**STANDARD OPERATIONAL PROCEDURE UNTUK PENGEMBANGAN DATABASE WEBSITE IEC**

Diotorisasi oleh Divisi Desain dan Pengembangan Database (2012):

1. Victor S. E. Rumengan ([v.rumengan@yahoo.com](mailto:v.rumengan@yahoo.com))
2. Jovi R. Salira ([salira.jovi.r@gmail.com](mailto:salira.jovi.r@gmail.com))
3. Ria Chaniago ([chaniago\_ria@yahoo.co.id](mailto:chaniago_ria@yahoo.co.id))

**Aturan penamaan untuk objek dalam Database**

Bahasa utama: Bahasa Utama

Dalam dokumen ini, aturan penamaan diberikan sebagai *template* yang terdiri dari beberapa hal yaitu:

* Garis bawah (\_): elemen-elemen dari nama sebuah objek dipisahkan sehingga mudah dibaca.
* Kata miring (italic): menandakan bahwa objek tersebut dinamakan sesuai kemauan DBA. Penamaan harus ditulis dalam format CamelCase (kata tidak dipisahkan oleh spasi dan menggunakan huruf kapital untuk setiap huruf pertama dari setiap kata).
* Kata yang tidak miring: menandakan kata ini adalah konstan yang artinya tidak dapat diganti.
* Kurung siku ([xxx][yyy]): menandakan untuk setiap kata yang di dalam kurung siku adalah pilihan, DBA harus memilih salah satu dari pilihan kata tersebut.
* Simbol tambah (+): menandakan bahwa kedua kata dari sebelah kiri dan kanan harus disatukan.

Sebagai contoh berikut adalah contoh aturan penamaan untuk trigger:  
**trig\_[Insert][Update][Delete]\_*TableName*\_*TriggerName***

* Kata miring (italic) yaitu "***tableName***" dan " ***triggerName*** " menandakan bahwa kata tesebut dinamakan sesuai kemauan DBA.
* Garis bawah memisahkan elemen-elemen yang ada.
* "**Insert**", "**Update**" dan "**Delete**” yang terdapat dalam kurung siku menandakan bahwa DBA harus memilih salah satu dari tiga kata tersebut.
* Kata "**trig**" adalah konstan yang artinya penamaan dimulai dengan kata ini, dan DBA tidak dapat menggantinya.

**Databases**

Untuk penamaan database menggunakan kata tunggal.

**Aturan Penamaan:**

db\_*DatabaseName*

**Contoh:**

db\_Collector

**Tables**

Tabel merepresentasikan instan dari sebuah entitas. Contoh, jika kita menyimpan nama Customer dalam sebuah tabel yang artinya tabel itu berisi data mengenai banyak Custoomer. Oleh karena itu, mengapa kita tidak menamakan tabel tersebut dengan kata jamak.

Jadi untuk tabel Customer kita menamainya 'Customers'.

Untuk tabel Order: 'Orders'.

Untuk tabel Error Message: 'ErrorMessages'.

**Aturan Penamaan:**

tb\_*TableName*

tb\_*TableName1\_TableName2*

*weak entity:* tb\_*TableName*\_WE

**Contoh:**

tb\_Customers

tb\_Customers\_Distributors

**Views**

Berikut adalah penamaan view tergantung dengan kondisi yang dihadapi DBA:

1. Sering sebuah view merupakan representasi dari 2 buah tabel, oleh karena itu penamaan dilakukan dengan memberikan nama dari 2 tabel tsb:  
     
   Jika sebuah view adalah kombinasi dari tabel 'Customers' dan 'Addresses', maka nama view tersebut adalah 'CustomersAddresses'.
2. View bisa merupakan hasil penghitunga dari sebuah tabel atau lebih, tergantung dari fungsi yang dibutuhkan. Oleh karena itu, view dapat dinamakan sesuai dengan deskripsi dari view tersebut:  
   'ProductSalesFor1997'  
   'SummaryOfSalesByQuarter'  
   'SummaryOfSalesByYear'

**Aturan Penamaan:**

vw\_*TableName\_TableName*

vw\_*ViewsDescription*

**Contoh:**

vw\_Customers\_Distributors

vw\_SummaryOfSalesByYear

**Columns**

Penamaan kolom dalam tabel hendaknya singkat dan familiar dengan istilah-istilah umum. Agar tidak menimbulkan ambiguitas nantinya maka untuk setiap nama kolom dimulai dengan prefix (3-4 huruf) dari tabel.

**Aturan Penamaan untuk Primary Key:**

*PrefixOfTables* \_ID

**Contoh:**

Cust\_ID

**Aturan penamaan untuk kolom lain:**

*PrefixOfTable*\_*ColumnName*

**Contoh:**

Cust\_Address

**Aturan penamaan untuk kolom yang merupakan foreign key lain:**

*PrefixOfTable*\_*ColumnNameAtReferencedTable*

**Contoh:**

Cust\_User\_Address

(kolom alamat user pada table *Customer* yang foreign key ke table *User*)

**Indexes**

Index berhubungan dengan tabel dan kolom, maka dinamakan berdasarkan nama tabel dan nama kolomnya. Juga disertakan apakah index tersebut unique atau not-unique untuk kemudahan dalam pengembangan.

**Aturan Penamaan:**

idx\_*TableName*\_[U][NU]\_ *ColumnName*

**Contoh:**

idx\_Customers\_U\_CustAddress

**Constraints**

Constraint berhungan dengan tabel dan kolom, maka dinamakan berdasarkan nama dari tabel dan nama dari kolom yang terkait

**Aturan Penamaan untuk Primay Key, Check, and Unique:**

[pk][chk][uniq]\_*TableName*\_*ColumnName*

**Contoh:**

pk\_Customers\_CustID

**untuk Foreign Key:**

fk\_*TableName*\_*ReferenceTable*\_*ColumnName*

**Contoh:**

fk\_Customers\_Distributors\_DistID

**Triggers**

Trigger dinamakan berdasarkan nama kolomnya dan deskripsi dari trigger tersebut. Diusahakan agar deskripsi singkat namun dapat mewakili proses yang dilakukan trigger tersebut

**Aturan Penamaan:**

trig\_[Insert][Delete][Update]\_*TableName*\_*TriggerDescription*

**Contoh:**

trig\_Insert\_Customers\_Validate

**Stored Procedures**

Stored procedure dinamakan berdasarkan deskripsi dari stored procedure tersebut. Diusahakan agar deskripsi singkat namun dapat mewakili proses yang dilakukan stored procedure tersebut

**Aturan Penamaan:**

proc\_*ProcedureDescription*

**Contoh:**

proc\_CustomersTransfer

**Functions**

Sama seperti stores procedure, function dinamakan dengan deskripsi function tersebut ditambah dengan kode dari tipe data yang dihasilkan function tersebut.

**Aturan Penamaan:**

func\_*ReturnType\_ FunctionName*

**Contoh:**

func\_ch\_ValidateCustomers

|  |  |
| --- | --- |
| **KODE** | **TIPE DATA** |
| In | *int* |
| si | *smallint* |
| ti | *tinyint* |
| nm | *numeric* |
| dc | *decimal* |
| fl | *float* |
| re | *real* |
| dp | *double precision* |
| mo | *money* |
| sm | *small money* |
| ch | *char* |
| vc | *varchar* |
| bi | *binary* |
| vb | *varbinary* |
| bt | *bit* |
| dt | *datetime* |
| sd | *smalldatetime* |
| tx | *text* |
| im | *image* |

\* DBA dapat menambahan tipe data balikan sesuai dengan kebutuhan.

Layak dibaca:

1. DBA adalah singkatan dari Database Administrator.
2. Penggunakan spasi, digit, dan special character lain adalah ide yang sangat buruk, namun dapat ditolerir tergantung masalah yang didapat.
3. Singkatan yang digunakan diharapkan singkatan yang familiar.
4. Hindari pencampuran bahasa dalam sebuah kata.
5. Selalu buat prefix yang unik untuk setiap tabel dalam database.
6. Deskripsi dari sebuah objek harus singkat dan jelas.
7. Selalu kreatif dan inovatif!!