

Pengenalan PL/SQL

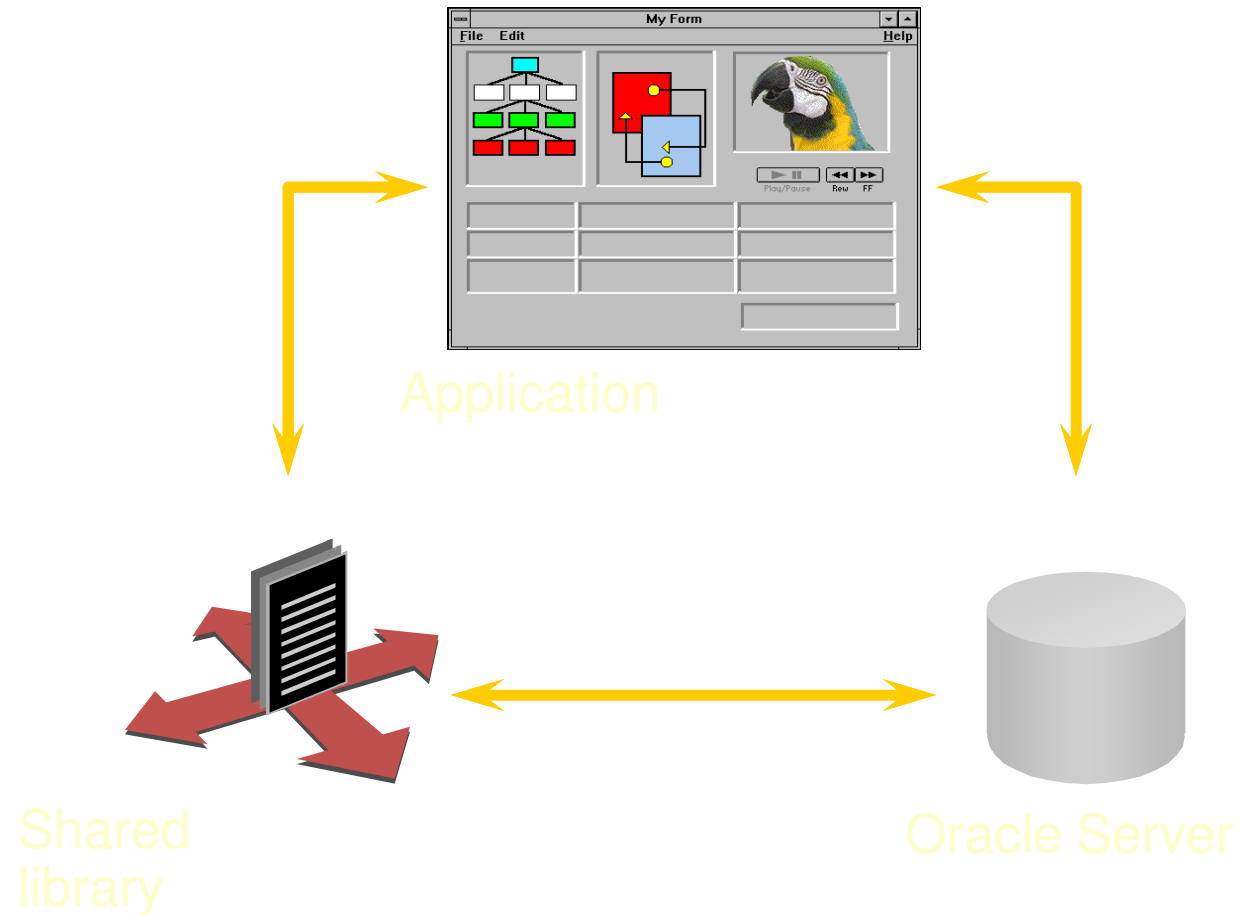
Pertemuan 3

Apakah PL/SQL

- PL/SQL adalah perluasan SQL dengan fitur bahasa pemrograman .
- SQL Data manipulation and query statements disertakan dalam unit kode prosedural.

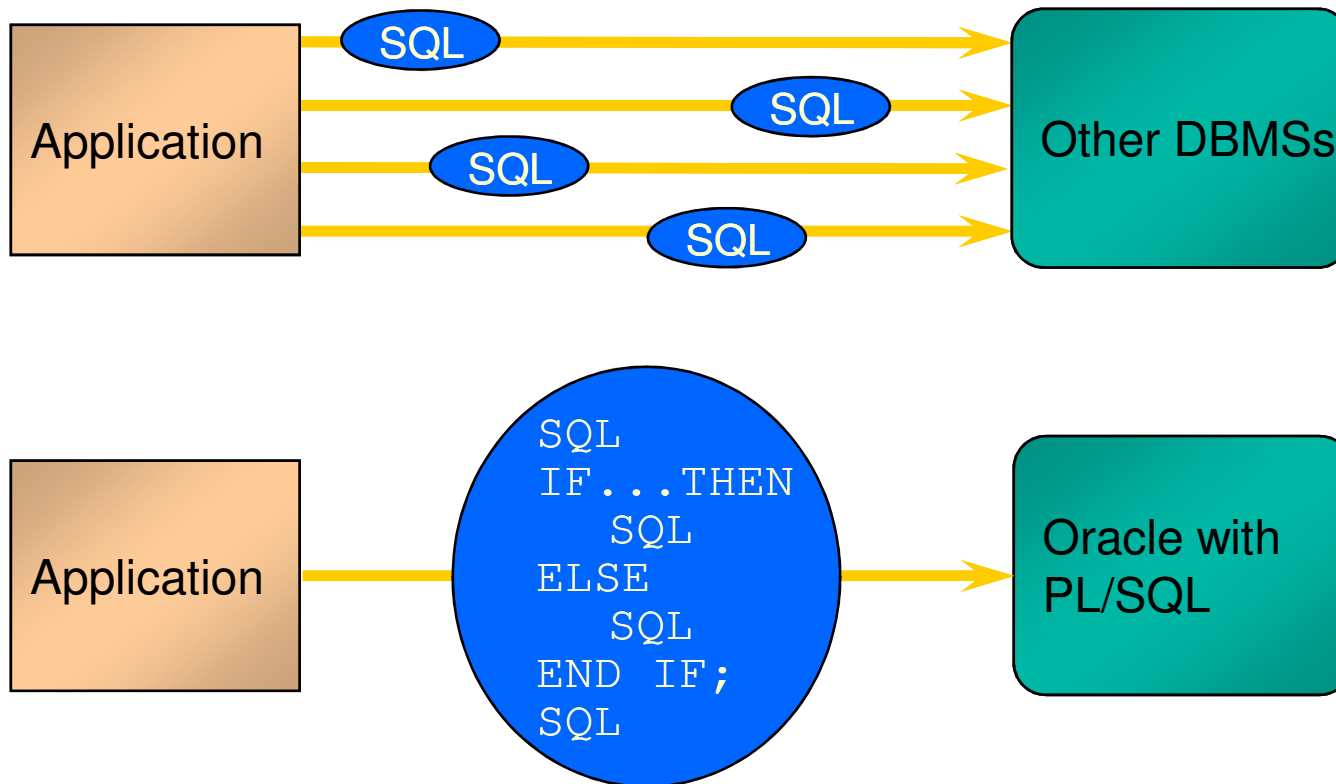
Keuntungan PL/SQL

- Integrasi



Keuntungan PL/SQL

- Meningkatkan Unjuk Kerja



Struktur Blok PL/SQL

- DECLARE – Optional
Variables, cursors, user-defined exceptions
- BEGIN – Harus ada
 - Statemen SQL
 - Statemen PL/SQL
- EXCEPTION – Opsional
Aksi yang ditampilkan ketika terjadi error
- END; – Harus ada

```
DECLARE  
...  
BEGIN  
...  
EXCEPTION  
...  
END;
```



Struktur Blok PL/SQL

```
DECLARE
    v_variable  VARCHAR2(5);
BEGIN
    SELECT      column_name
    INTO        v_variable
    FROM        table_name;
EXCEPTION
    WHEN exception_name THEN
        ...
END;
```

```
DECLARE
...
BEGIN
...
EXCEPTION
...
END;
```



Type Block

- Anonymous

```
[DECLARE]

BEGIN
  --statements

[EXCEPTION]

END;
```

Prosedur

```
PROCEDURE name
IS
BEGIN
  --statements

[EXCEPTION]

END;
```

Fungsi

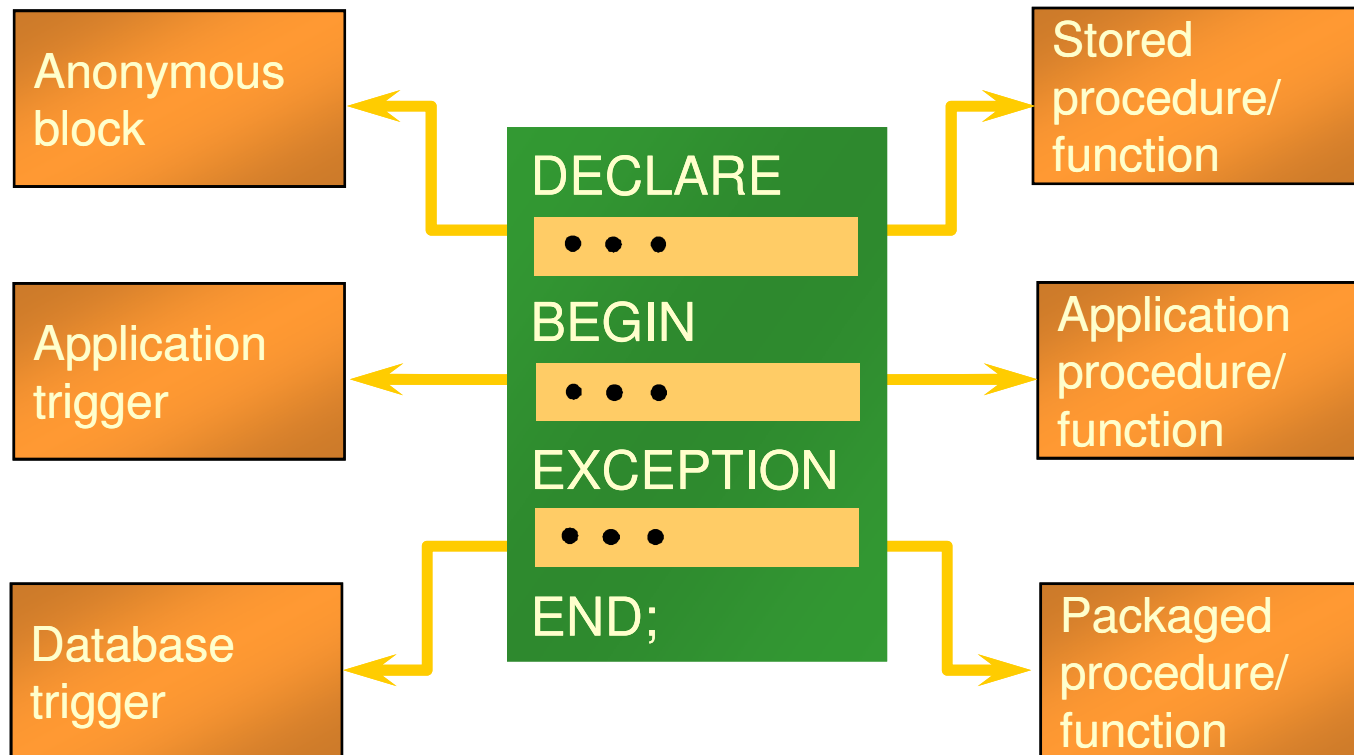
```
FUNCTION name
RETURN datatype
IS
BEGIN
  --statements
  RETURN value;

[EXCEPTION]

END;
```



Program Constructs



Kegunaan Variables

- Variables digunakan untuk:
 - Menyimpan data sementara
 - Memanipulasi nilai yang disimpan
 - Dapat digunakan kembali
 - Mudah dalam pemeliharaan



Penanganan Variable dalam PL/SQL

- Deklarasi dan inisialisasi variabel pada declaration section.
- Pemberian nilai baru untuk variable pada executable section.
- Melewatkan nilai kedalam blok PL/SQL melalui parameter.
- Menampilkan hasil melalui variable output



Tipe Variables

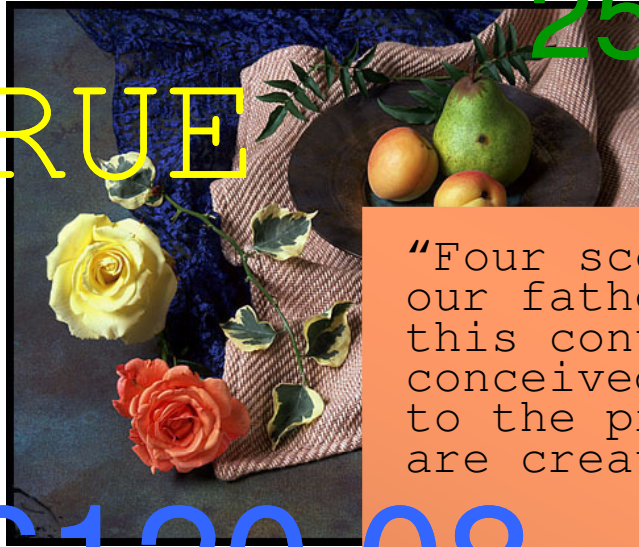
- Variabel PL/SQL s:
 - Scalar
 - Composite
 - Reference
 - LOB (large objects)
- Variabel Non-PL/SQL : Bind and host variables



Type Variable

25-OCT-99

TRUE



"Four score and seven years ago our fathers brought forth upon this continent, a new nation, conceived in LIBERTY, and dedicated to the proposition that all men are created equal."

256120.08



Atlanta



Deklarasi Variabel PL/SQL

Syntax

```
identifier [CONSTANT] datatype [NOT NULL]  
[:= | DEFAULT expr];
```

Contoh

```
Declare  
  v_hiredate      DATE;  
  v_deptno        NUMBER(2) NOT NULL := 10;  
  v_location      VARCHAR2(13) := 'Atlanta';  
  c_comm          CONSTANT NUMBER := 1400;
```



Deklarasi Variabel PL/SQL

- Guideline
 - Mengikuti aturan penamaan.
 - Inisialisasi variable dengan NOT NULL dan CONSTANT.
 - Inisialisasi identifier menggunakan assignment operator (:=) atau reserved word DEFAULT.
 - Deklarasi paling banyak satu identifier per baris.



Aturan Penamaan

- Dua variabel dapat memiliki nama yang sama, disediakan pada blok yang berbeda.
- Nama variabel (identifikasi) tidak seharusnya sama dengan nama kolom yang digunakan pada blok PL/SQL.

```
DECLARE
  empno NUMBER ( 4 ) ;
BEGIN
  SELECT      empno
  INTO        empno
  FROM        emp
  WHERE       ename = 'SMITH' ;
END;
```

*Adopt a naming convention for
PL/SQL identifiers:
for example, v_empno*



Pemberian nilai Variable

Syntax

- *identifier := expr;*

Contoh

Pemberian nilai predefined hiredate untuk karyawan baru.

```
v_hiredate := '31-DEC-98';
```

Set the employee name to Maduro.

```
v_ename := 'Maduro';
```



Inisialisasi dan Keyword Variable

- Menggunakan:
 - Assignment operator (:=)
 - DEFAULT keyword
 - NOT NULL constraint



Tipe data Scalar

- Hold a single value
- Have no internal components

25-OCT-99

"Four score and seven years ago our fathers brought forth upon this continent, a new nation, conceived in LIBERTY, and dedicated to the proposition that all men are created equal."

TRUE

256120.08

Atlanta



Tipe data Scalar Dasar

- VARCHAR2 (*maximum_length*)
- NUMBER [(*precision, scale*)]
- DATE
- CHAR [(*maximum_length*)]
- LONG
- LONG RAW
- BOOLEAN
- BINARY_INTEGER
- PLS_INTEGER



Base Scalar Datatypes

- VARCHAR2 (*maximum_length*)
- NUMBER [(*precision, scale*)]
- DATE
- CHAR [(*maximum_length*)]
- LONG
- LONG RAW
- BOOLEAN
- BINARY_INTEGER
- PLS_INTEGER



Deklarasi Variabel Scalar

- Examples

```
v_job          VARCHAR2(9);  
v_count       BINARY_INTEGER := 0;  
v_total_sal   NUMBER(9,2) := 0;  
v_orderdate   DATE := SYSDATE + 7;  
c_tax_rate    CONSTANT NUMBER(3,2) := 8.25;  
v_valid       BOOLEAN NOT NULL := TRUE;
```



Attribute %TYPE

- Deklarasi sebuah variabel sesuai dengan:
 - Definisi kolom sebuah database
 - Variabel yang telah dideklarasikan sebelumnya
- Awali %TYPE dengan:
 - The database table and column
 - The previously declared variable name



Declaring Variables dengan Attribute %TYPE

- Contoh

```
...  
v_ename           emp.ename%TYPE;  
v_balance         NUMBER(7,2);  
v_min_balance     v_balance%TYPE := 10;  
...
```




Deklarasi Boolean Variable

- Hanya nilai TRUE, FALSE, and NULL yang dapat diberikan untuk variable Boolean.
- Variable dihubungkan dengan logical operators AND, OR, dan NOT.
- Variable selalu menghasilkan nilai TRUE, FALSE, atau NULL.
- Ekspresi Arithmetic, character, and date dapat digunakan untuk menghasilkan nilai Boolean.



Structur PL/SQL Record

TRUE	23-DEC-98	ATLANTA	
------	-----------	---------	---

PL/SQL table structure

1	SMITH
2	JONES
3	NANCY
4	TIM

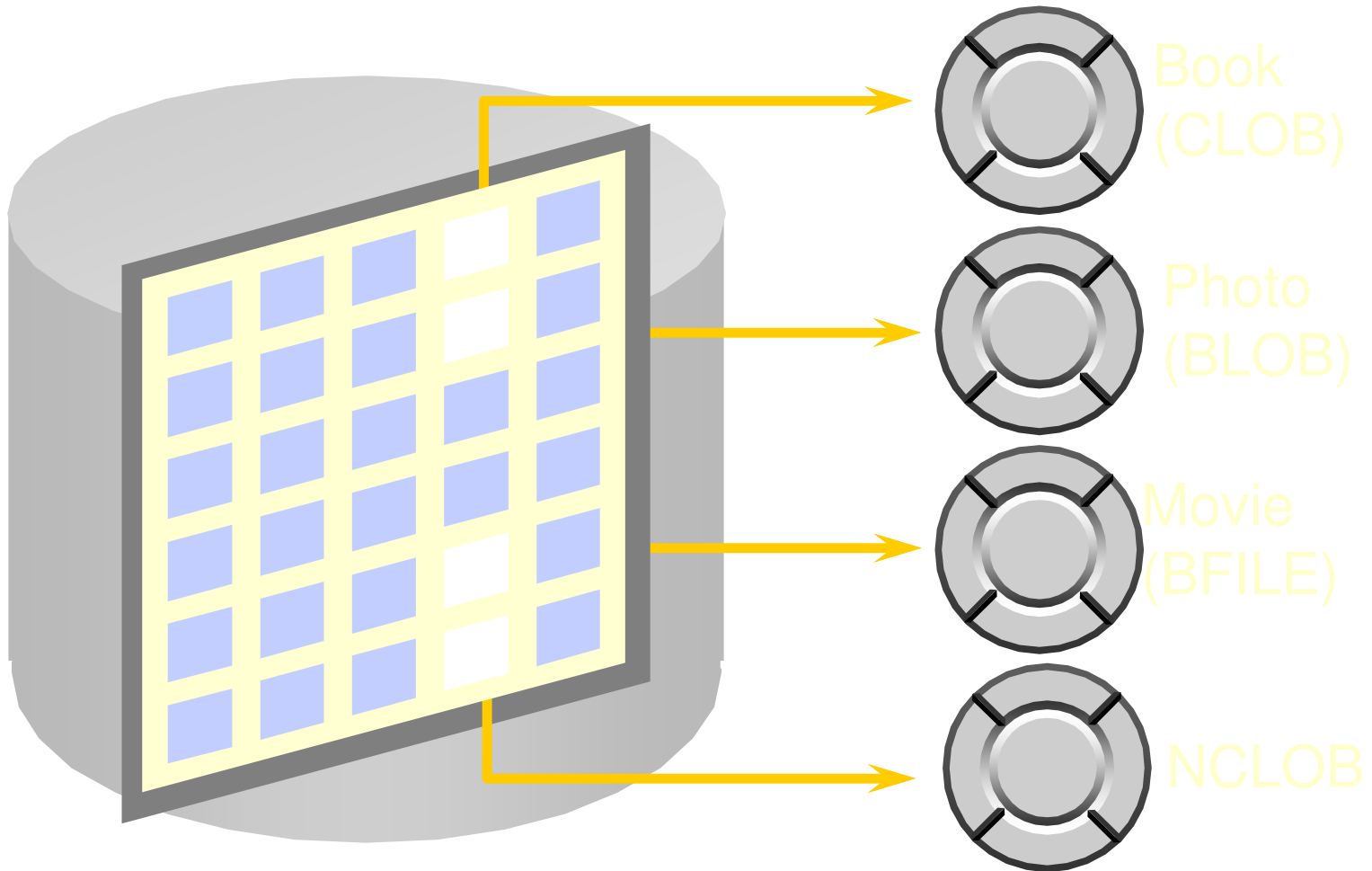


PL/SQL table structure

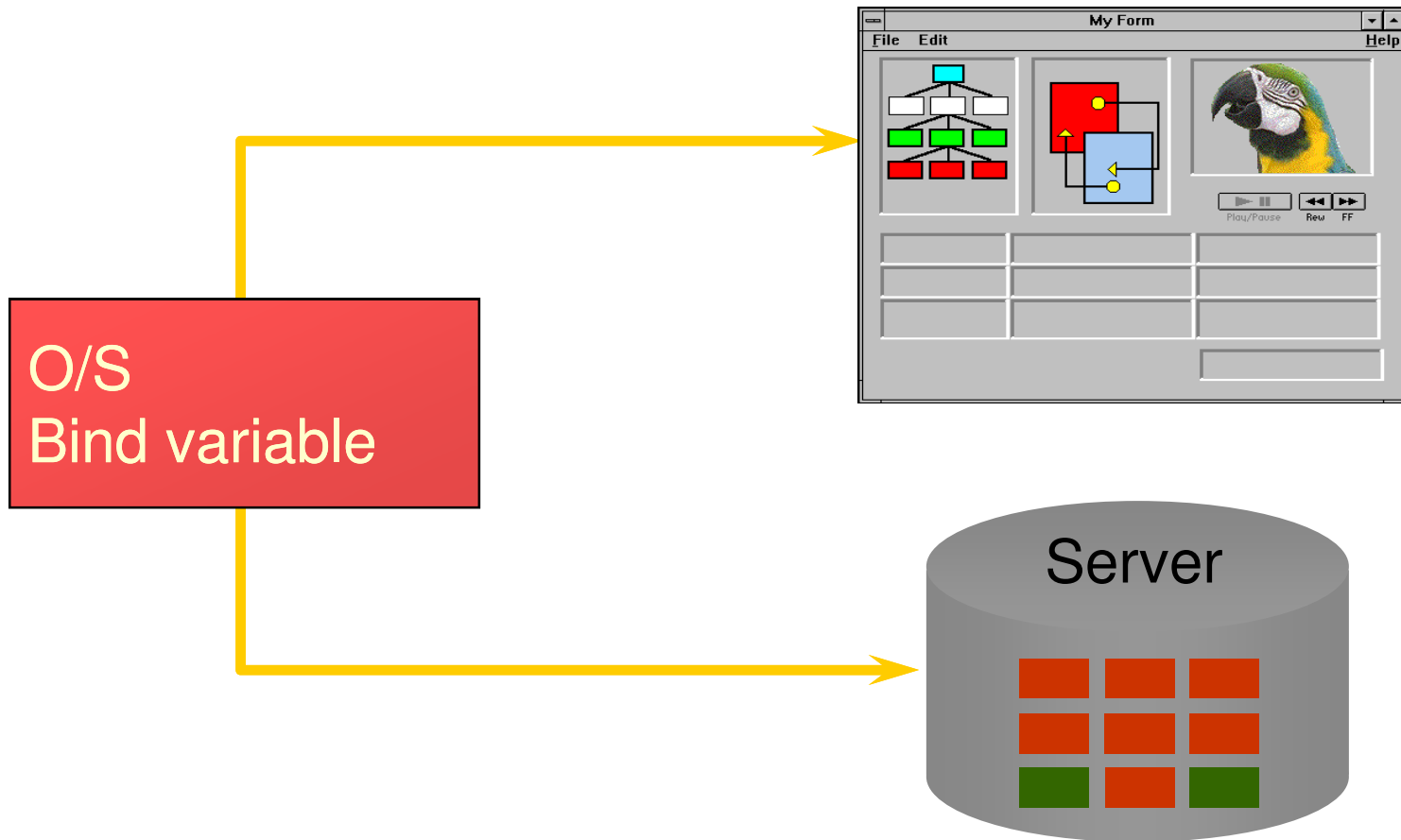
1	5000
2	2345
3	12
4	3456



Type Variable LOB



Bind Variables



Mereferensikan Variabel Non-PL/SQL

- Menyimpan gaji tahunan kedalam SQL*Plus host variable.

```
:g_monthly_sal := v_sal / 12;
```

- Mereferensikan variabel non-PL/SQL sebagai host variables.
- Awali referensi dengan tanda colon (:).



DBMS_OUTPUT.PUT_LINE

- Prosedur Oracle-supplied packaged
- Sebagai alternatif menampilkan data dari blok PL/SQL
- Harus di aktifkan dalam SQL*Plus dengan perintah `SET SERVEROUTPUT ON`



Kesimpulan

- PL/SQL blocks are composed of the following sections:
 - Declarative (optional)
 - Executable (required)
 - Exception handling (optional)
- A PL/SQL block can be an anonymous block, procedure, or function.

```
DECLARE  
...  
BEGIN  
...  
EXCEPTION  
...  
END;
```



Summary

- PL/SQL identifiers:
 - Are defined in the declarative section
 - Can be of scalar, composite, reference, or LOB datatype
 - Can be based on the structure of another variable or database object
 - Can be initialized

