



## AUDIT COMMAND LANGUAGE

### A. Pengertian ACL

ACL atau Audit Command Language adalah software yang dirancang secara khusus untuk audit, yang didesain untuk melakukan analisa data elektronik suatu perusahaan dan membantu menyiapkan atau menghasilkan laporan audit baik untuk pengguna biasa (*common/ non technical users*) maupun pengguna ahli (*expert users*) secara mudah dan interaktif (saling terkait).

Dengan menggunakan ACL pekerjaan audit akan jauh lebih cepat, dapat dibandingkan dengan proses auditing manual yang memerlukan waktu berjam-jam bahkan sampai berhari-hari. Dengan beberapa kemampuan ACL, analisis data akan lebih efisien dan lebih meyakinkan.

### B. Kemampuan ACL

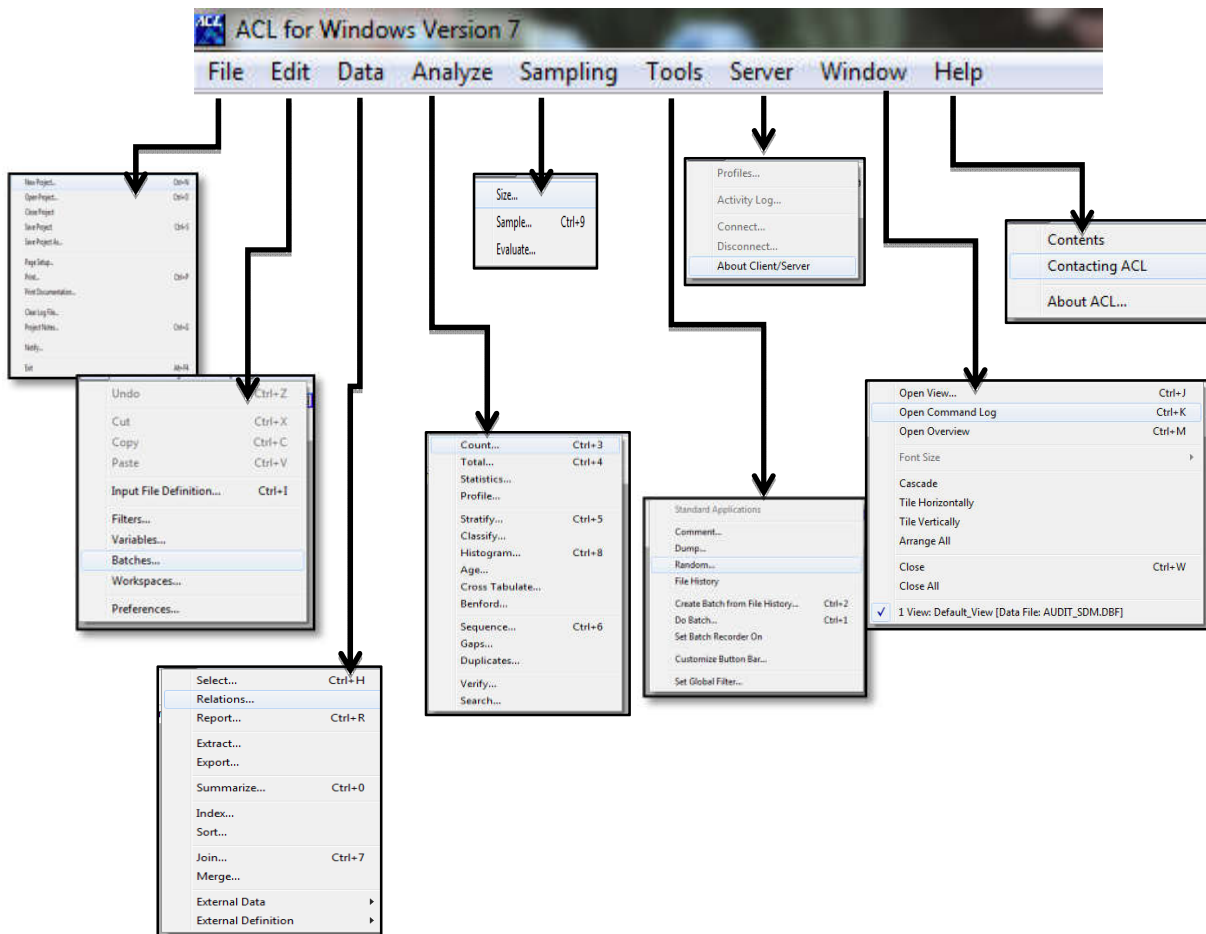
1. Mudah dalam penggunaan (*user friendly*).
2. Built-in audit dan analisis data secara fungsional.
3. Kemampuan untuk mengekspor hasil audit ke berbagai macam format data.
4. Kemampuan menangani ukuran file yang tidak terbatas.
5. Pembuatan laporan yang berkualitas.

### C. Manfaat ACL

1. Dapat membantu dalam mengakses data baik secara langsung (*direct*) ke dalam sistem jaringan ataupun tidak langsung (*indirect*) melalui media lain seperti softcopy dalam bentuk textfile/ report.
2. Menempatkan kesalahan dan potensial “fraud” sebagai pembanding dan menganalisa file-file menurut aturan yang ada.
3. Mengidentifikasi proses perhitungan kembali dan proses verifikasi yang benar.

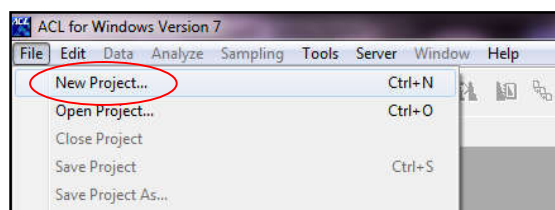
## D. Struktur Menu

Struktur menu pada ACL for Windows versi 7.0 digambarkan sebagai berikut:



## E. Cara Membuat Project Baru pada ACL

1. Masuk ke **ACL for windows V. 7.0**.
2. Pada tampilan utama pilih menu **File**, kemudian pilih **New Project**.



3. Apabila ingin membuka kembali project yang sudah tersimpan, pilih **Open Project**.
4. Setelah memilih **New Project**, ketik nama project anda dalam kotak **File Name**.

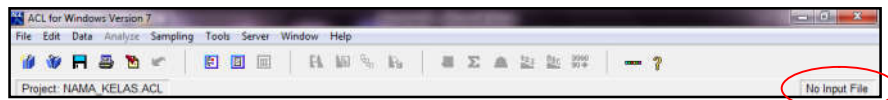


5. Klik **Save**.

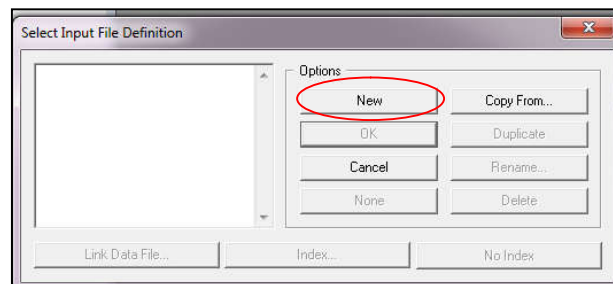


## F. Cara Input File Baru pada ACL

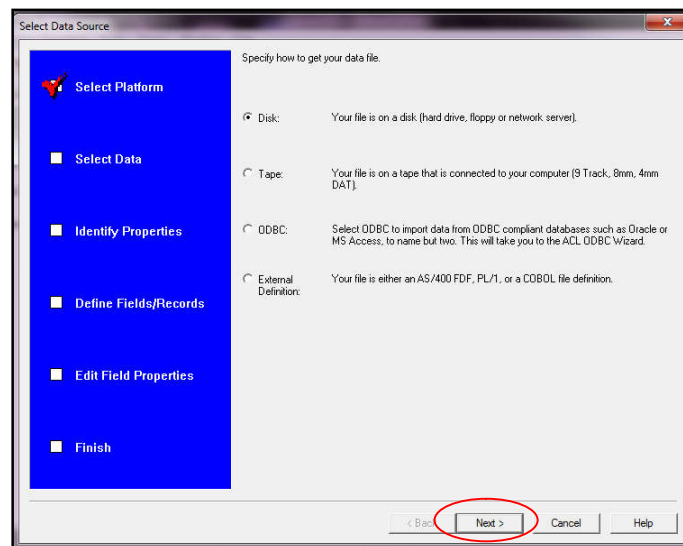
1. Klik **No Input File** pada pojok kanan atas tampilan utama ACL.



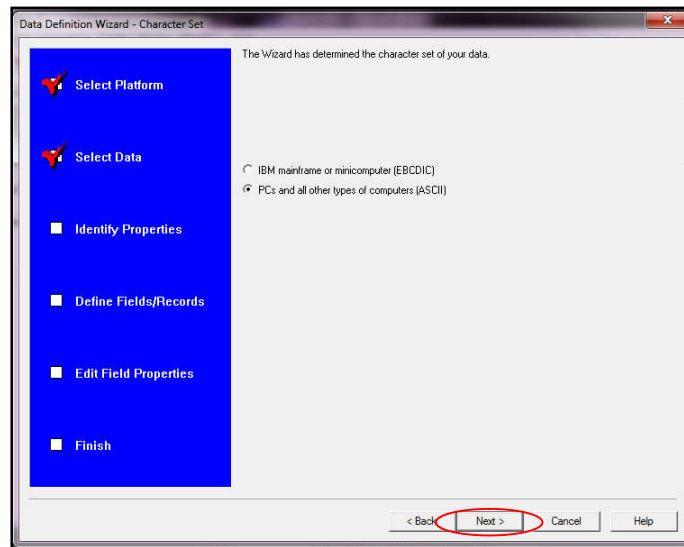
2. Setelah itu akan muncul tampilan Select Input File Definition, pilih **New**.



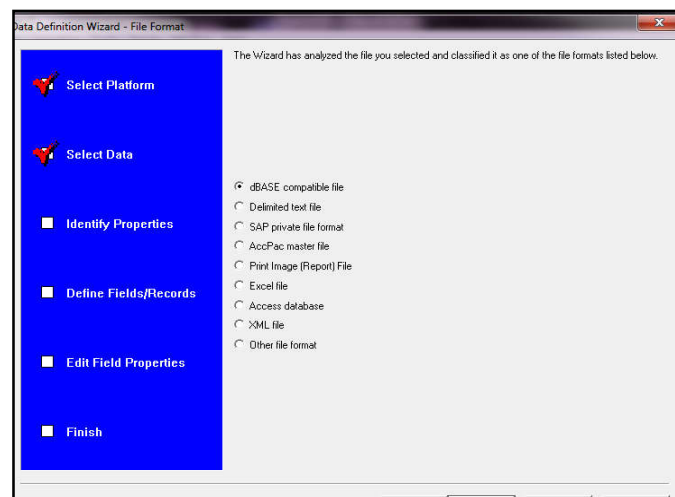
3. Kemudian pada tampilan Select Data Source, pilih radio button **Disk**, lalu **Next**.



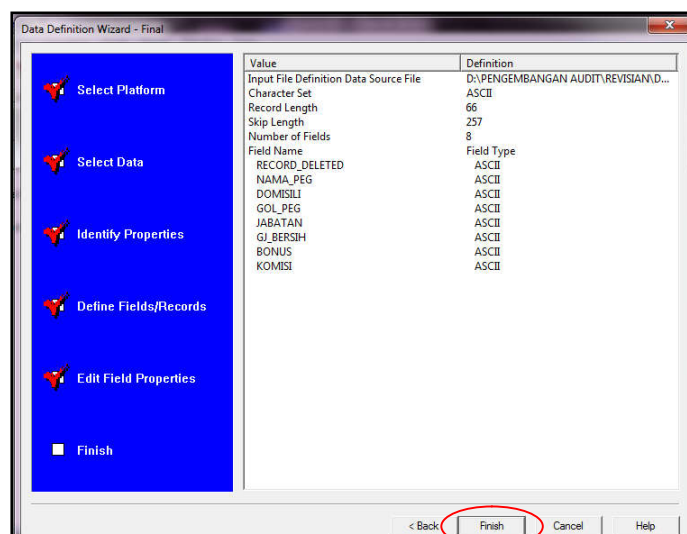
4. Kemudian pilih File yang ingin dibuka.
5. Lalu pada tampilan Data Definition Wizard – CharacterSet, pilih radio button **PCs and all other types of computer**, kemudian pilih **Next**.



6. Lalu pada tampilan Data Definition Wizard – File Format, pilih radio button **dBASE compatible file**, kemudian pilih **Next**.



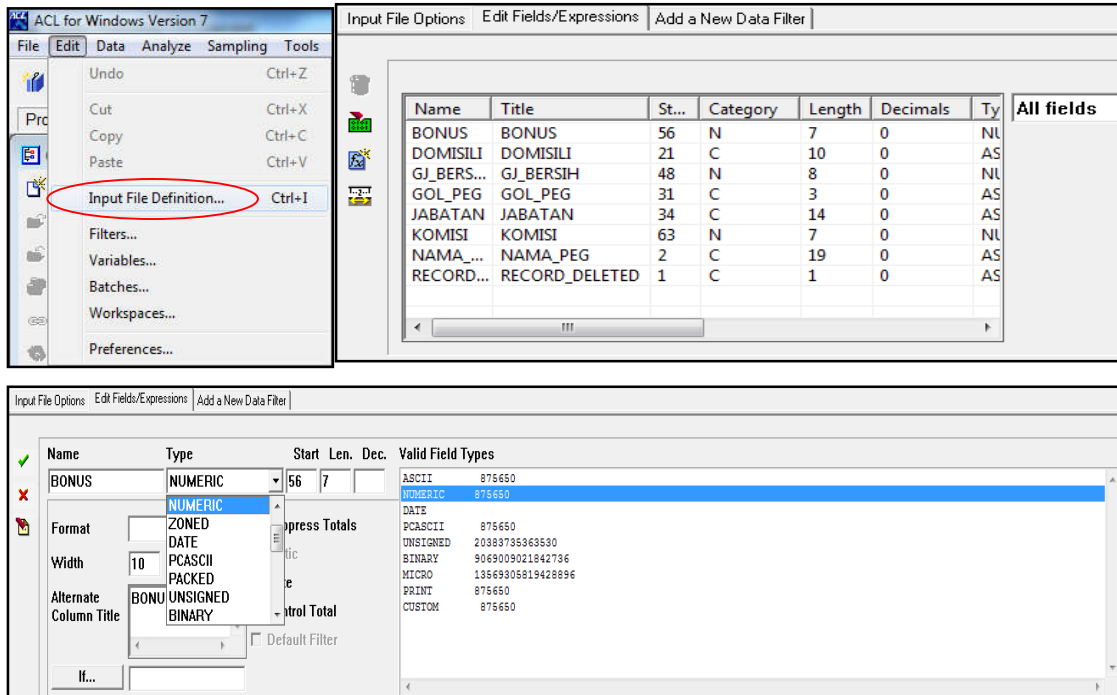
7. Lalu pada **Data Definition Wizard – Final**, pilih **Finish**.



8. Lalu apabila ada tampilan Input File 'Untitled' changed, Save As, klik OK.

### Melakukan Edit Input File Definition

Input File Definition yang sudah didefinisikan dapat di-edit kembali baik diperbaiki, dihapus maupun ditambah field baru. Untuk mengaksesnya pilih menu Edit kemudian pilih Input File Definition.



### G. Cara Untuk Memunculkan Command Log dan Merapihkan Tampilan

- Untuk Memunculkan Command Log:  
Pilih menu Window, lalu pilih Open Command Log.
- Untuk Merapihkan Tampilan:  
Pilih menu Window, lalu pilih Arrange All.

### H. Title, Description, dan Type File

#### 1. File AUDIT\_SDM

Title	Description	Type Field
BONUS	BONUS	NUMERIC
DOMISILI	DOMISILI	ASCII
GJ_BERSIH	GAJI BERSIH	NUMERIC
GOL_PEG	GOLONGAN PEGAWAI	ASCII



JABATAN	JABATAN	ASCII
KOMISI	KOMISI	NUMERIC
NAMA_PEG	NAMA PEGAWAI	ASCII
RECORD_DELETED	-	ASCII

## 2. File DATA\_PEGAWAI

Title	Description	Type
DIVISI	DIVISI	ASCII
DOMISILI	DOMISILI	ASCII
GJ_BERSIH	GAJI BERSIH	NUMERIC
HTG_PER_PR	HUTANG PER PERIODE	NUMERIC
JABATAN	JABATAN	ASCII
JNS_KLMN	JENIS KELAMIN	ASCII
KODE_POS	KODE POS	ASCII
KOTA_ASAL	KOTA ASAL	ASCII
NAMA_PEG	NAMA PEGAWAI	ASCII
NO_PEG	NOMOR PEGAWAI	ASCII
NO_TELP	NOMOR TELPON PEGAWAI	ASCII
RECORD_DELETED	-	ASCII
TGL_LAHIR	TANGGAL LAHIR PEGAWAI	ASCII

## 3. File GAJI

Title	Description	Type
DIVISI	DIVISI	ASCII
GJ_BERSIH	GAJI BERSIH	NUMERIC
GJ_KOTOR	GAJI KOTOR	NUMERIC
JABATAN	JABATAN	ASCII
NO_PEG	NOMOR PEGAWAI	ASCII
PAJAK	PAJAK	NUMERIC
RECORD_DELETED	-	ASCII
TGL_PMBYRN	TANGGAL PEMBAYARAN	ASCII



#### 4. File HUTANG

Title	Description	Type
HRG_SATUAN	HARGA SATUAN	NUMERIC
KODE_PROD	KODE PRODUK	ASCII
NAMA_SUPP	NAMA SUPPLIER	ASCII
NO_INVOICE	NOMOR INVOICE	ASCII
NO_SUPPLIE	NOMOR SUPPLIER	ASCII
RECORD_DELETED	-	ASCII
TGL_INV	TANGGAL INVOICE	ASCII
UNIT_HTG	UNIT HUTANG	NUMERIC

#### 5. File PELANGGAN

Title	Description	Type Field
ALMT_LKAP	ALAMAT LENGKAP	ASCII
KODE_POS	KODE POS	ASCII
KOTA_PLGN	KOTA PELANGGAN	ASCII
NAMA_PLGN	NAMA PELANGGAN	ASCII
NO_PLGN	NOMOR PELANGGAN	ASCII
RECORD_DELETED	-	ASCII

#### 6. File PERS\_PRODUK

Title	Description	Type
HRG_SATUAN	HARGA SATUAN	NUMERIC
JNS_PROD	JENIS PRODUK	ASCII
KODE_PROD	KODE PRODUK	ASCII
NAMA_PROD	NAMA PRODUK	ASCII
PERS_MIN	PERSEDIAAN MINIMAL	NUMERIC
PERS_PROD	PERSEDIAAN PRODUK	NUMERIC
RECORD_DELETED	-	ASCII
STAT_PROD	STATUS PRODUK	ASCII

#### 7. File PIUTANG

Title	Description	Type
HRG SATUAN	HARGA SATUAN	NUMERIC
KODE_PROD	KODE PRODUK	ASCII
NO_INV	NOMOR INVOICE	ASCII
NO_PLGN	NOMOR PELANGGAN	ASCII



RECORD_DELETED	-	ASCII
TGL_INV	TANGGAL INVOICE	ASCII
UNIT_PIUT	UNIT PIUTANG	NUMERIC

#### 8. File SUPPLIER

Title	Description	Type
ALMT_LKAP	ALAMAT LENGKAP	ASCII
KODE_POS	KODE POS	ASCII
KOTA_SUPP	KOTA SUPPLIER	ASCII
NAMA_SUPP	NAMA SUPPLIER	ASCII
NO_SUPPLIE	NOMOR SUPPLIER	ASCII
RECORD_DELETED	-	ASCII





## JOIN

Perintah Join digunakan untuk mengerjakan dua file terbuka secara simultan (berhubungan) atau mencocokkan field-field dari dua file dengan struktur yang berbeda untuk menghasilkan file ke tiga atau file baru.

Jika kita ingin mencocokkan antara dua file, yang harus kita lakukan pertama kali yaitu menentukan primary dan secondary file-nya kemudian lakukan perintah Join dan hasil output merupakan file ke tiga atau file baru.

### 1. Primary dan Secondary File

Primary file dan secondary file adalah dua file yang akan kita cocokkan dalam perintah join.

#### A. Primary File

File pertama yang sedang kita gunakan adalah Primary File. Kita hanya dapat memiliki satu Primary File yang sedang aktif. Primary File tidak selalu merupakan file pertama yang dibuka tetapi merupakan file yang jumlah recordnya terbesar.

#### B. Secondary file

File kedua yang kita buka disebut Secondary File. Kita juga hanya dapat memiliki satu Secondary File yang sedang aktif. Secondary file merupakan file kedua yang dibuka dan merupakan file yang jumlah recordnya lebih kecil.

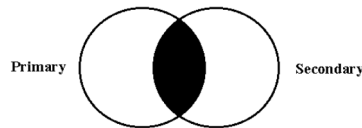
### 2. Perintah SORT

Sebelum melakukan proses Join kita harus melakukan perintah SORT yaitu mengatur record dalam input file secara ascending, berdasarkan field kunci tertentu. Karena SORT menghasilkan file data baru yang mana hasil dari perintah sort yaitu file data baru yang bisa digabungkan. File yang dipakai untuk proses join adalah file yang sudah di sort.

Pada perintah Join, terdapat 2 jenis perintah join yaitu :

#### A. Join Matched

Dalam menggunakan Join Matched Record, pemillihan kunci field haruslah sama antara file primary dan file secondary. Jika terdapat lebih dari satu kunci field yang cocok dangan secondary file, ACL hanya menggunakan satu field saja dari record pertama yang cocok. Jika ada record yang cocok pada secondary file, ACL tidak akan menghasilkan sebuah output record.



File output yang mengandung record ini

Apabila kita memilih pilihan outputnya adalah Join Matched Record maka ACL akan menampilkan record-record yang sama berdasarkan kunci field yang telah ditentukan. Kunci field haruslah sama antara primary dan secondary file. Sehingga dengan cocoknya kunci field maka field-field yang ingin ditampilkan baik yang bersumber dari primary dan secondary dapat dilakukan.

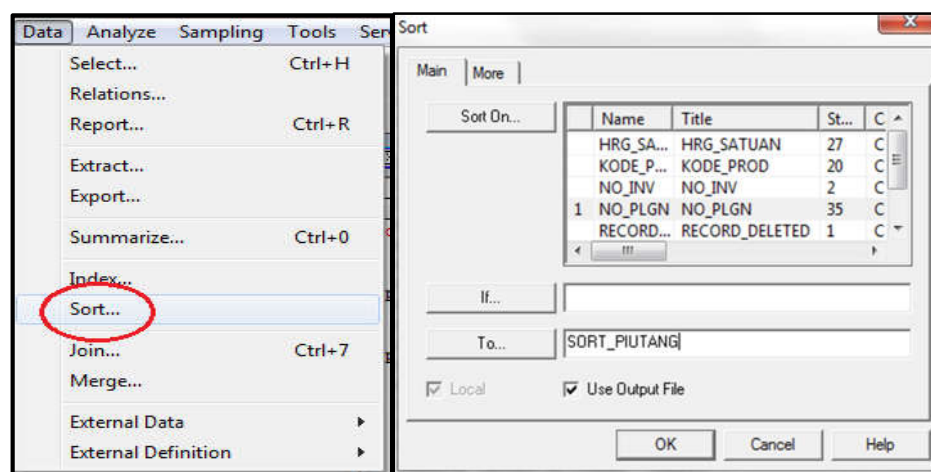
Jika terdapat lebih dari satu kunci field yang cocok dengan secondary file, ACL hanya akan menggunakan satu field saja dari record pertama yang cocok. Jika tidak ada record yang cocok pada secondary file, ACL tidak akan menghasilkan sebuah output record.

### Contoh Kasus :

PT. Labala Bersama akan melakukan pemeriksaan terhadap seluruh data perusahaan. Salah satu aspek yang akan diperiksa adalah data piutang yang mereka miliki. Perusahaan ingin melihat kecocokan antara daftar piutang dengan daftar pelanggan berdasarkan nomor pelanggan. Bagaimana cara yang harus dilakukan perusahaan untuk melihat kecocokkan data apabila perusahaan menggunakan software Audit Command Language (ACL)?

### Langkah – langkah :

1. Lakukan **SORT** pada file PIUTANG.dbf berdasarkan NO\_PLGN dan simpan SORT filenya dengan nama SORT\_PIUTANG.



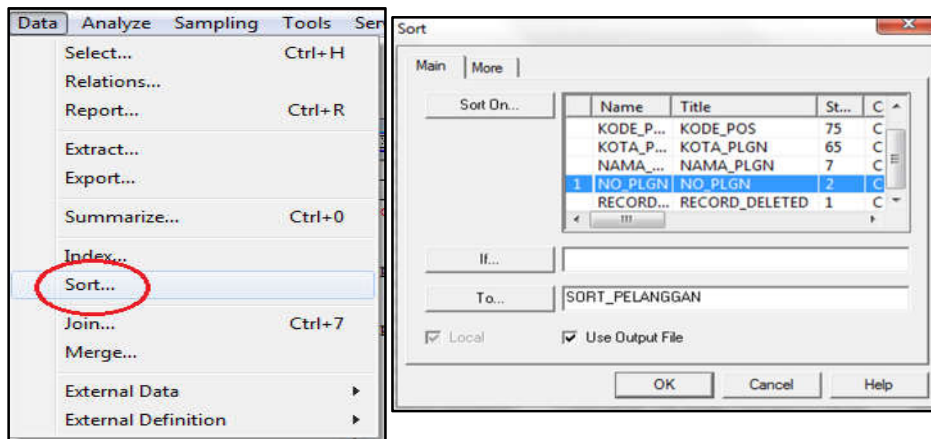


Tampilan View Default nya sebagai berikut :

RE	NO INV	TGL INV	KODE PROD	UNIT PIUT	HRG SATUAN	NO PLGN
1	LA170516	17/05/2016	LTP03	40	15000000	C0001
2	LA090216	19/02/2016	AND01	24	21000000	C0002
3	LA160216	16/02/2016	EXT01	24	4000000	C0002
4	LA140416	14/04/2016	IOS03	7	37000000	C0003
5	LA210316	21/03/2016	WDS10	36	30000000	C0003
6	LA010516	01/05/2016	MPL03	18	10000000	C0004
7	LA010516	01/05/2016	LTP15	10	130000000	C0004
8	LA060216	06/02/2016	MPL05	9	22000000	C0005
9	LA060216	06/02/2016	SLR07	17	200000000	C0005
10	LA040716	04/07/2016	WDS11	12	50000000	C0007
11	LA200316	20/03/2016	SLR06	30	180000000	C0011
12	LA160316	16/03/2016	EXT03	30	3800000	C0012
13	LA120716	12/07/2016	AND08	25	70000000	C0013
14	LA150116	15/01/2016	MPL02	30	12500000	C0013
15	LA150116	15/01/2016	SLR01	8	80000000	C0013
16	LA150116	15/01/2016	DSL03	48	55000000	C0013
17	LA150116	15/01/2016	HDS02	15	1500000	C0013
18	LA090716	09/07/2016	ACD04	10	36000000	C0014
19	LA070916	07/09/2016	SPK03	10	1150000	C0015
20	LA190816	19/08/2016	LCD04	15	22000000	C0016
21	LA101016	10/10/2016	LCD12	12	32000000	C0017
22	LA090316	09/03/2016	MPL07	20	5000000	C0021
23	LA100216	10/02/2016	LTP01	5	80000000	C0022
24	LA020116	02/01/2016	SPK15	30	2700000	C0023
25	LA020116	02/01/2016	BWR02	5	4500000	C0023
26	LA240516	24/05/2016	SPK02	15	3180000	C0023
27	LA060316	06/03/2016	DSL02	50	50000000	C0024
28	LA060316	06/03/2016	LCD03	15	24000000	C0024
29	LA060316	06/03/2016	HDS01	12	2500000	C0024
30	LA260416	26/04/2016	AND04	20	15000000	C0024

*Nb: File yang akan di Join adalah file yang sudah di SORT*

- Lakukan **SORT** pada file PELANGGAN.dbf berdasarkan NO\_PLGN dan simpan SORT filenya dengan nama SORT\_PELANGGAN.



Tampilan View Default nya sebagai berikut :

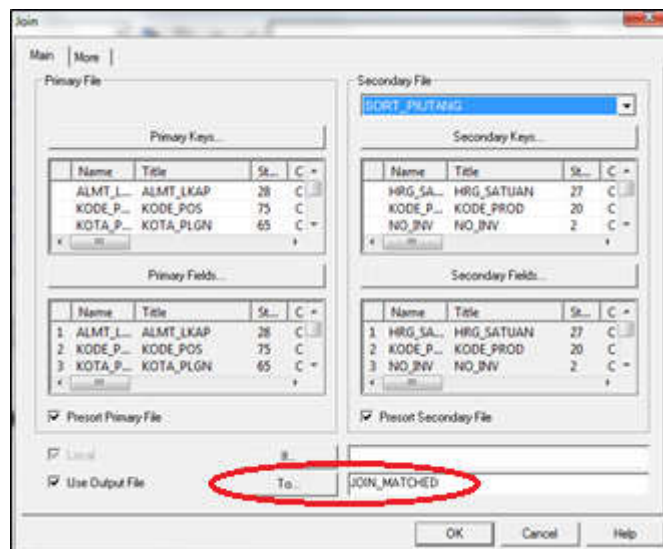
R	NO PL	NAMA_PLGN	ALMT_LKAP	KOTA_PLGN	KODE
1	C0001	Iri Rizka Handayani	Jl. Ku Bukan Jalan Mu No. 12	Jakarta	17145
2	C0002	Avel Rizki	Jl. Menuju Roma No. 23 Blok C	Bekasi	17421
3	C0003	Darul Bawar Maulana	Jl. Jalan Terus Kapan Jadiannya No. 1	Bandung	17422
4	C0004	Frischa Lantia	Jl. Kebenaran No. 1	Bandung	40657
5	C0005	Javerudi Ginting	Jl. Baberian Mas Permai No. 23	Jakarta	17610
6	C0006	Kurniawan Eka Putra	Jl. Pulang Sendiri No. 99	Bekasi	17928
7	C0007	Nurul Azzah	Jl. Jalan Sendiri Terus No. 11	Surabaya	60154
8	C0008	Rachmasari Ocktarina	Jl. Kemangan Masa Laju No. 10	Surabaya	60421
9	C0009	Sri Ningsih	Kp. Dakwah Tetap No. 1	Solo	57131
10	C0010	Udin Saepuloh	Kp. Cibiru No. 25 RT 01 RW 02	Bandung	40614
11	C0011	Mahmudin Abanda	Jl. Gedebage No. 13 Blok J1	Bandung	40234
12	C0012	Muhammad Arsaad	Jl. Penyukun Raya No. 21	Cirebon	45162
13	C0013	Almarhumah	Jl. Sukahati No. 04 blok F3	Bandung	45191
14	C0014	Grononi Aj	Jl. Sukasari No. 17 RT 08 RW 07	Bandung	40151
15	C0015	Amdal Eka Kurnia	Jl. Kenanga 4 blok6 No. 13	Bekasi	17772
16	C0016	Kurnia Meiga	Jl. Palembang blok cc 24 No. 24	Surabaya	60152
17	C0017	Irfan Bachdim	Kemang III Graha No. 179	Bekasi	17424
18	C0018	Syamsul Arif	Jl. Wijaya kusuma 7 No 6	Bekasi	16969
19	C0019	Raphael Maitimo	Jl. Nusa Indah 4 B 3/9 Margahayu jaya	Yogyakarta	55222
20	C0020	Kim Jeffrey Kurniawan	Perumahan The Green View Blok D No 59	Bekasi	17147
21	C0021	Yanto Basna	Jl. Suka Maju No. 345 Blok D2	Bandung	40111
22	C0022	Rizky Pelu	Jl. Palembang No. 98 perumahan 1	Jakarta	17145
23	C0023	Evan Omas Darmono	Jl. Bumi 3 No 23 Blok C3	Blitar	65148
24	C0024	Zulfandi Cole	Jl. Mustika Karang Satra	Solo	57132
25	C0025	Made Wirawan	Jl. LemahAbang Blok DA3	Bekasi	17550
26	C0026	Sarena Williams	Jl. Impian Jaya No. 15	Bekasi	17111
27	C0027	Ayu Rahayu	Kp. Sate Babakan	Yogyakarta	55283
28	C0028	Rina Hertina	Babakan Ito No. 12 RT 01 RW 03	Bogor	16980
29	C0029	Fitria Ramadhani	Disney St. Coca cola NSW	Depok	12345
30	C0030	Dyna Stefany	Jl. Mercurius No. 17 Blok H7	Jakarta	17122

*Nb: File yang akan di Join adalah file yang sudah di SORT*

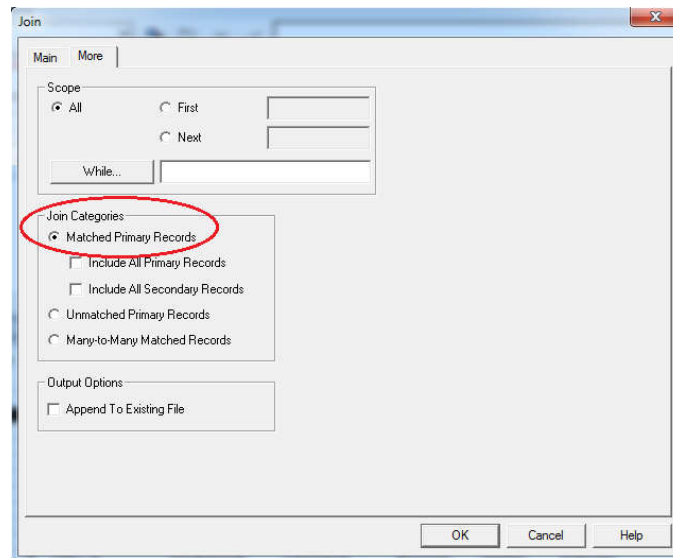
3. Aktif pada primary file, yang menjadi primary file adalah file yang memiliki jumlah record terbanyak. Untuk melihat jumlah record pertama-tama kita harus aktif pada file tersebut, lalu jumlah record dapat dilihat di:



4. Pilih menu **Data** lalu submenu **JOIN**.
5. Pada **Main** pilih SORT\_PIUTANG sebagai secondary filenya.
6. Pada **Primary Keys** dan **Secondary Keys** pilih NO\_PLGN.
7. Klik checkbox **Presort** untuk meyakinkan Primary Key sudah terurut.
8. Untuk **Primary Fields** dan **Secondary Fields** listboxnya pilih semua fields yaitu dengan mengklik **Add All**.
9. Kemudian klik check box **Presort Secondary File** untuk meyakinkan Secondary Fieldnya sudah terurut.
10. Pada text box To ketik **JOIN\_MATCHED**.



11. Klik tab **More** pada Scope klik **All** dan pada Join Categories **Matched Primary Records**.



12. Klik **Ok**, maka tampilan Command Log akan muncul sebagai berikut :

```
Presorting data
Presorting data
41 records produced
29 records bypassed
Extraction to file D:\labala\Data ACL 2016\JOIN_MATCHED.FIL is complete
Opening file "JOIN_MATCHED"
@ OPEN "JOIN_MATCHED"
13 fields activated
Opening file name JOIN_MATCHED.FIL as supplied in the format.
```

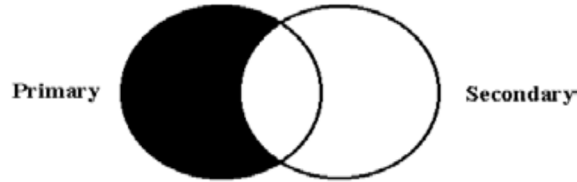
13. Kemudian akan muncul tampilan di Jendela View seperti gambar dibawah ini :

ALMT LKAP	KODE	KOTA	PLGN	NAMA PLGN	NO PLGN	R	HRG	SATUAN	KODE PROD	NO INV	NO PLG	R	TGL	INV
Jl. Ku Bukan Jalan Mu No. 12	17145	Jakarta		Ari Rizkia Handayani	C0001		15000000	LTP03		L4170516	C0001		17/05/2016	4
Jl. Menuju Roma No. 23 Blok C	17421	Bekasi		Axel Rizki	C0002		21000000	AND01		L4090216	C0002		19/02/2016	2
Jl. Jalan Terus Kapan Jadianya No. 1	17422	Bandung		Danul Bawar Maulana	C0003		37000000	IOS03		L4140416	C0003		14/04/2016	7
Jl. Kebenaran No. 1	40657	Bandung		Frischa Lamria	C0004		10000000	MPL03		L4010516	C0004		01/05/2016	1
Jl. Babelan Mas Permai No. 23	17610	Jakarta		Jevrendhi Ginting	C0005		22000000	MPL05		L4060216	C0005		06/02/2016	6
Jl. Jalan Sendiri Terus No. 11	60154	Surabaya		Nurul Azizah	C0007		50000000	WDS11		L4040716	C0007		04/07/2016	1
Jl. Gedebage No. 13 Blok J1	40294	Bandung		Mahmudin Abanda	C0011		18000000	SLR06		L4200316	C0011		20/03/2016	3
Jl. Penyukon Raya No. 21	45162	Cirebon		Muhammad Arsad	C0012		38000000	EXT03		L4160316	C0012		16/03/2016	3
Jl. Sukahati No. 04 blok F3	45191	Bandung		Almarhumah	C0013		70000000	AND08		L4120716	C0013		12/07/2016	2
Jl. Sukasari No. 17 RT 06 RW 07	40151	Bandung		Giononi Aji	C0014		36000000	ACD04		L4080716	C0014		08/07/2016	1
Jl. Kemanga 4 blok G No. 13	17772	Bekasi		Amdia Eka Kurnia	C0015		11500000	SPK03		L4070816	C0015		07/08/2016	1
Jl. Palembang Raya blok cc 24 No. 24	60152	Surabaya		Kurnia Meiga	C0016		22000000	LCD04		L4190816	C0016		19/08/2016	1
Kemang III Graha No. 179	17424	Bekasi		Irfan Bachdim	C0017		32000000	LCD12		L4101016	C0017		10/10/2016	1
Jl. Suka Maju No. 345 Blok D2	40111	Bandung		Yanto Basna	C0021		50000000	MPL07		L4090316	C0021		09/03/2016	2
Jl. Palembang No. 98 perumahan 1	17145	Jakarta		Rizky Pellu	C0022		80000000	LTP01		L4100216	C0022		10/02/2016	3
Jl. Bumi 3 No 23 blok C3	65148	Miaing		Evan Dimas Darmono	C0023		27000000	SPK15		L4020116	C0023		02/01/2016	2
Jl. Mustika Karang Satna	57132	Solo		Zulfandi Cole	C0024		50000000	DSL02		L4060316	C0024		06/03/2016	2
Jl. Lemahabang Blok DA3	17550	Bekasi		Made Wirawan	C0025		60000000	AND07		L4010216	C0025		01/02/2016	1
Jl. Impian Jaya No. 15	17111	Bekasi		Serena Williams	C0026		60000000	WDS12		L4130116	C0026		13/01/2016	1
Kp. Sate Babakan	55283	Yogyakarta		Ayu Rahayu	C0027		80000000	MPL01		L4170216	C0027		17/02/2016	1
Jl. Merkurius No. 17 Blok H7	17827	Jakarta		Dyna Stefany	C0030		38000000	BWR03		L4110616	C0030		11/06/2016	4
Jl. Melati 5 No. 77 blok B8	16672	Bogor		Anggo Julian	C0031		12000000	LED11		L4160716	C0031		16/07/2016	4
Jl. Sukamandi Raya No. 7 Blok G12	43122	Sukabumi		Diona Salva	C0032		87500000	LTP12		L4180416	C0032		18/04/2016	6
Jl. Sukses Selalu No 79 Blok K8	18970	Depok		Rizka Novita	C0033		11000000	LTP25		L4190516	C0033		19/05/2016	2
Kp. Cibirusah Raya No 178 Blok N7	10987	Depok		Anggi Novita	C0034		30000000	IOS04		L4210416	C0034		21/04/2016	1
Jl. SukaCita No 7 blok G05	40112	Bandung		Rina Nose	C0035		70000000	EXT04		L4070316	C0035		07/03/2016	1
Jl. Mega Puspita No 8 Blok U7	17893	Jakarta		Anggun Sudrajat	C0036		35000000	IOS02		L4040916	C0036		04/09/2016	2
Kp. SukaMaju No. 67	40112	Bandung		Raisa Sahaja	C0037		13000000	IOS13		L4060116	C0037		06/01/2016	1
Jl. Cinta Damai No. 10	10927	Depok		Isyana Sarasvati	C0038		20000000	DSL01		L4230116	C0038		23/01/2016	2
Jl. Venus Raya No. 19 Blok B6	40221	Bandung		Citra Scholastika	C0039		14000000	SLR02		L4220416	C0039		22/04/2016	2
Kp. Cibirusah No. 34 Blok C7	16738	Bekasi		Nagita Slavina	C0040		22000000	LCD01		L4010916	C0040		01/09/2016	2
Jl. In aja dulu No. 76 Blok H5	40111	Bandung		Prisia Nasution	C0041		30000000	ACD01		L4020916	C0041		02/09/2016	2
Jl. Suka suka No. 7 Blok I5	17639	Depok		Anggi Putri	C0042		42000000	BWR01		L4110916	C0042		11/09/2016	2
Jl. Mandiri Raya No. 76 Blok D4	13773	Jakarta		Lydiawati Ekasari	C0043		35000000	ACD03		L4111016	C0043		11/10/2016	1

**Analisis** : Berdasarkan hasil dari kecocokan antara daftar piutang dengan daftar pelanggan yang dilihat dari nomor pelanggannya, terdapat 41 records produced yang artinya ada 41 records yang memiliki kecocokan dan 29 records bypassed yang artinya ada 29 records yang memiliki ketidakcocokan.

## B. Join Unmatched Record

Apabila kita memilih pilihan outputnya adalah Join Unmatched Record maka ACL hanya akan menampilkan record-record dari primary file yang tidak sama dengan secondary file.



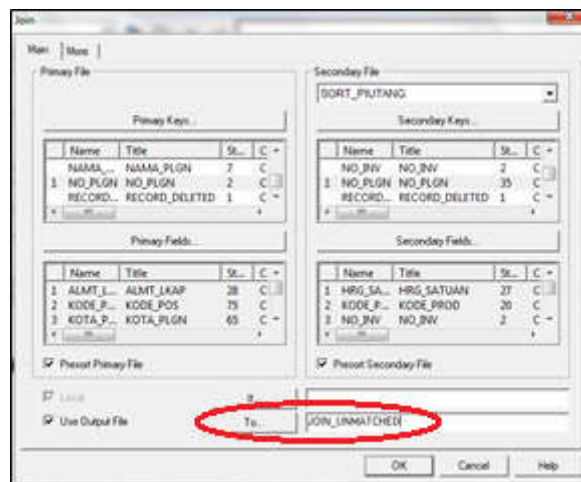
File output yang mengandung record ini

### Contoh Kasus :

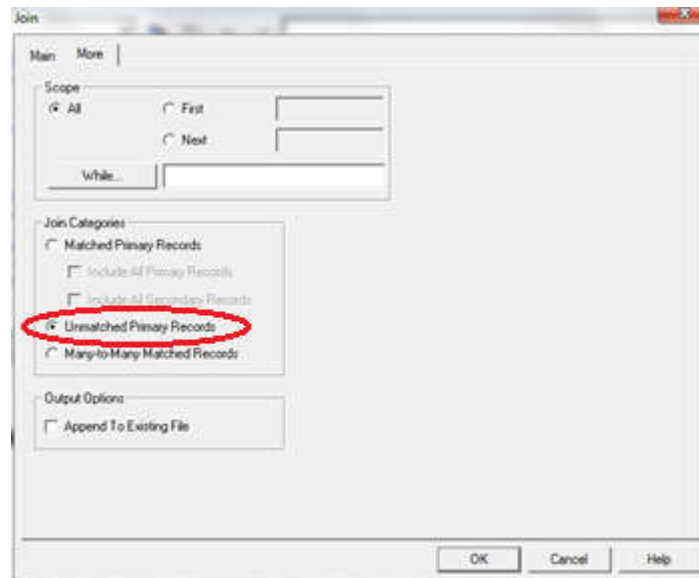
Apabila PT. Labala Bersama ingin melihat ketidakcocokkan antara daftar piutang dengan daftar pelanggan berdasarkan nomor pelanggan, maka bagaimana cara yang harus dilakukan perusahaan untuk melihat ketidakcocokkan tersebut apabila perusahaan menggunakan software Audit Command Language (ACL)?

### Langkah – langkah :

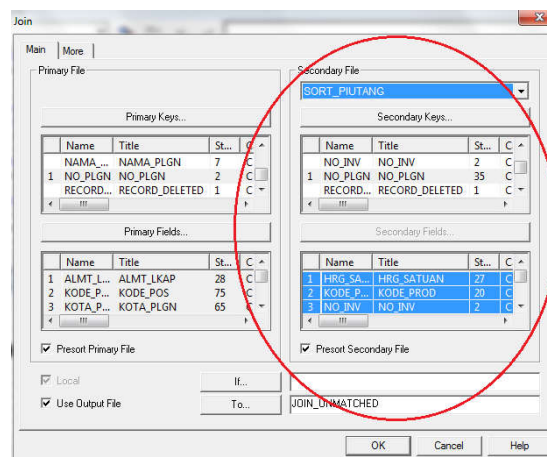
1. Pertama-tama, ikuti Langkah 1 s/d 9 sama sepertipada JOIN\_MATCHED.
2. Pada text box To ketik JOIN\_UNMATCHED.



3. Klik tab More, pada Scope klik All dan pada Join Categories klik **Unmatched Primary Records**.



4. Kemudian kembali ke tab Main, maka Secondary Fields akan otomatis ter-highlight.



5. Klik Ok, maka tampilan Command Log akan muncul sebagai berikut :

```
Presorting data
Presorting data
29 records produced
41 records bypassed
Extraction to file D:\labala\Data ACL 2016\JOIN_UNMATCHED.FIL is complete
Opening file "JOIN_UNMATCHED"
@ OPEN "JOIN_UNMATCHED"
6 fields activated
Opening file name JOIN_UNMATCHED.FIL as supplied in the format.
```



6. Lalu tampilan pada Default\_View sebagai berikut :

	ALMT_LKAP	KODE	KOTA_PLGN	NAMA_PLGN	NO_PLGN	R
1	Jl. Pulang Sendiri No. 99	17928	Bekasi	Kurniawan Eka Putra	C0006	
2	Jl. Kenangan Masa Lalu No. 10	60421	Surabaya	Rachmasari Ocktarina	C0008	
3	Kp. Dakwah Tetap No. 1	57131	Solo	Sri Ningsih	C0009	
4	Kp. Cibiru No. 25 RT 01 RW 02	40614	Bandung	Udin Saepuloh	C0010	
5	Jl. Wijaya kusuma 7 No 6	16969	Bekasi	Syamsul Arif	C0018	
6	Jl. Nusa Indah 4 B 3/9 Margahayu jaya	55222	Yogyakarta	Raphael Maitimo	C0019	
7	Perumahan The Green View Blok D No 59	17147	Bekasi	Kim Jeffrey Kurniawan	C0020	
8	Babakan lio No. 12 RT 01 RW 03	16680	Bogor	Rina Herlina	C0028	
9	Disney St. Coca cola NSW	12345	Depok	Fitria Ramadhan	C0029	
10	Jl. CitaCita No. 45 Blok G6	45114	Cirebon	Maesaroh	C0046	
11	Jl. Kepiting Raya No 32 Blok D/3	18927	Bekasi	Ria Recheese	C0047	
12	Jl. Kepiting Raya No 31 Blok D/3	37836	Bekasi	Lae Luaha	C0048	
13	Jl. Pulo sari No. 78 Blok B3	26637	Padang	Maria Sharavopa	C0049	
14	Jl. Mega Mendung No. 17 Blok S3	50234	Semarang	Alexandre Graham	C0050	
15	Jl. Putus Cinta No. 11 Blok j7	50230	Semarang	Danan Puspito	C0051	
16	Jl. Kepompong Raya No 52 Blok g4	28172	Medan	Zaenal Abidin	C0052	
17	Kp. Galau Terus No. 8	30162	Palembang	Abdullah Abdul Abdal	C0055	
18	Jl. Pantang Mundur No. 111 Blok GF 2	18376	Tangerang	Steve Jobs	C0057	
19	Jl. Suka Terus No. 123 Blok G6	19073	Tangerang	Steve Ji	C0058	
20	Jl. Pantang Mundur No. 345 Blok H8	12781	Bogor	Hendro Kartiko	C0060	
21	Jl. Pemah muda No. 45 Blok G5	28982	Medan	Nemo	C0061	
22	Jl. Pemuda No. 27 Blok G22	40112	Bandung	Dori	C0062	
23	Jl. Permai Indah No. 10 Blok J6	29763	Jakarta	Chicarito	C0063	
24	Jl. Permai Indah No. 15 Blok K1	19902	Jakarta	Mahmuda	C0064	
25	Jl. Wara wiri No. 121 Blok B8	20229	Medan	Xavier Dinan	C0065	
26	Jl. penjuken Raya No. 35 Blok DG7	45118	Cirebon	Muhammad Yahya	C0066	
27	Jl. Mars No. 7 Blok H6	45117	Cirebon	Alfatul Hasbullah	C0067	
28	Jl. Saturnus Raya No. 13 Blok G4	12763	Jakarta	Angel Di Maria	C0069	
29	Jl. Buenos Aires No. 13 Blok A1	11111	Jakarta	Lionel Andres Messi	C0070	

**Analisis** : Berdasarkan hasil dari ketidakcocokan antara daftar piutang dengan daftar gaji pelanggan yang dilihat dari nomor pelanggannya, terdapat 29 records produced yang artinya ada 29 records yang memiliki ketidak-cocokan dan 41 records bypassed yang artinya ada 41 records yang memiliki kecocokan.

### Kasus 1

PT. AXLE akan melakukan pemeriksaan terhadap seluruh data perusahaan. Salah satu aspek yang akan diperiksa adalah data hutang yang mereka miliki. Perusahaan ingin melihat kecocokan antara daftar hutang dengan daftar supplier berdasarkan nomor supplier. Bagaimana cara yang harus dilakukan perusahaan untuk melihat kecocokan apabila perusahaan menggunakan software Audit Command Language (ACL)?

### Kasus 2 :

PT. DOREMI akan melakukan pemeriksaan terhadap data gaji pegawai yang mereka miliki. PT. DOREMI ingin melihat ketidakcocokan antara daftar gaji dengan daftar pegawai berdasarkan nomor pegawai, maka bagaimana cara yang harus dilakukan perusahaan untuk melihat ketidakcocokan tersebut apabila perusahaan menggunakan software Audit Command Language (ACL)?





## MERGE

Perintah merge digunakan untuk menggabungkan atau mengkombinasikan dua file yang memiliki struktur record yang sama menjadi sebuah file baru. Sebelum melakukan perintah **merge**, kita harus melakukan perintah **extract**. Perintah extract yaitu untuk mengatur record dalam input file secara ascending.

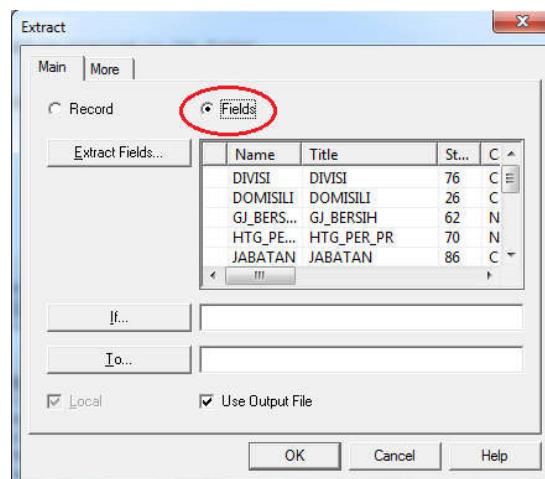
### Contoh Kasus :

PT. Labala Bersama akan melakukan pemeriksaan terhadap seluruh data perusahaan, salah satunya yaitu Data Pegawai. Seluruh karyawan telah melunasi hutangnya kecuali Programmer yang memiliki hutang per periode lebih dari samadengan Rp. 300.000. Oleh karena itu Perusahaan menugaskan seorang auditor untuk mengaudit Data Pegawai dan diminta untuk membuat rincian jumlah karyawan yang bekerja sebagai Programmer yang berjenis kelamin laki-laki dan perempuan, serta yang memiliki hutang per periode lebih dari samadengan Rp. 300.000. Bagaimanakah auditor melakukan pemeriksaan apabila perusahaan memerlukan data Nama Pegawai, Jenis Kelamin, Jabatan, Hutang Per Periode dan Nomor Telepon pegawai menggunakan Aplikasi Audit Command Language (ACL)?

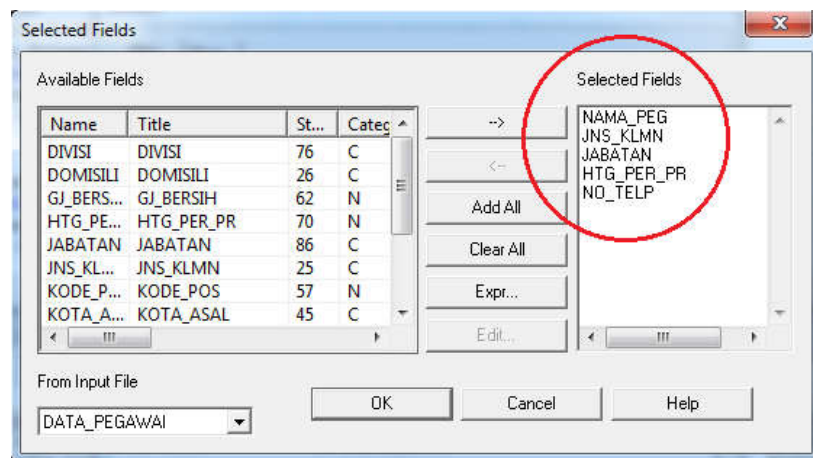
### Langkah-langkah :

#### A. Menciptakan Extract pertama:

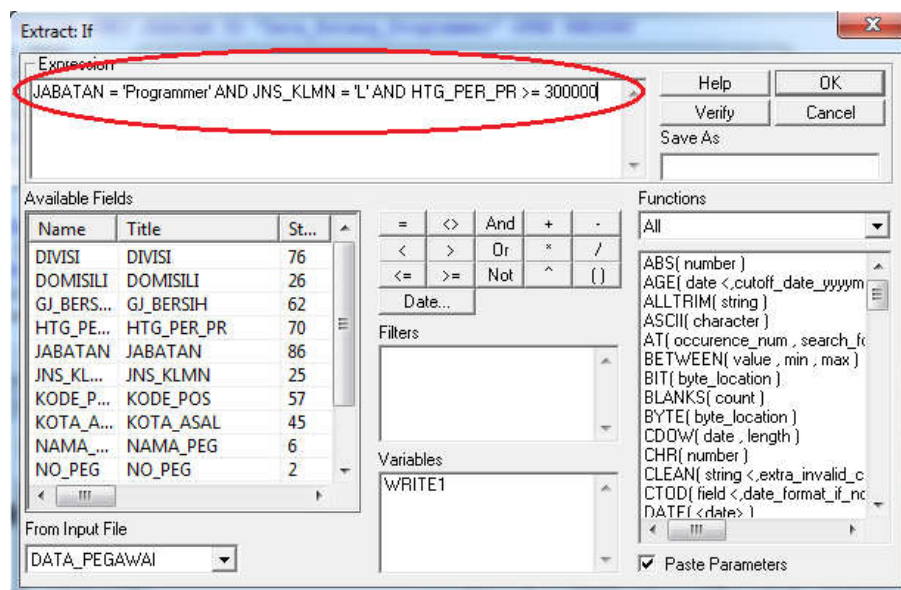
1. Aktif pada file DATA\_PEGAWAI.
2. Pilih perintah Extract pada menubar Data.
3. Pada Main, Klik radio button Fields.



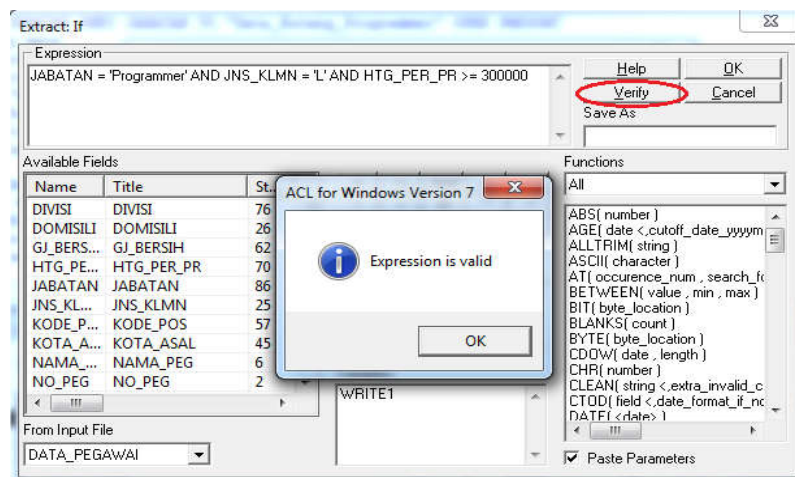
4. Klik Extract Fields, pilih fields yang dibutuhkan (Nama Pegawai, JenisKelamin, Jabatan, Hutang Per Periode dan Nomor Telepon) lalu klik OK.



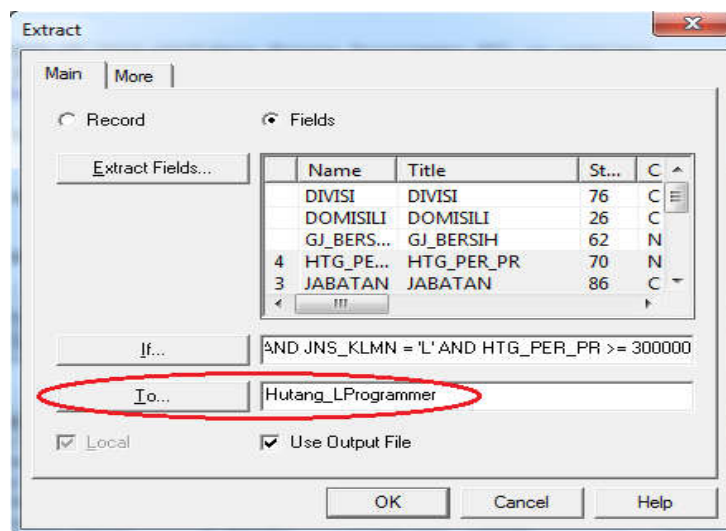
5. Klik If, isi kolom Expression dengan rumus: **JABATAN = 'Programmer' AND JNS\_KLMN = 'L' AND HTG\_PER\_PR >= 300.000**.



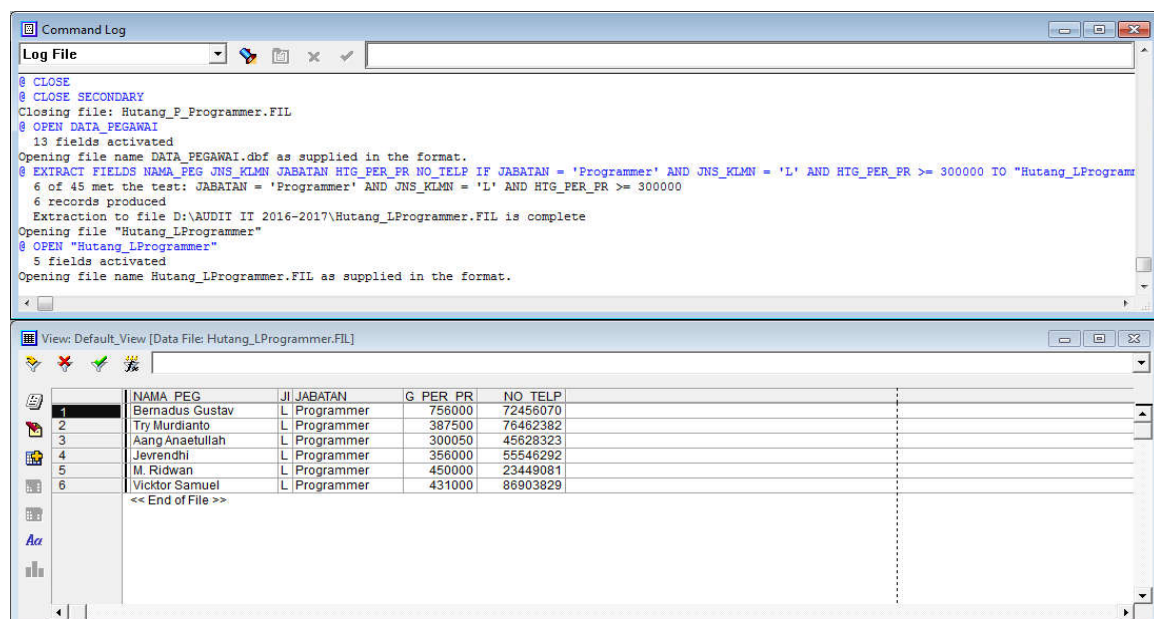
6. Jika sudah, Klik Verify untuk memastikan apakah rumus yang diinput sudah benar/valid, lalu klik Ok.



7. Pada text box To ketik **Hutang\_LProgrammer**, klik Ok.



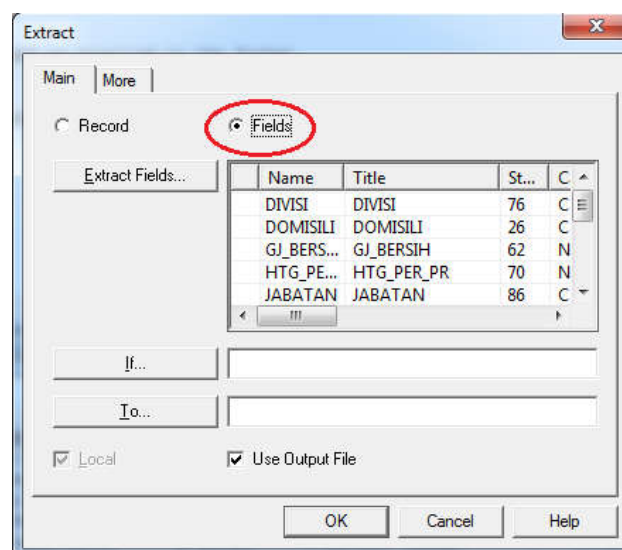
8. Maka hasil akan muncul pada Command Log dan Default\_View.



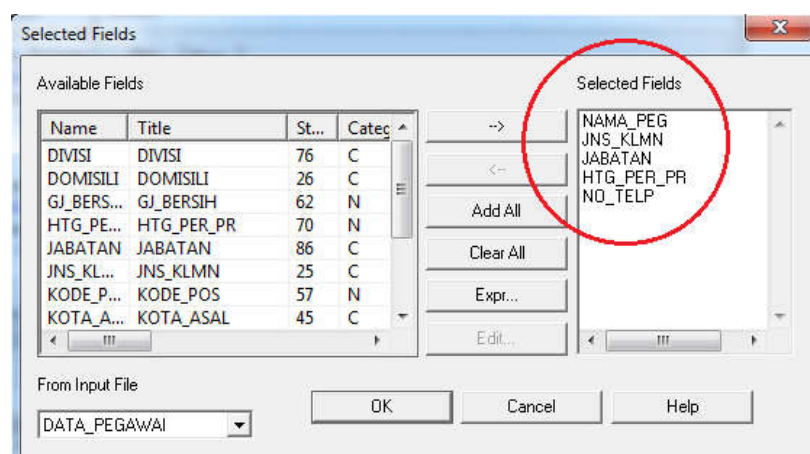
**Analisis** : Dari data diatas diketahui bahwa extract 1 yaitu Data Hutang\_LProgrammer berhasil dibuat dengan 6 file yang aktif dari 45 record yang memiliki jabatan sebagai programmer, berjenis kelamin laki-laki dan memiliki hutang per periode lebih besar sama dengan Rp 300.000.

### B. Menciptakan Extract kedua:

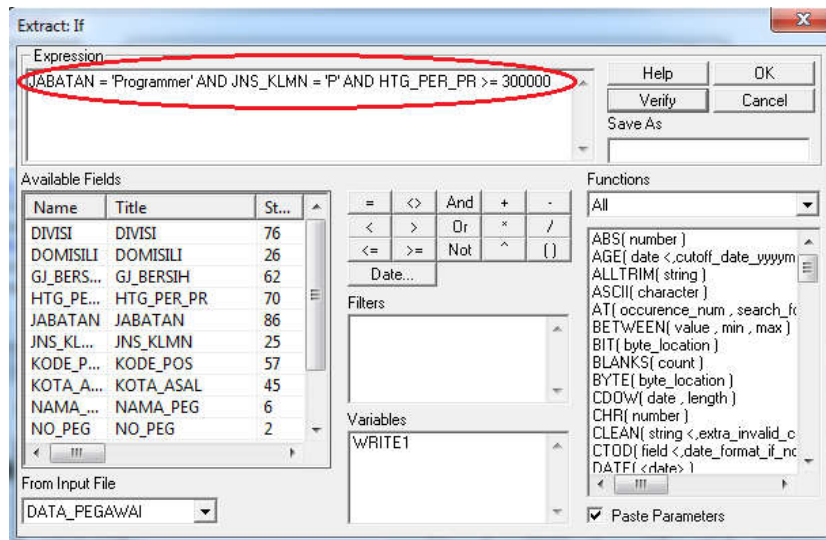
1. Aktif pada file DATA\_PEGAWAI.
2. Pilih perintah Extract pada menubar Data.
3. Pada Main, Klik radio button Fields.



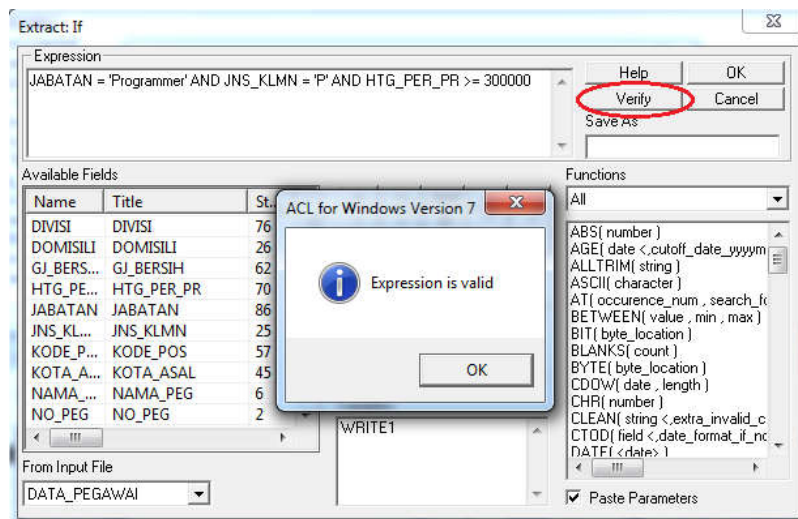
4. Klik Extract Fields, pilih fields yang dibutuhkan (Nama Pegawai, Jenis Kelamin, Jabatan, Hutang Per Periode dan Nomor Telepon) lalu klik OK.



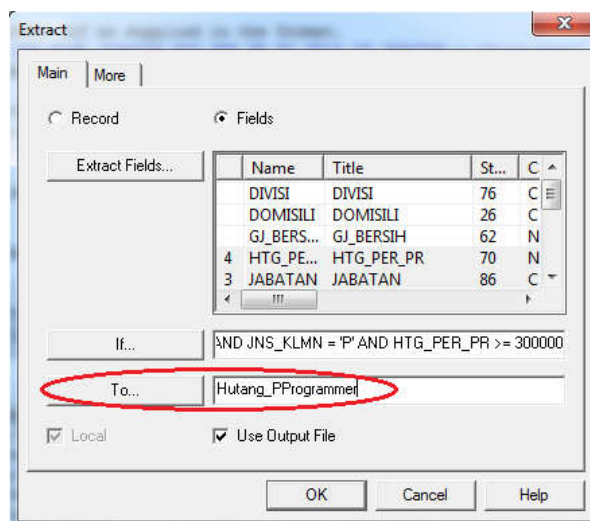
5. Klik If, isi kolom Expression dengan rumus: **JABATAN = 'Programmer' AND JNS\_KLMN = 'P' AND HTG\_PER\_PR >= 300.000.**



6. Jika sudah klik Verify untuk memastikan apakah rumus yang diinput sudah benar/valid lalu klik Ok.

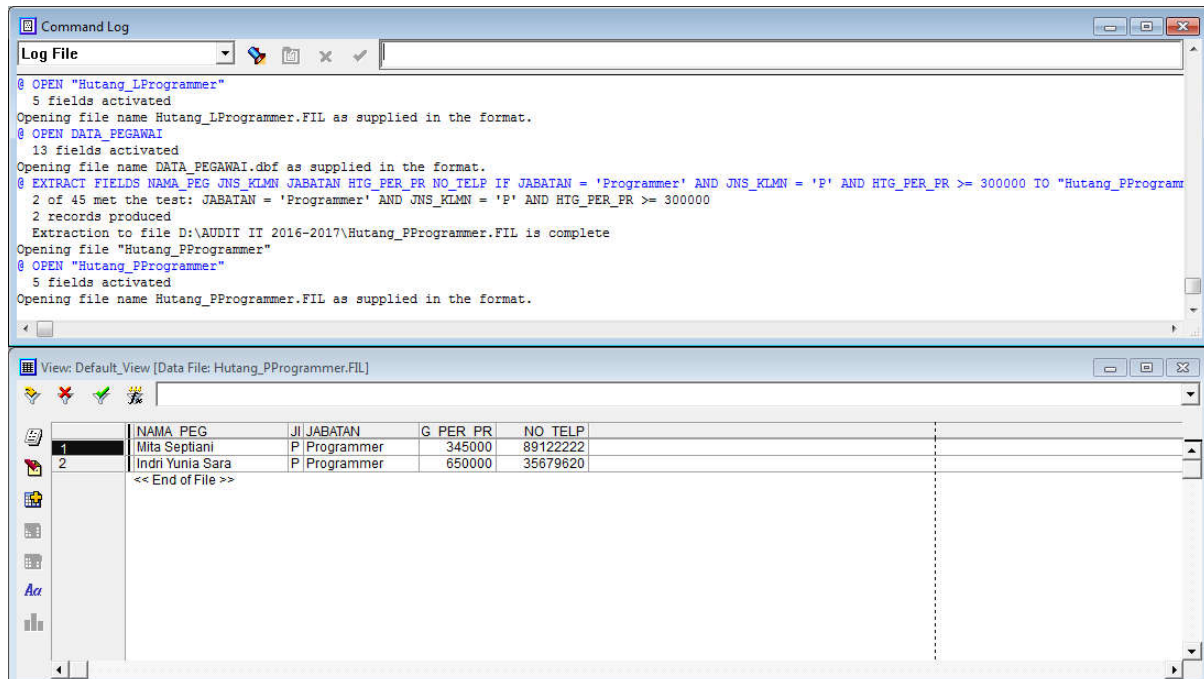


7. Pada text box To... ketik **Hutang\_PProgrammer**, klik Ok.





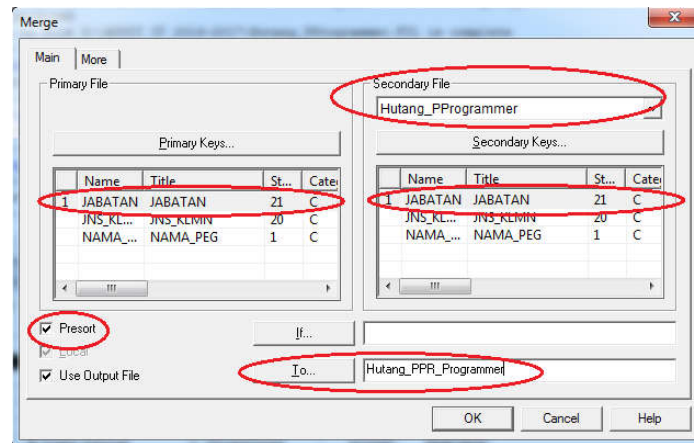
8. Maka Hasilnya akan muncul pada Command Log dan Default\_View.



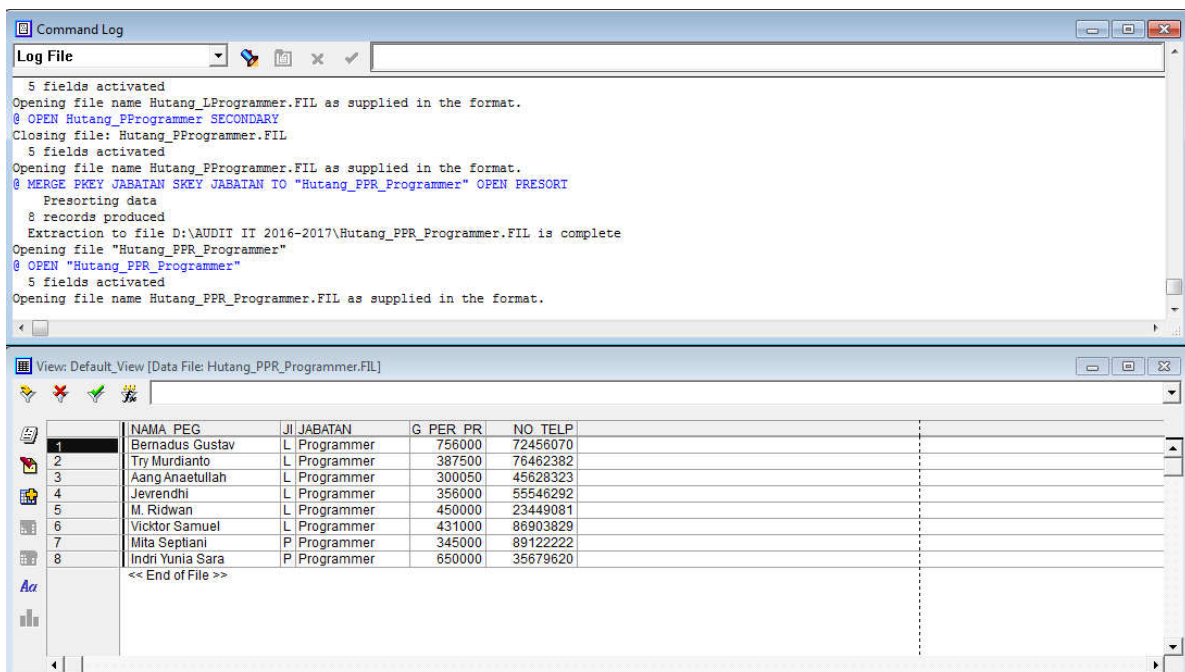
**Analisis** : Dari data diatas diketahui bahwa extract 2 yaitu Data Hutang\_PProgrammer berhasil dibuat dengan 2 file yang aktif dari 45 record yang memiliki jabatan sebagai programmer, berjenis kelamin perempuan dan memiliki hutang per periode lebih besar sama dengan Rp 300.000.

### C. Menggabungkan Extract 1 dan Extract 2

1. Aktif pada file Extract 1 (**Hutang\_LProgrammer**).
2. Klik menu **Data** lalu klik **Merge**.
3. Pada dropdown **Secondary File** pilih **Hutang\_Pprogrammer**.
4. Klik **Jabatan** pada list box **Primary Keys** dan **Secondary Keys**.
5. Checklist Presort pada checkbox.
6. Pada text box **To** ketik **Hutang\_PPR\_Programmer**, klik Ok.



7. Maka hasilnya akan muncul pada Command Log dan Default\_View.



**Analisis** : Dari hasil pada Command Log dan Default\_View diketahui bahwa dari 45 karyawan di perusahaan, terdapat 8 karyawan dengan jabatan sebagai Programmer dan memiliki Hutang Per Periode lebih dari sama dengan Rp.300.000, dimana 2 orang berjenis kelamin perempuan dan 6 orang berjenis kelamin laki-laki.



**Kasus 1 :**

Pada akhir periode PT. LABALA menugaskan kepada auditornya untuk mengetahui data hutang yang ada diperusahaan yang data tersebut berisi unit hutang yang berjumlah kurang dari sama dengan 10 dengan harga satuan lebih dari sama dengan Rp 500.000 (hutang 1) dan unit hutang yang berjumlah lebih dari sama dengan 10 dengan harga satuan lebih dari sama dengan Rp 500.000. Perusahaan membutuhkan Nomor Invoice, Unit Hutang dan Harga Satuan. Buatlah dalam sebuah file baru dengan nama Data Hutang dengan menambahkan kolom jumlah hutang. Kemudian tampilkan grafik data berdasarkan jumlah hutang lebih dari sama dengan Rp 500.000 dengan interval 10.

**Kasus 2 :**

Auditor pada PT. Abracadabra diminta untuk menganalisis pajak karyawan. Perusahaan ingin mengetahui berapa jumlah karyawan yang besar pajaknya tidak lebih dari Rp. 200.000. Hal ini untuk mempermudah perusahaan dalam pengurusan data karyawan dalam file Gaji karyawan. Auditor menganalisis Gaji Bersih karyawan yang kurang dari Rp. 3.500.000 dan Pajak kurang dari Rp. 200.000 (Data 1). Kemudian, menganalisis Gaji Bersih karyawan yang lebih besar dari Rp. 3.500.000 dan Pajak kurang dari Rp. 200.000 (Data2). Kemudian menggabungkan dan menyimpan 2 file yang telah di extract dengan nama RekapDataPajak.





## FILTER

Filter adalah sejenis ekspresi yang digunakan untuk mengidentifikasi record sesuai dengan kriteria yang diinginkan. Kriteria didefinisikan dengan ekspresi yang bertipe logika. Filter yang dibuat dapat bersifat sementara atau disimpan pada input file definisi sebagai computed field. Pada jendela ekspresi akan ditampilkan hasil filter yang telah dibuat. Ada dua buah jenis filter, yaitu :

### 1. Local Filter

Sebuah ekspresi yang digunakan sebagai filter harus mencakup sebuah kondisi logika, sehingga ACL dapat menghitung ekspresi tersebut benar atau salah. Sebuah local filter hanya digunakan pada perintah tunggal yang dilaksanakan pada satu kejadian saja. Suatu local filter dapat dikatakan sebagai satu contoh logical expression. Agar sebuah expression dapat digunakan sebagai suatu filter, maka harus mencakup suatu kondisi logika, sehingga ACL dapat mengevaluasi setiap record berdasarkan expression tersebut. Local filter digunakan ketika sebuah perintah tunggal dijalankan satu kali saja.

### 2. Global Filter

Yang digunakan untuk semua view pada input file definisi selama kita melaksanakan perintah dari sebuah menu atau tombol. Sebuah filter tetap pada tempatnya sampai kita menutupnya. Ketika anda menggunakan sebuah global filter , semua pemrosesan dalam input file, kecuali perintah yang anda masukkan melalui command log, akan terpengaruh. Kondisi yang akan digunakan oleh global filter tersebut akan tetap berpengaruh sampai anda mematikan filter tersebut, menggunakan filter lain, atau menggunakan input file definition yang berbeda. Anda dapat membuat dan menyimpan sejumlah filter yang berbeda, dan kemudian menggunakan salah satu diantaranya yang sesuai dengan kebutuhan.

### Contoh Kasus :

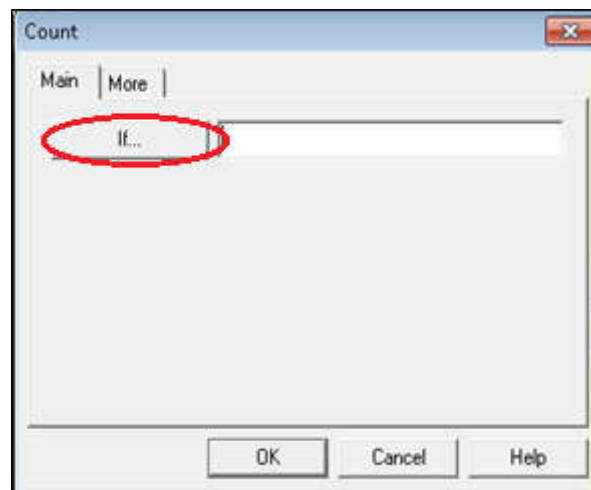
PT. Labala Bersama saat ini sudah menggunakan Audit Command Language sebagai aplikasi yang membantu perusahaan untuk mempermudah pekerjaan dalam melaksanakan pemeriksaan audit. Pada akhir periode ini, perusahaan akan menerima pelunasan piutang dari beberapa supplier yang salah satunya adalah GMT7 CAR. Bagian Akuntansi di PT. Labala Bersama diminta oleh pimpinan perusahaan untuk memeriksa dokumen hutang perusahaan berdasarkan nama supplier GMT7 CAR dengan nomor supplier SP1044. Berikut merupakan

langkah – langkah yang harus dilakukan PT.LABALA dalam melakukan identifikasi menggunakan software ACL.

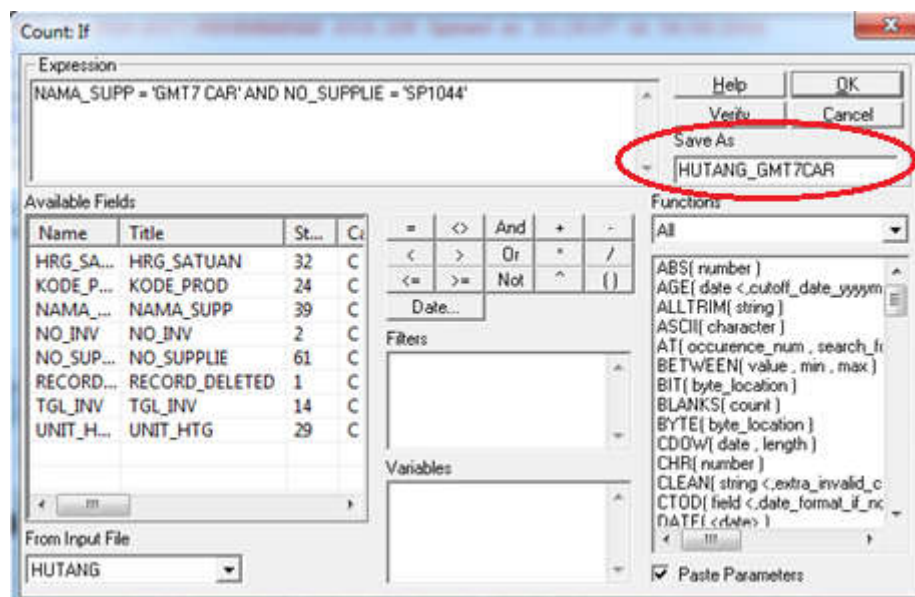
### Langkah-langkah:

#### FILTER LOKAL

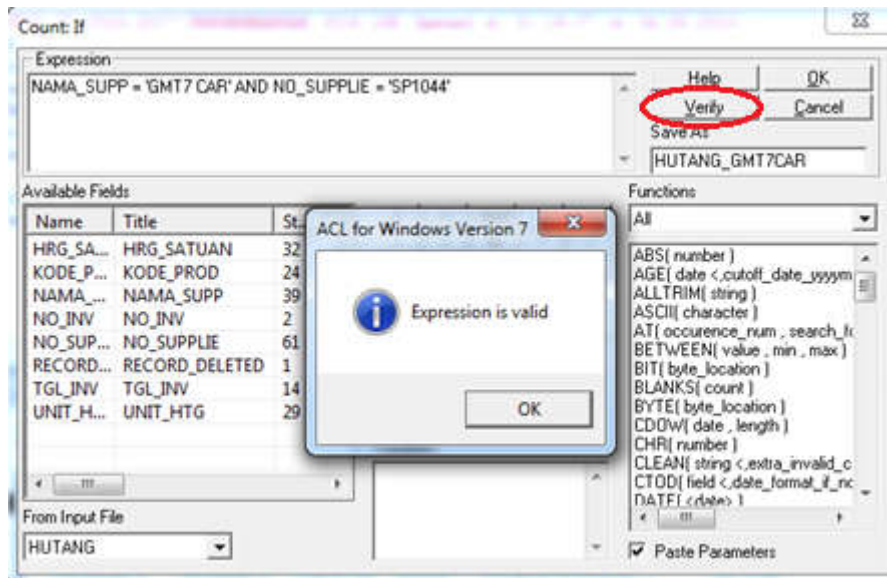
1. Aktifkan data file **HUTANG.DBF**.
2. Pilih menu **Analyze** lalu pilih **Count**.



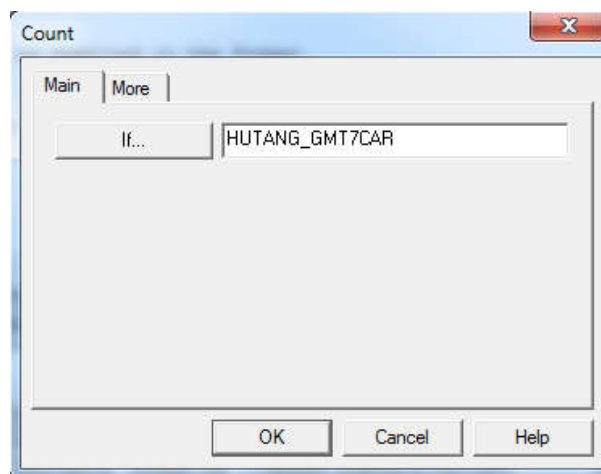
3. Klik **If** untuk membuka expression builder.
4. Lalu buat ekspresi **NAMA\_SUPP = 'GMT7 CAR' AND NO\_SUPPLIE = 'SP1044'**.
5. Dalam kotak teks **Save As**, ketik **HUTANG\_GMT7CAR** untuk menamai filter tersebut.



6. Setelah itu klik **Verify** untuk memastikan sudah sesuaikah ekspresi tersebut.



7. Klik **OK** untuk menutup expression builder dan kembali ke kotak dialog Count. Nama filter expression sekarang sekarang muncul dalam kotak teks **IF**.




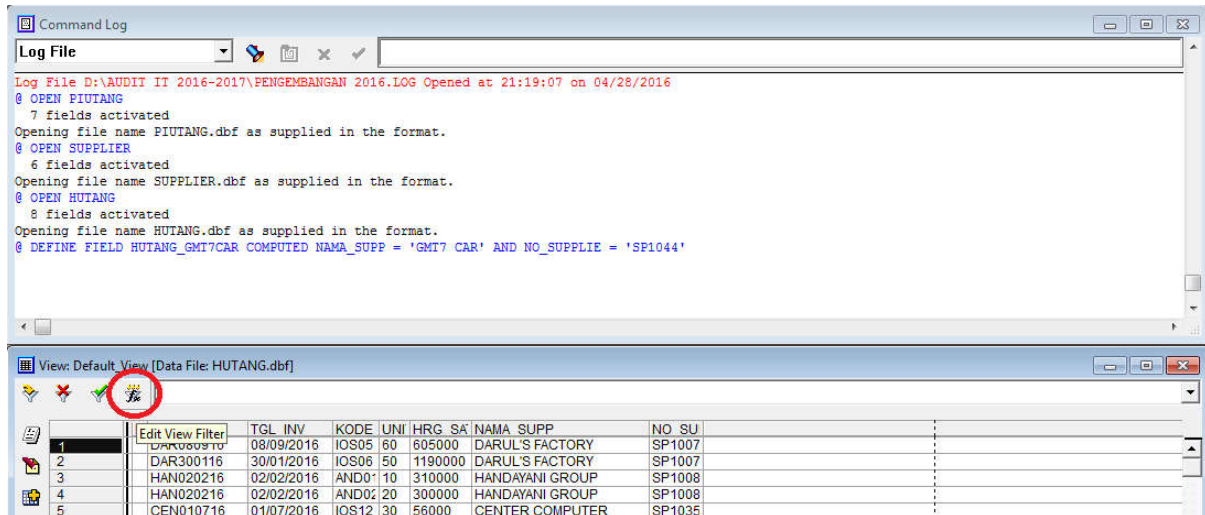
8. Klik **OK**. Dan akan keluar hasilnya pada command log sebagai berikut :

```
@ OPEN HUTANG
9 fields activated
Opening file name HUTANG.dbf as supplied in the format.
@ OPEN HUTANG
9 fields activated
Opening file name HUTANG.dbf as supplied in the format.
@ DEFINE FIELD HUTANG_GMT7CAR COMPUTED NAMA_SUPP = 'GMT7 CAR' AND NO_SUPPLIE = 'SP1044'
@ COUNT IF HUTANG_GMT7CAR
6 of 100 met the test: HUTANG_GMT7CAR
```

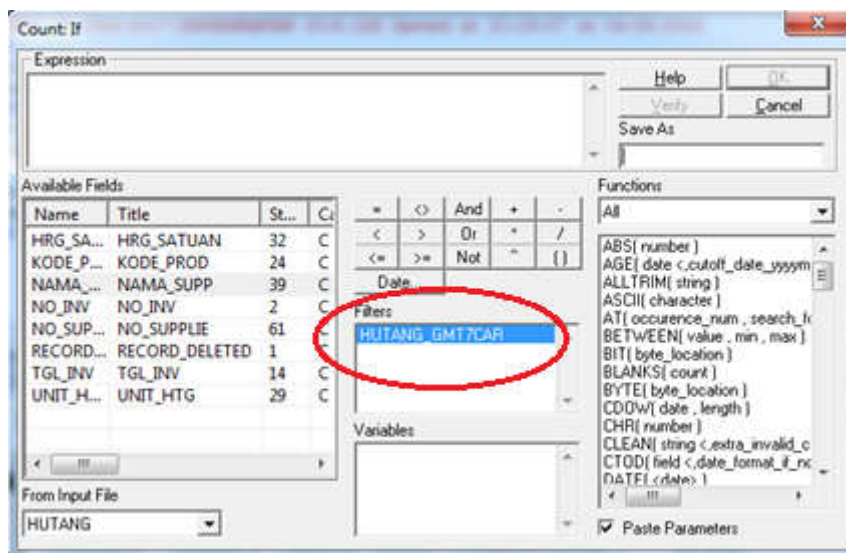
**Analisis** : Dari hasil pada Command Log diketahui bahwa dari 100 record HUTANG, terdapat 6 record HUTANG yang berdasarkan Nama Supplier GMT7 CAR dengan Kode Supplier 1044.

## FILTER GLOBAL

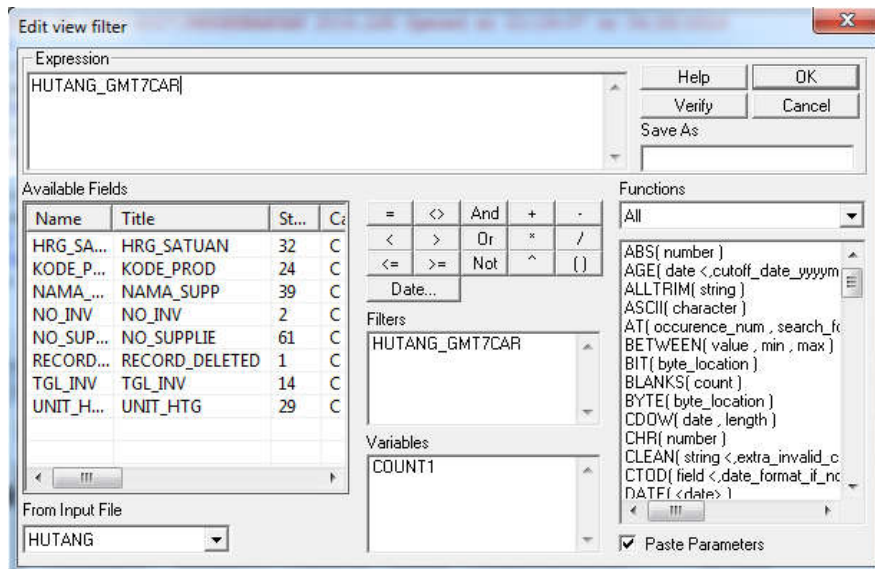
1. Aktifkan data file **DATA\_HUTANG.DBF**.
2. Klik tombol  (**Edit view filter**) pada jendela **View** atau pilih **Tools** dari menu dan pilih **Set Global Filter** untuk membuka kotak dialog set global filter.



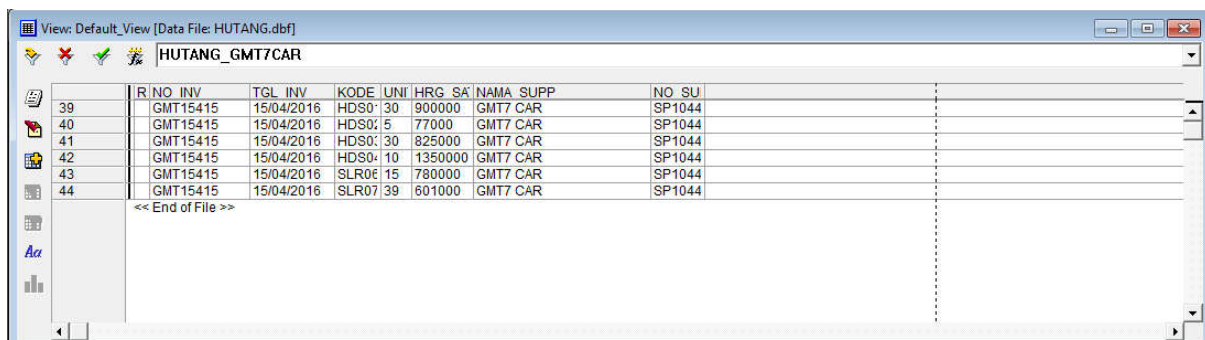
3. Pilih filter **HUTANG\_GMT7CAR** yang telah dibuat.



4. Maka secara otomatis filter **HUTANG\_GMT7CAR** akan muncul pada kolom Expression.



5. Klik **OK** untuk menjalankan filter **HUTANG\_GMT7CAR** sebagai global filter. **HUTANG\_GMT7CAR** akan muncul dalam kotak Default Filter dalam View. Kotak tengah pada status bar juga menunjukkan bahwa **HUTANG\_GMT7CAR** adalah global filter yang sedang berjalan.





**Kasus 1 :**

Seorang auditor pada PT. LABALA ingin melakukan pemeriksaan terhadap data perusahaan. Salah satu aspek yang akan diperiksa adalah data pegawai. Auditor membutuhkan data pegawai perusahaan untuk memeriksa gaji bersih yang lebih besar sama dengan Rp 6.000.000 dan hutang per periode lebih kecil sama dengan Rp 500.000. Lalu auditor akan membuat filter untuk mempermudah pekerjaan audit dan menyimpannya dengan nama file **GJ\_BERSIH\_AUDIT**.

**Kasus 2 :**

Kepala HRD PT LABALA ingin mengetahui pajak pegawainya. Kepala HRD ingin mengetahui pajak yang lebih besar sama dengan Rp 120.000 dan akan dilampirkan dalam laporan pajak pegawai perusahaan.



## RELATION & SUMMARIZE

Relation digunakan untuk mempermudah user dalam melihat hubungan relasi antara, field dalam 2 atau lebih file. Pada Relation sering digunakan istilah Parent File dan Child File. **Parent file** adalah file induk yang memiliki beberapa field atau data yang sama dengan child file, sedangkan **Child file** adalah file yang akan dihubungkan dengan parent file.

Dalam proses relation, field kunci harus di **index** terlebih dahulu. Fungsi index pada dasarnya memiliki fungsi yang sama dengan sort, tetapi index tidak membentuk data file baru, melainkan membentuk index file yang bisa di-namai, di-save, dan di-apply ke input file kapanpun. Jika index diaktifkan, maka record-record akan ditampilkan sesuai urutan yang diinginkan.

Perintah summarize digunakan untuk menampilkan ringkasan data berdasarkan field tertentu (field kunci) dan field yang mengikutinya kemudian dihitung jumlah record sesuai dengan field kuncinya.

### Contoh Kasus:

Pada akhir periode, bagian accounting pada PT.LABALA ingin melakukan pemeriksaan terhadap data piutang customer yang dimiliki perusahaan. Untuk melakukan pemeriksaan tersebut perlu dilakukan identifikasi antara hubungan **Data Piutang**, **Data Pelanggan**, dan **Data Persediaan**. Field kunci yang dibutuhkan untuk **Data Pelanggan** adalah **Nomor Pelanggan** dan pada **Data Persediaan** adalah **Kode Produk**. Setelah itu perlu dibuat ringkasan dari hasil identifikasi tersebut dalam satu file dengan nama **Relation One** dengan menambahkan field **Nama Produk** dari **Data persediaan**, **Harga satuan** dari **Data Piutang**, serta **Nama Pelanggan** dan **Alamat Lengkap** dari **Data Pelanggan**. Berikut merupakan langkah – langkah yang harus dilakukan PT.LABALA dalam melakukan identifikasi menggunakan software ACL.

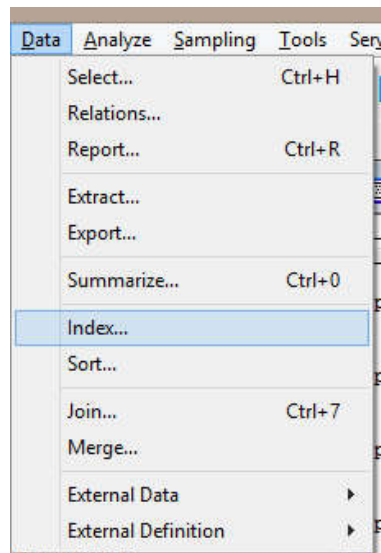
### A. Mendefinisikan Rencana

1. Data file **PIUTANG.DBF** sebagai **Parent file**.
2. Data file **PELANGGAN.DBF** dan **PERS\_PRODUK.DBF** sebagai **Child files**.
3. **Piutang** dan **Pelanggan** dapat dihubungkan dengan field **NO\_PLGN**.
4. **Piutang** dan **Persediaan** dapat dihubungkan dengan field **KODE\_PROD**.
5. Indexlah **Pelanggan** berdasarkan field kunci **NO\_PLGN** dan **Persediaan** berdasarkan

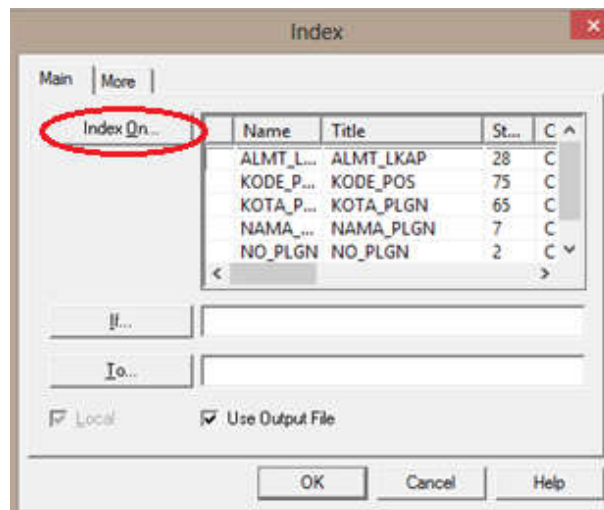
field kunci **KODE\_PROD**.

## B. Mengindex File Child

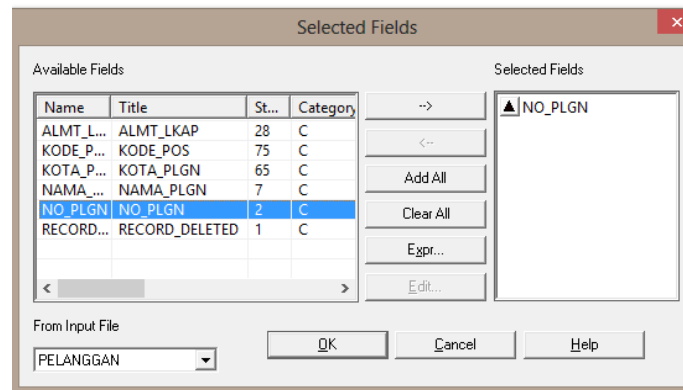
1. Untuk mengindex **Pelanggan**, aktif pada data file **PELANGGAN.DBF**
2. Pilih **Data** dari menu bar, lalu klik **Index**.



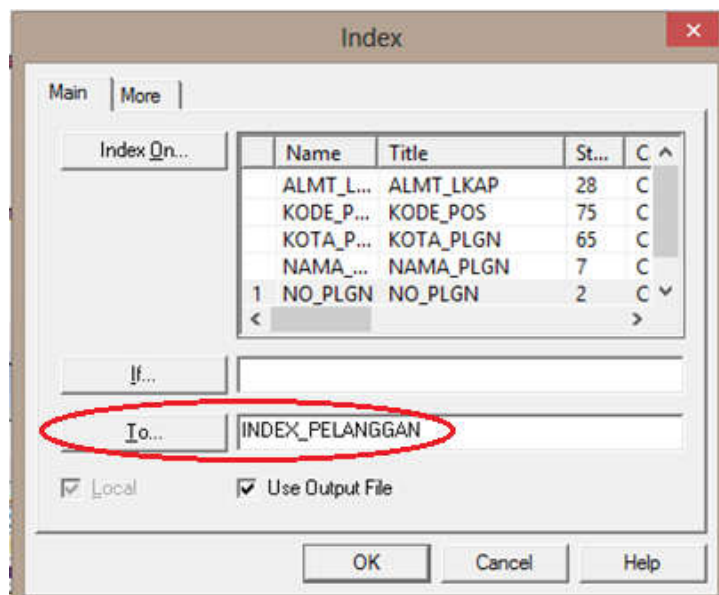
3. Pada tab **Main** klik list box **Index On** lalu pilih **NO\_PLGN** untuk memilih field kunci yang akan diindex, Kemudian klik **OK**.







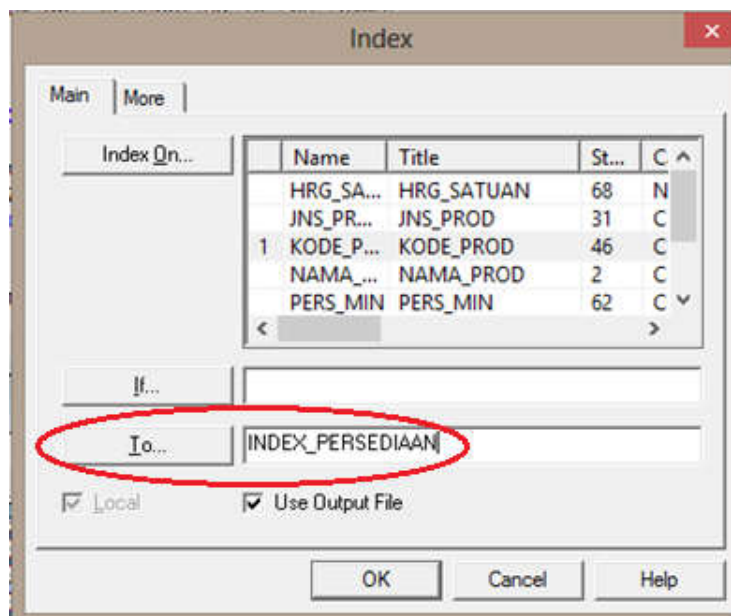
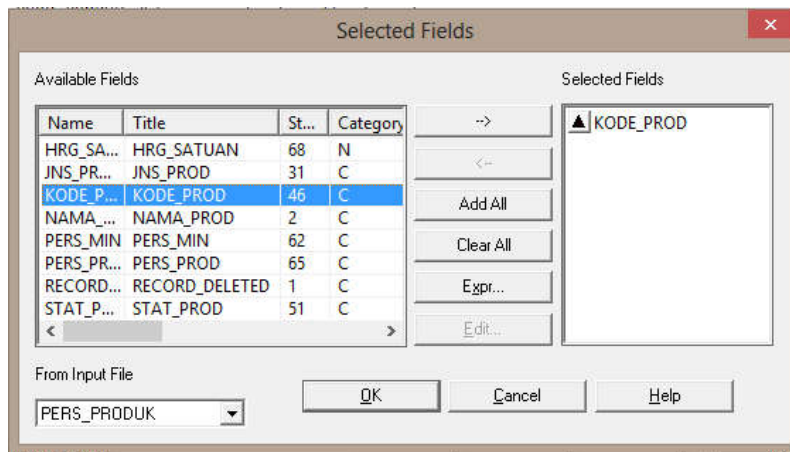
4. Dalam kotak **To**, ketik **INDEX\_PELANGGAN** untuk memberi nama file index, lalu klik **OK**.



5. Setelah Klik **OK**. Jumlah record yang diindex akan tampak di kotak status line bagian tengah. ACL menayangkan hasilnya dalam window command log sebagai berikut:

```
@ INDEX ON NO_PLGN TO "INDEX_PELANGGAN" OPEN
70 records produced
Output to C:\Users\kurniawan\Downloads\data acl labala 2016\INDEX_PELANGGAN.INX is done
Opening file "INDEX_PELANGGAN"
@ SET INDEX TO "INDEX_PELANGGAN"
```

6. Untuk mengindex **Persediaan**, aktif pada file **PERS\_PRODUK.DBF**.
7. Ulangi langkah 2 sampai 5, untuk **Index On** klik **KODE\_PROD**, dan pada kotak **To** ketik **INDEX\_PERSEDIAAN** lalu klik **OK**.

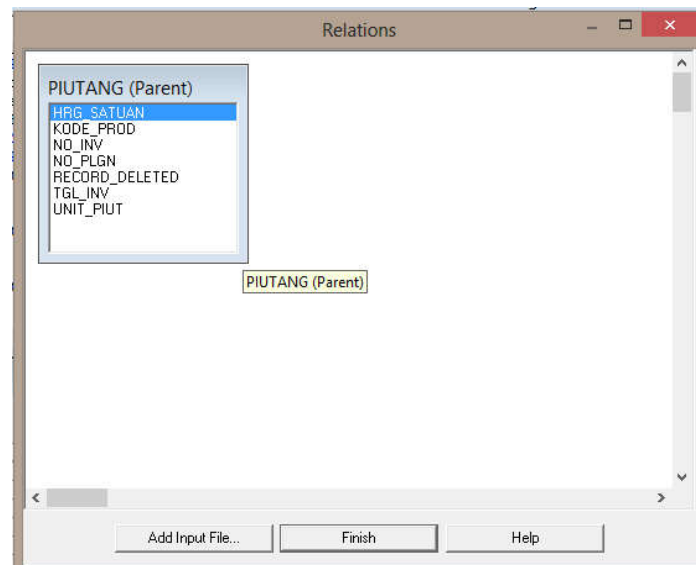


8. Setelah Klik **OK**. Jumlah record yang diindex akan tampak di kotak status line bagian tengah. ACL menayangkan hasilnya dalam window command log sebagai berikut:

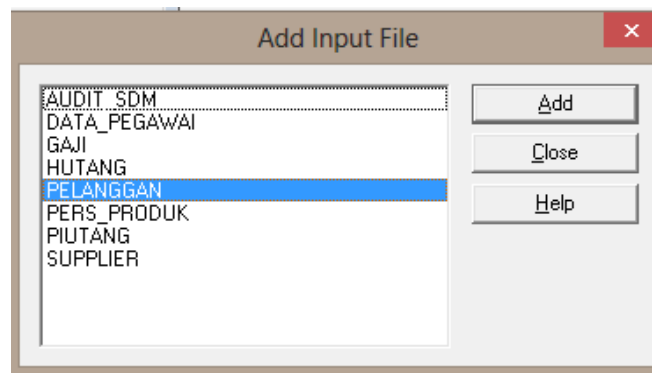
```
@ INDEX ON KODE_PROD TO "INDEX_PERSEDIAAN" OFEN
135 records produced
Output to C:\Users\kurniawan\Downloads\data acl labala 2016\INDEX_PERSEDIAAN.INX is done
Opening file "INDEX_PERSEDIAAN"
@ SET INDEX TO "INDEX_PERSEDIAAN"
```

### C. Menetapkan Relasi Data

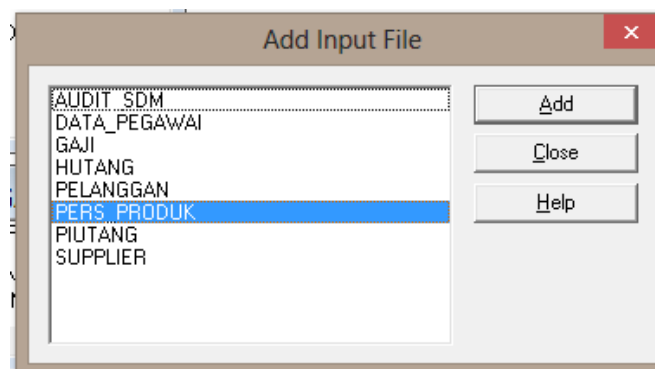
1. Aktifkan file pada **parent file**, yaitu data file **PIUTANG.DBF**, lalu pilih **Data** pada Menu Bar dan klik **Relations**. Lalu pilih **Add Input File**.



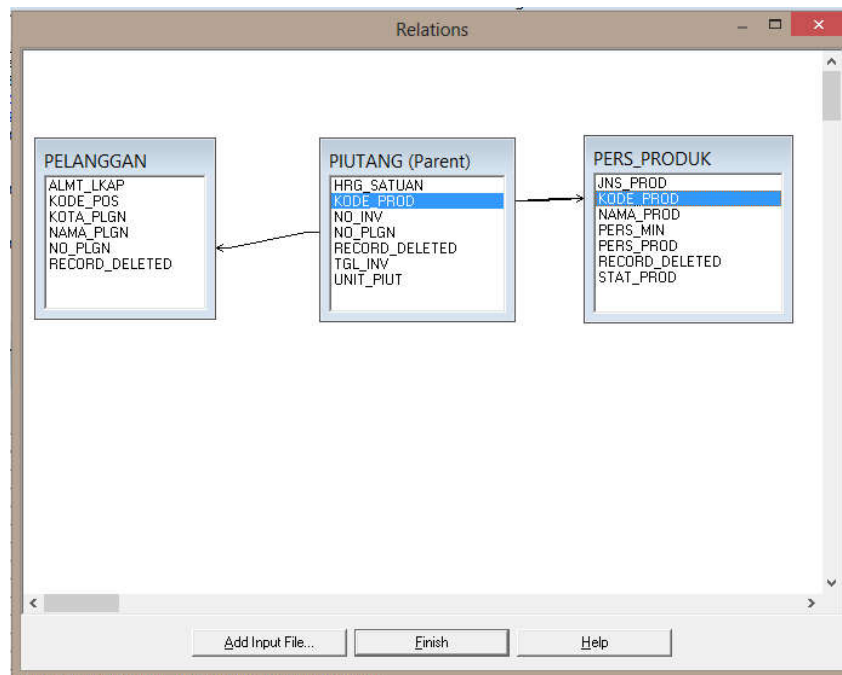
2. Pada button **Add Input File** klik file **PELANGGAN**, lalu klik **Add**.



3. Lakukan hal yang sama untuk Add file **PERS\_PRODUK**, lalu **Add** Kemudian **Close**.

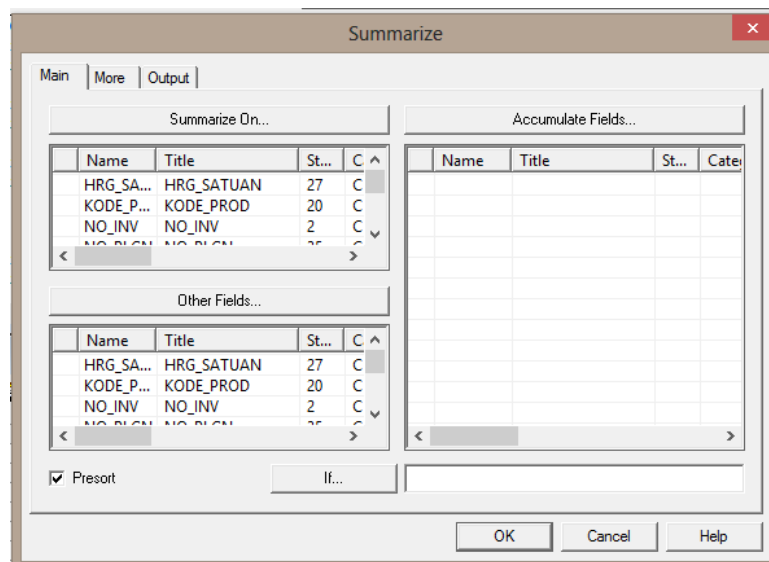


4. Untuk merelasionkan drag field **NO\_PLGN** pada file **PELANGGAN** ke **NO\_PLGN** pada file **PIUTANG**. Lalu drag field **KODE\_PROD** yang ada pada file **PERS\_PRODUK** ke **KODE\_PROD** pada file **PIUTANG** lalu klik **Finish**.

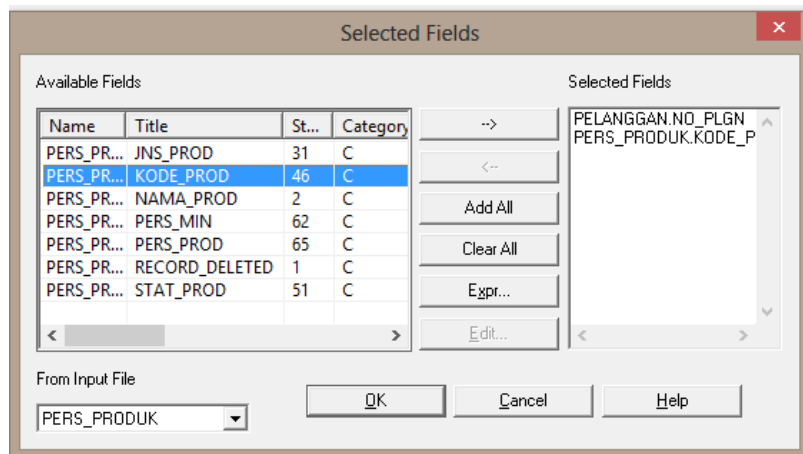
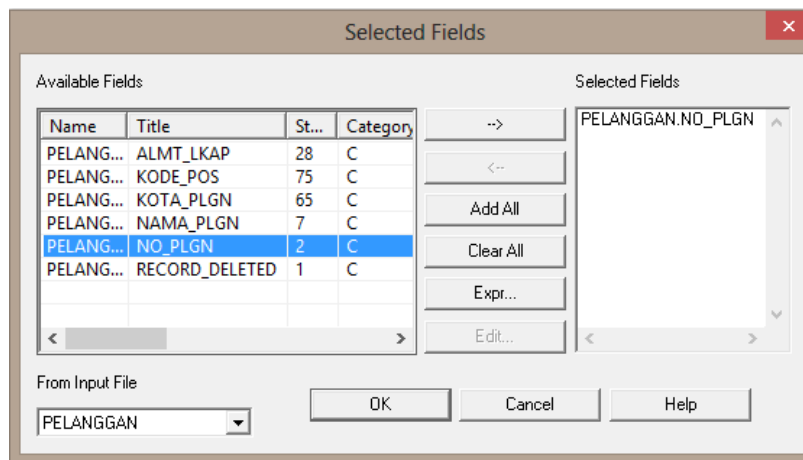


### C. Membuat Ringkasan

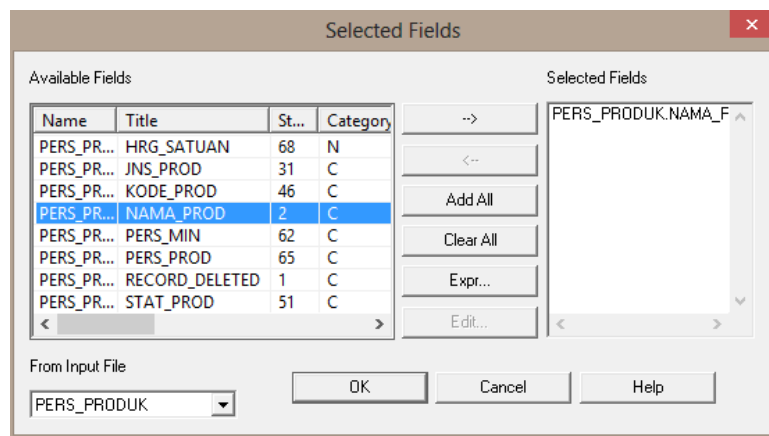
1. Aktifkan data pada **Parent File**, yaitu data file **PIUTANG**, lalu pilih **Data** pada **Menu Bar** dan klik **Summarize**. Sehingga akan muncul jendela seperti berikut.

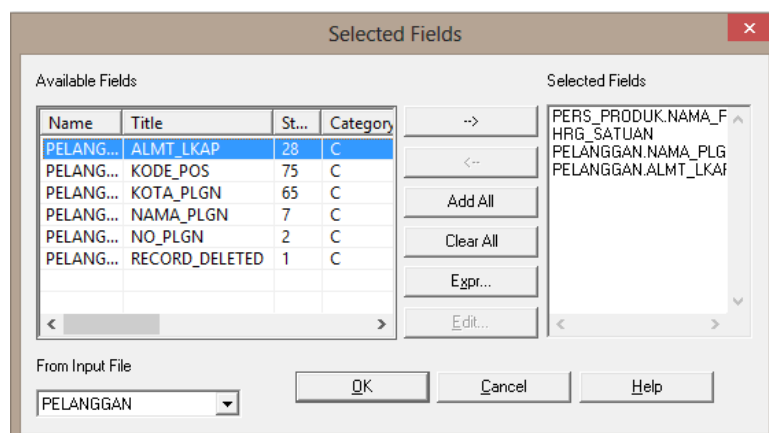
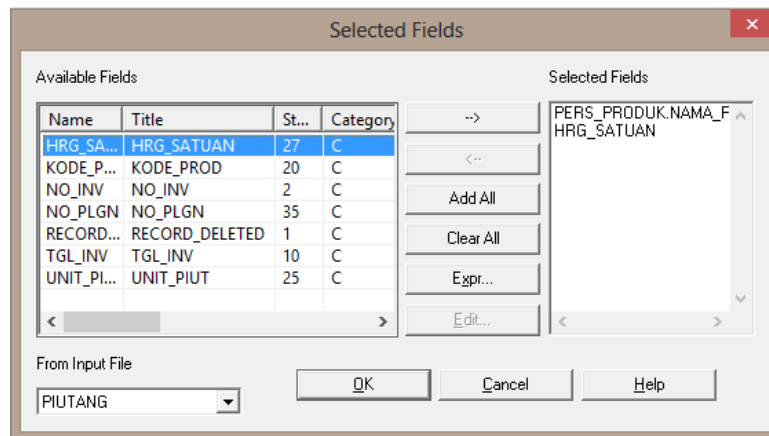


2. Klik **Summerize On**, masukan field Kunci pada masing masing Child File, lalu klik **OK**.

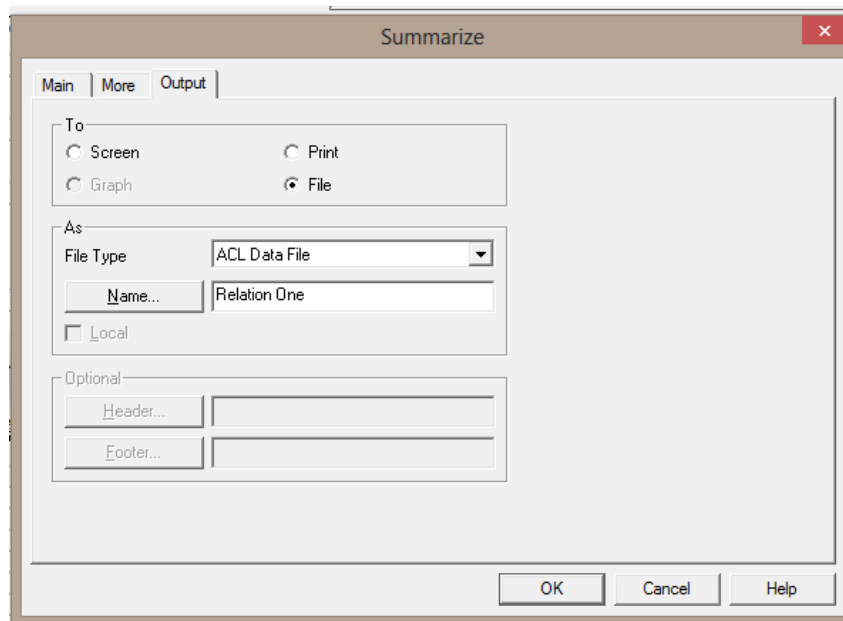


3. Klik **Other Fields**, masukkan field tambahan seperti sesuai dengan soal pada masing – masing file , kemudian klik **OK**.





4. Di tab **Output**, pada **To** pilih **File**, dan pada **As** ketikkan nama file yang akan disimpan (**Relation One**) lalu klik **OK**.



5. Maka hasil dari fungsi Summarize dapat dilihat pada command log dan memunculkan file baru berupa **Relation One**.



```
@ SUMMARIZE ON PELANGGAN.NO_FLGN PERS_PRODUK.KODE_PROD OTHER PERS_PRODUK.NAMA_PROD HRG_SATUAN PELANGGAN.NAMA_FLGN PELANGGAN.ALMT_LKAP TO "Relation One.1"
Presorting data
65 records produced
Opening file "Relation_One"
@ OPEN "Relation_One"
7 fields activated
Opening file name Relation One.FIL as supplied in the format.
```

View: Default\_View [Data File: Relation One.FIL]

NO	PL	KODE	COUNT	NAMA_PROD	HRG_SATI	NAMA_PLGN	ALMT_LKAP
1	C0001	LTP03	1	ASUS ROG 15	15000000	Ari Rizkia Handayani	Jl. Ku Bukan Jalan Mu No. 12
2	C0002	AND01	1	ASUS ZENFONE 2 KG500KL	21000000	Axel Rizki	Jl. Menuju Roma No. 23 Blok C
3	C0002	EXT01	1	SAMSUNG S1 1.8" 200	4000000	Axel Rizki	Jl. Menuju Roma No. 23 Blok C
4	C0003	IOS03	1	APPLE I PHONE 2S	37000000	Darul Bawar Maulana	Jl. Jalan Terus Kapan Jadiannya No. 1
5	C0003	WDS1	1	ACER LIQUID C4	30000000	Darul Bawar Maulana	Jl. Jalan Terus Kapan Jadiannya No. 1
6	C0004	LTP15	1	ACER Z01 ROG01	130000000	Frischa Lamria	Jl. Kebenaran No. 1
7	C0004	MPL02	1	HEWLETT PACKARD MP	10000000	Frischa Lamria	Jl. Kebenaran No. 1
8	C0005	MPL02	1	SONY NANO EAR	22000000	Jevrendhi Ginting	Jl. Babelan Mas Permai No.23
9	C0005	SLR07	1	NIKON EXPERIA X2	200000000	Jevrendhi Ginting	Jl. Babelan Mas Permai No.23
10	C0007	WDS1	1	HTC ONE	50000000	Nurul Azizah	Jl. Jalan Sendiri Terus No. 11
11	C0011	SLR06	1	NIKON EXPERIA X1500	180000000	Mahmudin Abanda	Jl. Gedebage No. 13 Blok J
12	C0012	EXT03	1	TOSHIBA BASIC EXT 2.5	3800000	Muhammad Arsad	Jl. Penyjukan Raja No. 21
13	C0013	AND02	1	ASUS ZENTAB 7 DELUXE	70000000	Almarhumah	Jl. Sukahati No. 04 blok F3

### Kasus 1 :

Pada akhir periode, bagian accounting pada PT.LABALA ingin melakukan pemeriksaan terhadap data hutang yang dimiliki perusahaan. Untuk melakukan pemeriksaan tersebut perlu dilakukan identifikasi antara hubungan **Data Hutang**, **Data Supplier** dan **Data Persediaan**. Field kunci yang dibutuhkan untuk **Data Supplier** adalah **Nomor Supplier** dan pada **Data Persediaan** adalah **Kode Produk**. Setelah itu perlu dibuat ringkasan dari hasil identifikasi tersebut dalam satu file dengan nama **RelationTwo** dengan menambahkan field **Nomor Invoice** dari **Data Hutang**, **Nama dan Kota** dari **Data Vendor**, serta **Nama Produk** dari **Data Persediaan**. Berikut merupakan langkah – langkah yang harus dilakukan PT.LABALA dalam melakukan identifikasi menggunakan software ACL.

### Kasus 2 :

Pada akhir periode, bagian kepegawaian pada PT.LABALA ingin melakukan pemeriksaan terhadap data pegawai yang dimiliki perusahaan. Untuk melakukan pemeriksaan tersebut perlu dilakukan identifikasi antara hubungan **Data Pegawai**, **Data Gaji**, dan **Data Audit SDM**. Field kunci yang dibutuhkan untuk **Data Gaji** adalah **Nomor Pegawai** dan pada **Data Audit SDM** adalah **Jabatan**. Setelah itu perlu dibuat ringkasan dari hasil identifikasi tersebut dalam satu tampilan screen dengan menambahkan field **Nama Pegawai** dari **Data Pegawai**, **Divisi, Tanggal Pembayaran, Gaji Bersih** dari **Data Gaji**, serta **Bonus** dari **Data Audit SDM**. Berikut merupakan langkah – langkah yang harus dilakukan PT.LABALA dalam melakukan identifikasi menggunakan software ACL.



## MySQL

### A. PENGENALAN TENTANG MySQL

MySQL pertama kali dirintis oleh seorang programmer database bernama Michael Widenius. MySQL database server adalah RDBMS (Relasional Database Management System) yang dapat menangani data yang bervolume besar. MySQL adalah database yang paling populer diantara database-database lainnya.

MySQL adalah program database yang mampu mengirim dan menerima data dengan sangat cepat dan multiuser. MySQL memiliki dua bentuk lisensi, yaitu free software dan shareware.

MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya; SQL (Structured Query Language). SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis.

Fase penting dalam pengembangan MySQL:

- MySQL dirilis pertama kali secara internal pada 23 Mei 1995.
- Versi windows dirilis pada 08 Januari 1998 untuk windows 95 dan windows NT.
- Versi 3.23 : beta dari bulan Juni 2000, dan dirilis pada bulan Januari 2001.
- Versi 4.0 : beta dari bulan Agustus 2002, dan dirilis pada bulan Maret 2003.
- Versi 4.1 : beta dari bulan Juni 2004, dan dirilis pada bulan Oktober 2004.
- Versi 5.0 : beta dari bulan Maret 2005, dan dirilis pada bulan Oktober 2005.
- Sun Microsystems membeli MySQLAB pada tanggal 26 Februari 2008.
- Versi 5.1 dirilis 27 November 2008.

### B. TIPE DATA MySQL

Sering kali pada saat kita membuat sebuah database dan membuat fields yang ada didalamnya, kita kebingungan dalam memilih tipe data apa yang akan digunakan. Berikut ini adalah tipe data yang terdapat dalam MySQL :





- Tipe data untuk bilangan

No	Tipe Data	Keterangan
1	TINYINT	Ukuran 1 Byte, bilangan bulat terkecil, dengan jangkauan untuk bilangan bertanda -128 s/d 127 dan untuk yang tidak bertanda 0 s/d 255. Bilangan tak bertanda ditandai dengan kata UNSIGNED.
2	SMALLINT	Ukuran 2 Byte , bilangan bulat dengan jangkauan untuk bilangan bertanda-32768 s/d 32767 dan untuk yang tidak bertanda 0 s/d 65535.
3	MEDIUMINT	Ukuran 3 Byte, bilangan bulat dengan jangkauan untuk bilangan bertanda-8388608 s/d 8336607 dan untuk yang tidak bertanda 0 s/d 16777215.
4	INT	Ukuran 4 Byte, bilangan bulat dengan jangkauan untuk bilangan bertanda-2147483648 s/d 2147483647 dan untuk yang tidak bertanda 0 s/d 4294967295.
5	INTEGER	Sama Dengan INT
6	BIGINT	Ukuran 4 Byte, bilangan bulat dengan jangkauan untuk bilangan bertanda-9223372036854775808 s/d 9223372036854775807 dan untuk yang tidak bertanda 0 s/d 18446744073709551615.
7	FLOAT	Ukuran 4 Byte, bilangan pecahan.
8	DOUBLE	Ukuran 8 Byte, bilangan pecahan.
9	DOUBLE PRECISION	Ukuran 8 Byte, bilangan pecahan berpresisi ganda.
10	REAL	Ukuran 8 Byte, bilangan pecahan (sinonim dari DOUBLE).
11	DECIMAL (M,D)	Ukuran M Byte, bilangan pecahan. Misalnya DECIMAL (5,2) dapat digunakan untuk menyimpan bilangan -99,99 s/d 99,99.
12	NUMERIC (M,D)	Ukuran M Byte, sama dengan DECIMAL.

- Tipedata untuk tanggal dan jam

No	Tipe Data	Keterangan
1	DATETIME	Ukuran 8 Byte. Kombinasi tanggal dan jam dengan jangkauan dari 1000-01-01 00:00:00 s/d 9999-12-31 23:59:59.
2	DATE	Ukuran 8 Byte. Kombinasi tanggal dan jam dengan jangkauan dari 1000-01-01 s/d 9999-12-31.
3	TIMESTAMP	Ukuran 4 Byte. Kombinasi tanggal dan jam dengan jangkauan dari 1970-01-01 s/d 2037.



4	TIME	Ukuran 3 Byte. Waktu dengan jangkauan dari -838:59:59 s/d 838:59:59.
5	YEAR	Ukuran 1 Byte. Data tahun antara 1901 s/d 2155.

- Tipe data untuk karakter dan lain-lain

No	Tipe Data	Keterangan
1	CHAR (M)	Ukuran M byte, $1 \leq M \leq 255$ . Data string dengan panjang yang tetap. CHAR (1) cukup ditulis dengan CHAR.
2	VARCHAR (M)	Ukuran L+1 Byte dengan $L \leq M$ dan $1 \leq M \leq 255$ . Data string dengan panjang bervariasi tergantung datanya.
3	TINYBLOB, TINYTEXT	L+1 Byte, dengan $1 < 2^8$ . Type TEXT atau BLOB dengan panjang maksimum 255 karakter.
4	BLOB, TEXT	L+2 Byte, dengan $1 < 2^8$ . Type TEXT atau BLOB dengan panjang maksimum 255 karakter.
5	MEDIUMBLOB, MEDIUMTEXT	L+3 Byte, dengan $1 < 2^8$ . Type TEXT atau BLOB dengan panjang maksimum 255 karakter.
6	LONGBLOB, LONGTEXT	L+4 Byte, dengan $1 < 2^8$ . Type TEXT atau BLOB dengan panjang maksimum 255 karakter.
7	ENUM ( 'nilai1', 'nilai2', ... )	Ukuran 1 atau 2 Byte tergantung nilai enumerasinya maksimum 65535 nilai.
8	SET ( 'nilai1', 'nilai2', ... )	Ukuran 1, 2, 3, 4 atau 8 Byte tergantung jumlah anggota himpunan maks 64 anggota.

### C. KEISTIMEWAAN MySQL

#### 1. Portabilitas

MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, Free BSD, MacOS XServer, Solaris, Amiga, dan masih banyak lagi.

#### 2. Database yang Open Source

MySQL didistribusikan secara open source, dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.

#### 3. Multiuser

MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.



#### 4. **Performance Tuning**

MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQLpersatuan waktu.

#### 5. **Jenis Kolom**

MySQL memiliki tipe kolom yang sangat kompleks, seperti signed/ unsigned, integer, float, double, char, text, date, timestamp, dan lain-lain.

#### 6. **Perintah dan Fungsi**

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (query).

#### 7. **Keamanan**

MySQL memiliki beberapa lapisan sekuritas seperti level sub net mask, nama host dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.

#### 8. **Skalabilitas dan Pembatasan**

MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu table serta 5 milyar baris.Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

#### 9. **Konektivitas**

MySQL dapat melakukan koneksi dengan klien menggunakan protocol TCP/IP, Unixsocket (UNIX), atau Named Pipes (NT).

#### 10. **Lokalisasi**

MySQL dapat mendeteksi pesan kesalahan-pada klien dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk di dalamnya.

#### 11. **Antar Muka**

MySQL memiliki interface (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).

#### 12. **Klien dan Peralatan**

MySQL dilengkapi dengan berbagai peralatan (tool) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada.

#### 13. **Struktur Tabel**

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTE TABLE, dibandingkan basis data lainnya semacam Postgre SQL ataupun Oracle.



## D. PERINTAH DASAR MySQL

Untuk membangun suatu database pada program MySQL mempunyai perintah dasar. Perintah dasar yang digunakan dibagi menjadi 3 yaitu, perintah DDL (Data Definition Language), DML (Data Manipulation Language), dan DCL (Data Control Language).

### 1. PERINTAH DDL

DDL (Data Definition Language) merupakan sub bahasa SQL yang digunakan untuk membangun kerangka dari sebuah database ada 3 perintah yang termasuk dalam DDL, yaitu:

- **CREATE** : Perintah yang digunakan untuk membuat, termasuk diantaranya database baru, tabel baru, view baru, dan kolom.
- **ALTER** : Perintah ini digunakan untuk merubah struktur tabel yang telah dibuat. Melingkupi mengganti nama tabel. Menambah kolom, mengubah kolom, dan menghapus kolom.
- **DROP** : Perintah ini digunakan untuk menghapus database dan tabel.

### Contoh perintah DDL:

Buatlah database dengan data sebagai berikut:

Nomor Pegawai	Nama Pegawai	Jenis Kelamin	Domisili	Kode Pos	Golongan Pegawai
2891	Ari Rizkia	P	Bekasi	17612	III
9442	Axel Rizky	L	Depok	16425	I
1273	Darul Maulana	L	Bogor	16524	II
3094	Frischa Lamria	P	Bogor	16109	II
6055	Jevrendhi	L	Tangerang	15160	I
3016	Kurniawan Eka	L	Depok	16245	III
1287	Nurul Azizah	P	Jakarta	16254	II
1128	Rachmasari	P	Bogor	11011	I
2459	Sriningsih	P	Depok	17425	I
3220	Mita Septiani	P	Jakarta	17610	I

### ✓ PERINTAH CREATE

Membuat database dengan nama "audit".

*CREATE DATABASE nama\_database;*

```
mysql> create database audit;  
Query OK, 1 row affected (0.00 sec)
```



Untuk melihat database yang telah kita buat, gunakan perintah:

*SHOW DATABASES;*=

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| audit      |
| mysql     |
| performance_schema |
| sakila    |
| test     |
| testing  |
| world    |
+-----+
8 rows in set (0.00 sec)
```

Untuk mengaktifkan database yang ingin kita pakai, gunakan perintah:

*USE nama\_database;*

```
mysql> use audit;
Database changed
```

Setelah mengaktifkan database, maka untuk membuat tabel dari database “audit”. Sintaks penulisannya sebagai berikut:

```
CREATE TABLEnama_tabel (
Kolom1 tipe_data opsi_kolom,
Kolom2 tipe_data opsi_kolom,
.....
KolomN tipe_data opsi_kolom,
PRIMARYKEY(nama_kolom)
);
```

```
mysql> create table Data_Pegawai(
-> Nomor_Pegawai int(20) NOT NULL AUTO_INCREMENT,
-> Nama_Pegawai varchar(50) NOT NULL,
-> Jenis_Kelamin varchar(10) NOT NULL,
-> Domisili varchar(50) NOT NULL,
-> Kode_Pos varchar(20) NOT NULL,
-> PRIMARY KEY (Nomor_Pegawai)
-> );
Query OK, 0 rows affected (0.55 sec)
```

Setelah membuat tabel, untuk melihat output dari tabel tersebut gunakan perintah:

*SHOW TABLES;*

```
mysql> show tables;
+-----+
| Tables_in_audit |
+-----+
| data_pegawai    |
+-----+
1 row in set (0.00 sec)
```



Untuk memeriksa tabel yang telah dibuat sesuai dengan spesifikasi yang ditentukan, gunakan perintah:

*DESCRIBE nama\_tabel; atau DESC nama\_tabel;*

```
mysql> desc Data_Pegawai;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| Nomor_Pegawai | int(20)       | NO   | PRI | NULL    | auto_increment |
| Nama_Pegawai  | varchar(50)   | NO   |     | NULL    |                |
| Jenis_Kelamin | varchar(10)   | NO   |     | NULL    |                |
| Domisili      | varchar(50)   | NO   |     | NULL    |                |
| Kode_Pos      | varchar(20)   | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

### ✓ PERINTAH ALTER

Membuat field “Golongan Pegawai” pada tabel “Data Pegawai” menggunakan perintah:

*ALTER TABLE nama\_tabel*

*ADD COLUMN (column\_definition,...);*

```
mysql> alter table Data_Pegawai
  -> add column Golongan_Pegawai varchar(20) NOT NULL;
Query OK, 0 rows affected (0.30 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Setelah menjalankan perintah diatas maka Data\_Pegawai akan bertambah kolom Golongan\_Pegawai:

```
mysql> desc Data_Pegawai;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| Nomor_Pegawai | int(20)       | NO   | PRI | NULL    | auto_increment |
| Nama_Pegawai  | varchar(50)   | NO   |     | NULL    |                |
| Jenis_Kelamin | varchar(10)   | NO   |     | NULL    |                |
| Domisili      | varchar(50)   | NO   |     | NULL    |                |
| Kode_Pos      | varchar(20)   | NO   |     | NULL    |                |
| Golongan_Pegawai | varchar(20)   | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

### ✓ PERINTAH DROP

Jika kita menghapus database maka tabel-tabel yang ada didalamnya juga akan terhapus.

*DROP DATABASE nama\_database;*

**Keterangan :** Jangan drop database jika ingin melanjutkan ke perintah selanjutnya.



## 2. PERINTAH DML

DML (Data Manipulation Language) merupakan sub bahasa SQL yang digunakan untuk memanipulasi dalam database yang telah dibuat, perintah yang digunakan diantaranya adalah :

- **INSERT** : Perintah ini digunakan untuk memasukkan data kedalam tabel yang sudah dibuat.
- **SELECT** : Perintah ini digunakan untuk menampilkan isi dari tabel atau relasinya. Data yang tampil dapat melalui console MySQL atau pada aplikasi.
- **UPDATE** : Perintah ini digunakan untuk memperbaharui data lama menjadi data terbaru.
- **DELETE** : Perintah ini digunakan untuk menghapus isi dari tabel. Pada saat perintah dijalankan, data yang telah dihapus tidak dapat dikembalikan lagi.

### ✓ PERINTAH INSERT

Memasukkan data secara bersamaan kedalam database “audit”. Sebelum memasukkan data ke dalam tabel “audit” perlu diperhatikan tipe data yang digunakan dalam setiap kolom. Lalu masukkan sintaks sebagai berikut :

*INSERT INTO nama\_tabel VALUES*

*(‘data\_kol1’, ‘data\_kol2’, ‘.....’, ‘data\_kolN’), (‘data\_kol1’, ‘data\_kol2’, ‘.....’, ‘data\_kolN’);*

```
mysql> insert into Data_Pegawai values
-> (2891,'Ari Rizkia','P','Bekasi','17612','III'),
-> (9442,'Axel Rizky','L','Depok','16425','I'),
-> (1273,'Darul Maulana','L','Bogor','16524','II'),
-> (3094,'Frischa Lamria','P','Bogor','16109','II'),
-> (6055,'Jevrendhi','L','Tangerang','15160','I'),
-> (3016,'Kurniawan Eka','L','Depok','16245','III'),
-> (1287,'Nurul Azizah','P','Jakarta','16254','I'),
-> (1128,'Rachmasari','P','Bogor','11011','I'),
-> (2459,'Sriningsih','P','Depok','17425','I'),
-> (3220,'Mita Septiani','P','Jakarta','17610','II');
Query OK, 10 rows affected (0.08 sec)
Records: 10 Duplicates: 0 Warnings: 0
```

### ✓ PERINTAH SELECT

Perintah SELECT digunakan untuk menampilkan kembali sebuah data yang telah disimpan didalam database. Pengambilan data dapat dilakukan dengan beberapa cara diantaranya adalah :



➤ Pengambilan data sederhana

Untuk melihat data-data yang telah dimasukkan ke dalam tabel “Data\_Pegawai” menggunakan sintaks :

*SELECT \*FROM nama\_tabel;*

```
mysql> select *from Data_Pegawai;
+-----+-----+-----+-----+-----+-----+
| Nomor_Pegawai | Nama_Pegawai | Jenis_Kelamin | Domisili | Kode_Pos | Golongan_Pegawai |
+-----+-----+-----+-----+-----+-----+
| 1128 | Rachmasari | P | Bogor | 11011 | I |
| 1273 | Darul Maulana | L | Bogor | 16524 | II |
| 1287 | Nurul Azizah | P | Jakarta | 16254 | I |
| 2459 | Sriningsih | P | Depok | 17425 | I |
| 2891 | Ari Rizkia | P | Bekasi | 17612 | III |
| 3016 | Kurniawan Eka | L | Depok | 16245 | III |
| 3094 | Frischa Lamria | P | Bogor | 16109 | II |
| 3220 | Mita Septiani | P | Jakarta | 17610 | II |
| 6055 | Jevrendhi | L | Tangerang | 15160 | I |
| 9442 | Axel Rizky | L | Depok | 16425 | I |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0.06 sec)
```

Selain itu perintah SELECT juga dapat digunakan untuk menampilkan data berdasarkan kolom-kolom, dengan menentukan nama kolom yang akan diambil. Gunakan perintah :

*SELECT daftar\_pilihan\_kolom FROM nama\_tabel;*

```
mysql> select Nama_Pegawai, Domisili from Data_Pegawai;
+-----+-----+
| Nama_Pegawai | Domisili |
+-----+-----+
| Rachmasari | Bogor |
| Darul Maulana | Bogor |
| Nurul Azizah | Jakarta |
| Sriningsih | Depok |
| Ari Rizkia | Bekasi |
| Kurniawan Eka | Depok |
| Frischa Lamria | Bogor |
| Mita Septiani | Jakarta |
| Jevrendhi | Tangerang |
| Axel Rizky | Depok |
+-----+-----+
```

➤ Membatasi suatu perintah SELECT dengan WHERE

Dengan menggunakan WHERE dapat mengambil atau memilih baris data tertentu saja dari tabel.





Contoh 1 : Mencari datadari pegawai bernama “Darul Maulana”.

```
SELECT *FROM nama_tabel  
WHERE batasan_batasan_utama;
```

```
mysql> select *from Data_Pegawai where Nama_Pegawai='Darul Maulana';  
+-----+-----+-----+-----+-----+-----+  
| Nomor_Pegawai | Nama_Pegawai | Jenis_Kelamin | Domisili | Kode_Pos | Golongan |  
+-----+-----+-----+-----+-----+-----+  
| 1273 | Darul Maulana | L | Bogor | 16524 | II |  
+-----+-----+-----+-----+-----+-----+  
1 row in set (0.13 sec)
```

- Mengurutkan pemilihan data menggunakan ORDER BY Perintah *ORDER BY* digunakan untuk *mengurutkan baris data*.

Contoh 1 :Mengurutkan baris data dimana Nomor\_Pegawai dipengaruhi oleh urutan "Nama\_Pegawai".

```
SELECT daftar_pilihan_kolom *FROM nama_tabel  
WHERE Nomor_Pegawai ORDER BY Nama_Pegawai;
```

```
mysql> select *from Data_Pegawai where Nomor_Pegawai order by Nama_Pegawai;  
+-----+-----+-----+-----+-----+-----+  
| Nomor_Pegawai | Nama_Pegawai | Jenis_Kelamin | Domisili | Kode_Pos | Golon  
gan_Pegawai |  
+-----+-----+-----+-----+-----+-----+  
| 2891 | Ari Rizkia | P | Bekasi | 17612 | III |  
| 9442 | Axel Rizky | L | Depok | 16425 | I |  
| 1273 | Darul Maulana | L | Bogor | 16524 | II |  
| 3094 | Frischa Lamria | P | Bogor | 16109 | II |  
| 6055 | Jevrendhi | L | Tangerang | 15160 | I |  
| 3016 | Kurniawan Eka | L | Depok | 16245 | III |  
| 3220 | Mita Septiani | P | Jakarta | 17610 | II |  
| 1287 | Nurul Azizah | P | Jakarta | 16254 | I |  
| 1128 | Rachmasari | P | Bogor | 11011 | I |  
| 2459 | Sriningsih | P | Depok | 17425 | I |  
+-----+-----+-----+-----+-----+-----+  
10 rows in set (0.10 sec)
```

- Membatasi pengambilan data menggunakan klausa LIMIT

Perintah LIMIT berguna untuk mengambil baris data berdasarkan posisinya dalam sekumpulan baris data hasil pemilihan LIMIT mungkin diberikan dengan satu atau dua agrument, yang mana berupa.

*LIMIT jumlah\_baris*

*LIMIT jumlah\_diabaikan,jumlah\_baris*



Contoh : Menampilkan tabel Data\_Pegawai sebanyak 2 baris pertama :

```
mysql> select *from Data_Pegawai where Nomor_pegawai limit 2;
```

Nomor_Pegawai	Nama_Pegawai	Jenis_Kelamin	Domisili	Kode_Pos	Golongan
1128	Rachmasari	P	Bogor	11011	I
1273	Darul Maulana	L	Bogor	16524	II

```
2 rows in set (0.00 sec)
```

➤ Mengelompokkan pemilihan baris data menggunakan klausa GROUPBY

Perintah GROUPBY adalah fungsi untuk mengelompokkan data dalam sebuah kelompok yang ditunjuk. Fungsi ini akan menghasilkan kelompok data dengan menghilangkan data yang sama dalam satu tabel. Jadi, apabila dalam satu kolom terdapat beberapa data yang sama, data yang akan ditampilkan hanya salah satu.

```
mysql> select Domisili from Data_Pegawai group by Domisili;
```

Domisili
Bekasi
Bogor
Depok
Jakarta
Tangerang

```
5 rows in set (0.33 sec)
```

### ✓ PERINTAH UPDATE

Perintah UPDATE dipakai untuk mengubah isi data di baris-baris yang ada.

*UPDATE nama\_tableSET*

*Kolom1='data\_baru',*

*Kolom2='data\_baru',*

.....

*kolomN='data\_baru'*

*WHERE[kondisi];*

Contoh : Merubah Nama\_Pegawai “Ari Rizkia” menjadi “Ari Handayani”.

```
mysql> update Data_Pegawai set Nama_Pegawai = 'Ari Handayani' where Nomor_Pegawai = '2891';
Query OK, 1 row affected (0.11 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```



### ✓ PERINTAH DELETE

Menggunakan perintah DELETE tetap harus dengan klausa WHERE karena jika tidak, maka data dalam tabel akan terhapus semua.

Contoh : Akan menghapus baris dengan nomor pegawai "2891".

```
mysql> delete from Data_Pegawai where Nomor_Pegawai = '2891';  
Query OK, 1 row affected (0.09 sec)
```

Dari gambar diatas dapat terlihat bahwa pegawai dengan nama Ari Rizkia dengan Nomor Pegawai 2891 sudah dihapus dari dalam tabel Data Pegawai.

### Kasus 1 :

Buatlah tabel dengan nama Data\_Barang kemudian buatlah tabel dengan data sebagai berikut :

KODE_BARANG	NAMA_BARANG	JENIS_BARANG	STOK_BARANG	HARGA_BARANG
143	ASUS ZENPHONE 2 DELUXE	ANDROID SP	20	6000000
166	APPLE I PHONE 5	IOS SP	40	5500000
247	LOGITECH Z313	SPEAKER	50	288000
274	HTC ONE ULTRA	WINDOWS SP	19	6000000
278	SAMSUNG S1	DVD	25	400000
416	SHARP AQUOUS	LED TV	20	3600000
473	HEWLETT PACKARD	LAPTOP	79	5500000
543	CANON FOCUS	SLR CAMERA	17	14000000
563	TOSHIBA OPERA	HEADSET	15	180000
679	LG GENERATION	LCD TV	30	2500000

### Kasus 2 :

- Tampilkan tabel Data\_Barang dengan hanya menampilkan kolom Nama Barang dan Harga Barang !
- Tampilkan baris hanya yang memiliki Harga Barang 3.600.000 !
- Tampilkan tabel Data\_Barang dengan 3 baris pertama!
- Tampilkan dengan urutan data Kode Barang dengan 5 baris pertama dimulai dari baris ke 5 !



## INNER JOIN

### Contoh soal:

1. Pada akhir tahun, PT. LABALA ingin melihat gaji para pegawainya.
2. Selain itu PT. LABALA juga ingin melihat data-data gaji dan bonus pegawainya sesuai golongan dan status pegawainya.

### Tabel Pegawai

NIP	Nama	Alamat	Golongan	Kode
3556	Darul Maulana	Depok	2C	T
4444	Nurul Azizah	Jakarta	1B	H
2343	Anis Setyarini	Bekasi	2A	S
1432	Rachmasari Octarina	Bogor	1A	T
6666	Rizky Andika	Bekasi	2B	T
7766	Sriningsih	Bekasi	2C	S

### Tabel Gaji

Golongan	Gaji	Bonus
1A	40.000.000	10.000.000
1B	35.000.000	8.000.000
2A	30.000.000	6.000.000
2B	25.000.000	4.000.000
2C	20.000.000	2.000.000

### Tabel Status

Kode	Status
T	Tetap
S	Sementara
H	Honorer

### 1. Membuat Tabel Pegawai :

```
mysql> create table pegawai(\n  -> nip int(15) not null primary key,\n  -> nama varchar(20) not null,\n  -> alamat varchar(15) not null,\n  -> golongan varchar(5) not null,\n  -> kode varchar(5) not null);\nQuery OK, 0 rows affected (0.14 sec)
```



2. Untuk memeriksa tabel yang telah dibuat sesuai dengan spesifikasi yang ditentukan, gunakan perintah:

```
mysql> desc pegawai;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nip   | int(15)       | NO   | PRI | NULL    |       |
| nama  | varchar(20)   | NO   |     | NULL    |       |
| alamat | varchar(15)   | NO   |     | NULL    |       |
| golongan | varchar(5)   | NO   |     | NULL    |       |
| kode  | varchar(5)    | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

3. Memasukan Data Pada Tabel

```
mysql> insert into pegawai values
-> (3556,'Darul Maulana','Depok','2C','T'),
-> (4444,'Nurul Azizah','Jakarta','1B','H'),
-> (2343,'Anis Setyarini','Bekasi','2A','S'),
-> (1432,'Rachmasari Octarina','Bogor','1A','T'),
-> (6666,'Rizky Andika','Bekasi','2B','T'),
-> (7766,'Sriningsih','Bekasi','2C','S');
Query OK, 6 rows affected (0.42 sec)
Records: 6 Duplicates: 0 Warnings: 0
```

4. Melihat Data-Data yang Telah Dimasukkan ke Dalam Tabel

```
mysql> select *from pegawai;
+-----+-----+-----+-----+-----+
| nip | nama                | alamat | golongan | kode |
+-----+-----+-----+-----+-----+
| 1432 | Rachmasari Octarina | Bogor  | 1A       | T   |
| 2343 | Anis Setyarini      | Bekasi | 2A       | S   |
| 3556 | Darul Maulana       | Depok  | 2C       | T   |
| 4444 | Nurul Azizah        | Jakarta | 1B       | H   |
| 6666 | Rizky Andika        | Bekasi | 2B       | T   |
| 7766 | Sriningsih          | Bekasi | 2C       | S   |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```



## 5. Membuat Tabel Gaji

```
mysql> create table gaji(  
  -> golongan varchar(5) not null primary key,  
  -> gaji int(20) not null,  
  -> bonus int(20) not null);  
Query OK, 0 rows affected (0.41 sec)
```

6. Untuk memeriksa tabel yang telah dibuat sesuai dengan spesifikasi yang ditentukan, gunakan perintah:

```
mysql> desc gaji;  
+-----+-----+-----+-----+-----+-----+  
| Field | Type | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| golongan | varchar(5) | NO | PRI | NULL | |  
| gaji | int(20) | NO | | NULL | |  
| bonus | int(20) | NO | | NULL | |  
+-----+-----+-----+-----+-----+-----+  
3 rows in set (0.01 sec)
```

## 7. Memasukkan Data Pada Tabel

```
mysql> insert into gaji values  
  -> ('1A', '40000000', '10000000'),  
  -> ('1B', '35000000', '8000000'),  
  -> ('2A', '30000000', '6000000'),  
  -> ('2B', '25000000', '4000000'),  
  -> ('2C', '20000000', '2000000');  
Query OK, 5 rows affected (0.10 sec)  
Records: 5 Duplicates: 0 Warnings: 0
```

## 8. Melihat Data-Data yang Telah Dimasukkan ke Dalam Tabel

```
mysql> select *from gaji;  
+-----+-----+-----+  
| golongan | gaji | bonus |  
+-----+-----+-----+  
| 1A | 40000000 | 10000000 |  
| 1B | 35000000 | 8000000 |  
| 2A | 30000000 | 6000000 |  
| 2B | 25000000 | 4000000 |  
| 2C | 20000000 | 2000000 |  
+-----+-----+-----+  
5 rows in set (0.00 sec)
```

## 9. Membuat Tabel Status

```
mysql> create table status(  
  -> kode varchar(5) not null primary key,  
  -> status varchar(20) not null);  
Query OK, 0 rows affected (0.40 sec)
```



10. Untuk memeriksa tabel yang telah dibuat sesuai dengan spesifikasi yang ditentukan, gunakan perintah:

```
mysql> desc status;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| kode  | varchar(5)    | NO   | PRI | NULL    |       |
| status| varchar(20)   | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)
```

11. Memasukkan Data Pada Tabel

```
mysql> insert into status values
-> ('T','Tetap'),
-> ('H','Honorar'),
-> ('S','Sementara');
Query OK, 3 rows affected (0.08 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

12. Melihat Data-Data yang Telah Dimasukkan ke Dalam Tabel

```
mysql> select *from status;
+-----+-----+
| kode | status |
+-----+-----+
| H    | Honorar |
| S    | Sementara |
| T    | Tetap |
+-----+-----+
3 rows in set (0.00 sec)
```

Jawaban :

1.

```
mysql> select pegawai.nip, pegawai.nama, gaji.gaji
-> from pegawai inner join gaji
-> on pegawai.golongan = gaji.golongan;
+-----+-----+-----+
| nip  | nama                | gaji |
+-----+-----+-----+
| 3556 | Darul Maulana       | 20000000 |
| 4444 | Nurul Azizah       | 35000000 |
| 2343 | Anis Setyarini     | 30000000 |
| 1432 | Rachmasari Octarina | 40000000 |
| 6666 | Rizky Andika       | 25000000 |
| 7766 | Sriningsih         | 20000000 |
+-----+-----+-----+
```



2.

```
mysql> select pegawai.nip, pegawai.nama, status.status, gaji.gaji, gaji.bonus
-> from pegawai inner join status inner join gaji
-> on pegawai.golongan=gaji.golongan and status.kode=pegawai.kode;
```

nip	nama	status	gaji	bonus
3556	Darul Maulana	Tetap	20000000	2000000
4444	Nurul Azizah	Honnorer	35000000	8000000
2343	Anis Setyarini	Sementara	30000000	6000000
1432	Rachmasari Octarina	Tetap	40000000	10000000
6666	Rizky Andika	Tetap	25000000	4000000
7766	Sriningsih	Sementara	20000000	2000000

6 rows in set (0.00 sec)