expected Benford frequency for sequential leading positive numeric digits to a precision of eight decimal places.
FTYPE( )	Returns a character identifying the data category of a field or variable, or the type of an ACL project item.
LEADING( )	Returns a string containing a specified number of leading digits.
OFFSET( )	Returns the value of field with the starting position offset by a specified number of bytes.
RECLen( )	Returns the length of the current record.
RECNO( )	Returns the current record number.
RECOFFSET( )	Returns a field value from a record that is a specified number of records from the current record.
ZSTAT( )	Returns the standard Z-statistic.





### III. PENUTUP

Selamat Anda telah mempelajari mata ajar “Teknik Pengolahan Data Lanjutan” dengan sukses. Selanjutnya, Anda kami persilahkan untuk mencermati sekali lagi rangkuman/intisari mata ajar ini.

#### A. Rangkuman

Ada enam hal penting yang harus dikuasai dalam melakukan pengolahan data menggunakan ACL yaitu pertama membuat *project*, mengimport *table*, mengecek integritas data, memahami karakteristik data, melakukan analisa data dan menggunakan *function* untuk mengolah data.

#### B. Evaluasi

1. Memastikan bahwa data konsisten dengan jenis *field* yang dimaksud serta memastikan tidak adanya *error* pada setiap *record* dan *field*-nya merupakan fungsi:
  - a. Memahami data
  - b. Pengecekan integritas data
  - c. Analisa Data
  - d. Semua jawaban salah
2. Perintah yang digunakan untuk menghitung banyaknya *record* (baris) dalam suatu *file* yaitu:
  - a. *Count*
  - b. *Statistics*
  - c. *Sequance*
  - d. *Gaps*
3. Berikut ini tidak termasuk perintah yang digunakan untuk memahami data yaitu:
  - a. *Statistics*
  - b. *Gaps*
  - c. *Count*
  - d. *Merge*

4. Perintah yang digunakan untuk menentukan atau memeriksa apakah *key field* dalam suatu *input file definition* (IFD) berada dalam susunan keterurutan data yaitu:
  - a. *Gaps*
  - b. *Sequance*
  - c. *Sort*
  - d. *Index*
5. Perintah yang digunakan untuk membuat *file* dengan format selain ACL yaitu:
  - a. *Export*
  - b. *Classify*
  - c. *Sort*
  - d. *Extract*
6. Perintah yang digunakan untuk mengelompokkan *record* berdasarkan kategori tertentu yang diwakili *field* jenis *character* yaitu:
  - a. *Index*
  - b. *Summarize*
  - c. *Sort*
  - d. *Classify*
7. Perintah yang digunakan untuk mengelompokkan *record* berdasarkan lebih dari satu *key field* dengan tipe *character* (ASCII) maupun *Date* yaitu:
  - a. *Summarize*
  - b. *Stratify*
  - c. *Classify*
  - d. *Join*
8. Perintah yang digunakan untuk membuat pengelompokan *record* berdasarkan *range* nilai tertentu yaitu:
  - a. *Extract*
  - b. *Sort*
  - c. *Classify*
  - d. *Stratify*
9. Perintah yang digunakan untuk mengelompokkan data *record* berdasarkan umur suatu transaksi yang didapat dari selisih data *field* berjenis *date* tertentu dengan data jenis *date* yang lain yaitu:
  - a. *Sort*

- b. *Index*
- c. *Ages*
- d. *Join*

10. Perintah yang digunakan untuk menggabungkan data secara bertahap yaitu:

- a. *Stratify*
- b. *Summarize*
- c. *Merge*
- d. *Relation*

### **C. Kunci Jawaban**

- 1. B
- 2. A
- 3. D
- 4. B
- 5. A
- 6. D
- 7. A
- 8. D
- 9. C
- 10. C

## DAFTAR PUSTAKA

ACL™ Analytics