

Layer, Style dan Tema pada Oracle Map Viewer

Yuafanda Kholfi Hartono

yuafanda@yahoo.com

<http://ilmukomputer.org/author/yofanda/>

Lisensi Dokumen:

Copyright © 2003-2015 IlmuKomputer.Com

Seluruh dokumen di IlmuKomputer.Com dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari IlmuKomputer.Com.

Tulisan ini membahas tentang aplikasi Oracle Map Viewer, sebuah aplikasi yang dapat mempresentasikan informasi yang diperoleh dari data untuk kemudian ditampilkan pada peta spasial sehingga menghasilkan informasi yang lebih deskriptif. Oracle Map Viewer memiliki lima macam layer yang menyusun peta secara utuh, yaitu layer dasar peta (Map Tile Layer(s)), Theme-Based FOI Layer(s), user-defined FOI atau Redline Layer(s), jendela informasi (Information Window Layer), dan Fixed Figures Layer.

Pendahuluan

Oracle Map Viewer adalah sebuah middleware yang digunakan untuk menampilkan informasi yang terhubung dengan peta serta data spasial. Aplikasi ini mampu merepresentasikan kumpulan data yang tersimpan di database menjadi tampilan yang mudah dipahami karena terintegrasi dengan peta.

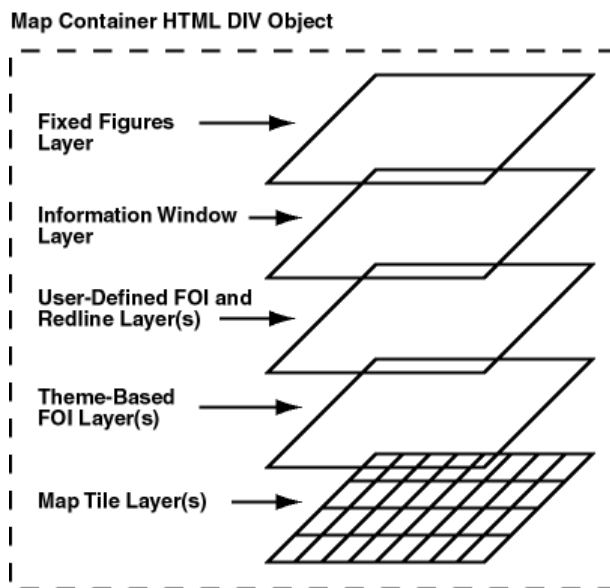
Ketika sebuah aplikasi menggunakan MapViewer, maka pada aplikasi tersebut berlaku style tertentu (seperti warna dan pola) dengan tema tertentu (yaitu, koleksi fitur spasial, seperti kota, sungai, dan jalan raya) untuk membuat peta (seperti gambar GIF untuk tampilan pada halaman Web). Sebagai contoh, dalam aplikasi tersebut mungkin menampilkan peta di mana taman nasional muncul dalam warna hijau dan restoran yang ditandai dengan bintang berwarna merah. Sebuah peta biasanya memiliki beberapa tema yang mewakili entitas politik atau fisik, atau keduanya. Misalnya, peta mungkin menunjukkan batas-batas negara, propinsi, kota, pegunungan, sungai, dan situs bersejarah. Ketika peta telah mengalami proses render, maka setiap tema akan direpresentasikan dalam satu layer pada sebuah gambar peta yang lengkap.

MapViewer memungkinkan Anda menentukan style, tema, dan peta dasar, termasuk aturan untuk menerapkan satu atau lebih style untuk setiap tema. Style, tema, peta dasar, dan aturan terkait tersebut akan disimpan dalam database dalam tabel definisi peta di bawah skema MDSYS, dan mereka terlihat oleh Anda melalui view metadata. Semua style dalam contoh

database dimiliki oleh semua pengguna. Metadata pemetaan (set style, tema, dan peta dasar) yang dapat Anda akses ditentukan oleh view metadata MapViewer (misalnya, USER_SDO_STYLES, USER_SDO_THEMES, dan USER_SDO_MAPS). Himpunan definisi objek peta dimana pengguna tertentu dapat mengakses biasanya disebut sebagai profil pemetaan pengguna. Anda juga dapat mengatur style, tema, dan peta dasar secara mandiri dengan tool Map Builder.

Isi

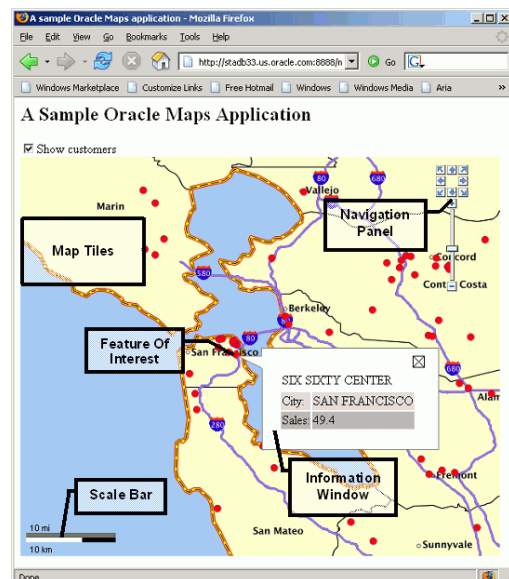
Bagaimana konten peta terbentuk dalam Oracle Map Viewer?



Seperti ditunjukkan pada gambar di samping, terdapat lima jenis konten yang berbeda dari lapisan peta pada Oracle Map Viewer : lapisan dasar peta (Map Tile Layer(s)), Theme-Based FOI Layer(s), user-defined FOI atau Redline Layer(s), jendela informasi (Information Window Layer), dan Fixed Figures Layer. Semua layer kecuali Fixed Figures Layer dipindahkan secara keseluruhan ketika pengguna menggambar sebuah peta. Lapisan-lapisan tersebut bergerak secara otomatis dan diperbarui oleh peta client ketika peta diseret atau diperbesar. (Fixed Figures Layer tidak pernah bergeser).

1. Lapisan Dasar Peta (Map Tile Layer)

Sebuah aplikasi Oracle Maps memiliki setidaknya satu lapisan dasar peta, yang tersusun dan ditampilkan pada gambar dasar peta sebelum tergenerate dari server peta dasar. Lapisan dasar peta menampilkan peta konten statis yang tidak terlalu sering berubah, dan layer tersebut biasanya digunakan sebagai latar belakang peta pada aplikasi klien. Sebagai contoh, diilustrasikan pada gambar di samping, laut, batas kabupaten, kota, dan jalan raya semuanya ditampilkan sebagai lapisan dasar peta. Hanya interaksi terbatas pengguna, seperti melakukan drag pada peta, dapat dilakukan di lapisan dasar peta.



Sebuah lapisan dasar peta biasanya berhubungan dengan peta dasar MapViewer, dan dikelola oleh server MapViewer. Namun, Anda dapat mengkonfigurasi lapisan dasar peta

dengan gambar cache dasar peta yang diproduksi oleh penyedia peta eksternal (non-MapView) .

Oracle Maps Client juga dapat menampilkan lapisan dasar built-in eksternal yang dihasilkan langsung oleh server eksternal, seperti Built-in Google Maps dan Microsoft Bing Maps based layer. Layer dasar peta selalu ditempatkan di bagian bawah hirarki layer. Layer ini menampilkan statis dan latar belakang isi peta. Ketika beberapa layer tersebut disertakan, mereka semua harus memiliki sistem koordinat yang sama dan definisi tingkat zoom.

Secara internal, lapisan peta dasar biasanya lebih besar dari ukuran peta DIV container window. Hal ini memungkinkan lapisan dasar tambahan akan diambil dan disimpan sementara oleh browser. Akibatnya, lapisan dasar ini akan segera terlihat ketika lapisan peta diseret memutar oleh pengguna.

2. Theme-Based FOI Layer(s)

Theme-Based FOI layer dapat terdiri dari satu atau lebih layer. Setiap Theme-Based FOI Layers terdiri dari kumpulan fois interaktif yang memenuhi kriteria pencarian tertentu yang telah ditetapkan dan didefinisikan dalam tema MapViewer. Fois bisa berupa titik, garis, atau poligon. Misalnya, semua toko dengan volume penjualan yang lebih besar dari \$ 100.000 dapat ditampilkan sebagai lapisan FOI berbasis tema titik. Pengguna dapat berinteraksi dengan fois dengan menggerakkan mouse di atas mereka atau mengklik pada fois tersebut. Aplikasi dapat menyesuaikan bagaimana peta client bereaksi terhadap interaksi pengguna tersebut.

Semua fitur (geografis dan non-geografis) dari lapisan FOI berdasarkan tema yang disimpan dalam database. Fitur akan diquery dan diberikan oleh server FOI ketika aplikasi client meminta mereka untuk menjalankannya. Jendela query untuk lapisan FOI berdasarkan tema-dapat disesuaikan untuk menjadi lebih besar dari peta jendela DIV, sehingga memberikan beberapa ruang tambahan untuk menyeret peta tanpa melakukan refresh lapisan FOI berdasarkan tema-dari server.

3. User-Defined FOI Layers

Sebuah layer user-defined FOI adalah fitur interaktif yang didefinisikan di sisi client. FOI bisa berupa titik, garis, atau fitur poligon. Pengguna dapat berinteraksi dengan user-defined FOI dengan cara yang sama dengan yang mereka dapat dengan tema berdasarkan FOI. Namun, berbeda dengan layer FOI berdasarkan tema yang diberikan sebagai kumpulan fitur, masing-masing user-defined FOI diminta dan diberikan secara individual. Semua atribut dari user-defined FOI, termasuk representasi geometri dan gaya rendering, harus disediakan oleh aplikasi. Misalnya, geometri rute berdasarkan yang ditentukan pengguna pada awal dan akhir alamat harus ditampilkan sebagai garis yang ditetapkan pengguna FOI pada peta.

4. Information Window Layer

Jendela informasi (Information Window) adalah jendela pop-up kecil yang menampilkan konten di peta. Semua jendela informasi, ketika ditampilkan, ditempatkan di dalam layer langsung di atas sebuah layer FOI yang ditetapkan pengguna atau beberapa layer.

5. Fixed Figures Layer

Lapisan paling atas yang mengandung gambar tetap, yang merupakan elemen tidak bergerak seperti catatan hak cipta, skala bar, panel navigasi, dan fitur peta dekorasi yang

ditetapkan pengguna. (Sebuah fitur peta dekorasi yang ditetapkan pengguna adalah aplikasi yang dapat mengandung konten HTML, seperti judul peta atau tombol kontrol) Layer gambar tetap ditampilkan di atas layer lainnya, dan tidak mengalami perpindahan ketika pengguna melakukan drag pada peta.

Konsep dasar dari style, tema (theme), base map, mapping metadata dan map pada Oracle Map Viewer.

- Style (gaya) mendefinisikan hasil render untuk fitur yang berhubungan dengan karakteristik. Sebagai contoh, style teks menentukan bagaimana fitur tersebut diberi label pada peta, sementara style garis menentukan penampakan dari fitur linear pada peta seperti jalan, serta fitur linear lainnya.
- Tema adalah kumpulan fitur (entitas dengan atribut spasial dan nonspasial) yang berkaitan dengan style melalui penggunaan aturan styling.
- Sebuah base map terdiri dari satu atau lebih tema.
- Mapping metadata (Pemetaan metadata) terdiri dari repositori style, tema, dan peta dasar yang disimpan dalam database.
- Map (Sebuah peta) adalah salah satu komponen dalam MapViewer yang menciptakan dan merespon permintaan atas sebuah peta. Peta bisa file gambar, representasi objek file gambar, atau URL yang mengacu ke file gambar.

Style

Style adalah atribut visual yang dapat digunakan untuk mewakili fitur spasial. Simbol-simbol dan label-label dari peta dasar untuk mewakili titik, garis, dan fitur area yang didefinisikan dan disimpan sebagai personal style. Setiap style memiliki nama yang unik untuk mendefinisikan satu atau lebih elemen grafis dalam sintaks XML. Style dapat merupakan salah satu jenis sebagai berikut :

- Warna : sebuah warna dari bagian dalam (fill) atau garis batas (stroke) suatu objek, atau keduanya.
- Marker (penanda) : Sebuah bentuk dengan warna fill dan stroke yang telah ditentukan, atau sebuah gambar. Marker seringkali berupa sebuah ikon untuk mewakili sebuah titik yang menggambarkan tempat seperti bandara, resort ski dan tempat bersejarah. Ketika style marker ditentukan untuk fitur garis (line), mesin rendering akan memilih titik yang cocok pada baris dan menerapkan style marker (misalnya, penanda perisai untuk US interstate highway) ke titik tersebut.
- Garis (line): Style garis (lebar, warna, gaya akhir, gaya penggabungan) dan opsional garis tengah, tepi, dan tanda pagar. Garis sering digunakan untuk fitur linear seperti jalan raya, sungai, pipa, dan jalur transmisi listrik. Anda juga dapat menggunakan gaya garis *cased*, yang berguna untuk menggambar jalan-jalan dan jalan raya.
- Area: warna atau tekstur, dan opsional pada warna stroke. Area sering digunakan untuk fitur poligonal seperti kabupaten dan risalah sensus
- Teks: spesifikasi font (ukuran dan bentuk) dan opsional format (bold, italic) dan warna latar depan. Teks sering digunakan untuk penjelasan dan pelabelan (seperti nama-nama kota dan sungai)
- Advanced: Gabungan style yang digunakan terutama untuk pemetaan tematik. Kunci dari advance style adalah BucketStyle, yang mendefinisikan hubungan antara satu set render style sederhana (dan pelabelan opsional) dan satu set bucket.

Untuk setiap fitur yang akan diplot, nilai yang ditunjuk atau ditetapkan adalah nilai dari fitur yang digunakan untuk menentukan bucket fitur jatuh ke dalam, dan kemudian gaya terkait dengan bucket yang digunakan untuk merencanakan fitur tersebut.

Style Warna

Sebuah style warna memiliki warna fill, warna stroke, atau keduanya. Ketika diterapkan pada sebuah bentuk atau geometri, warna fill (jika ada) digunakan untuk mengisi interior dengan warna tertentu, dan warna stroke (jika ada) digunakan untuk mewarnai batas-batas dari bentuk tersebut. Masing-masing warna (fill dan stroke) juga dapat memiliki nilai alpha, yang mengontrol transparansi warna.

Untuk style warna, atribut class dari `<g>` elemen harus diatur ke "color". Elemen `<g>` harus memiliki atribut style, yang menentukan komponen warna dan nilai alpha opsional mereka. Sebagai contoh:

- `<g class="color" style="fill:#ff0000">` menspesifikasikan sebuah style warna satu jenis warna fill (dengan nilai RGB #ff0000).
- `<g class="color" style="fill:#ff0000;stroke:blue">` menspesifikasikan sebuah style warna dengan satu jenis warna fill dan warna stroke (biru).

Anda dapat menentukan nilai warna baik menggunakan string heksadesimal (seperti # 00FF00) atau nama warna dari daftar berikut: hitam, biru, cyan, darkGray, abu-abu, hijau, lightGray, magenta, oranye, pink, merah, putih dan kuning.

Untuk menentukan transparansi pada style warna, Anda dapat menentukan nilai fill-opacity dan stroke-opacity dari 0 (benar-benar transparan) ke 255 (buram). Contoh berikut menetapkan komponen fill dengan setengah transparansi:

```
<g class="color" style="fill:#ff00ff;fill-opacity:128">
```

Contoh berikut menetapkan stroke dan fill-opacity:

```
<g class="color" style="stroke:red;stroke-opacity:70;
fill:#ff00aa;fill-opacity:129">
```

Sintaks untuk atribut style adalah string yang terdiri dari satu atau lebih nama: pasangan nilai yang dipisahkan oleh titik koma. (Sintaks dasar ini digunakan juga dalam jenis lain dari style). Anda juga dapat menentukan lebar warna stroke. Lebar Stroke default ketika menggambar batas suatu bentuk adalah 1 pixel. Untuk mengubah itu, tambahkan stroke-width: tentukan nilai untuk style atribut string. Contoh berikut menetapkan lebar stroke 3 piksel:

```
<g class="color" style="stroke:red;stroke-width:3">
```

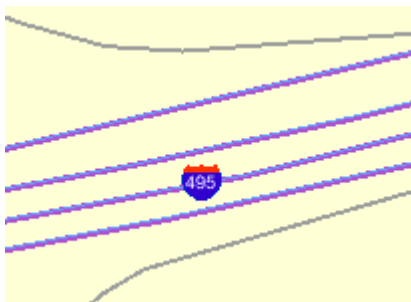
Style Marker

Sebuah style marker merupakan penanda untuk ditempatkan pada fitur titik atau pada titik-titik label daerah dan fitur linear. Sebuah marker dapat berupa penanda vektor atau penanda gambar raster. Sebuah marker juga dapat memiliki opsional notasi teks. Untuk penanda vektor, koordinat elemen vektor harus didefinisikan dalam dokumen XMLnya. Untuk penanda berdasarkan gambar raster, dokumen XML untuk style menunjukkan bahwa style ini didasarkan pada gambar eksternal.

Marker pada dokumen XML menentukan ukuran tampilan secara lebih detail : detil lebar dan tinggi didefinisikan oleh `width:value;height:value` yang terpasang dalam atribut `style` `<g>` elemen. Atribut `class` harus diatur ke "marker". Beberapa marker harus dilapis dengan beberapa notasi teks, seperti marker perisai US interstate highway, yang, ketika diberikan, juga harus memiliki sejumlah rute yang diplot di atasnya. Gaya untuk teks notasi tersebut adalah atribut `style` dengan satu atau lebih dari pasangan nama-nilai berikut: `font-family: value`, `font-style: value`, `font-size: value`, dan `font-weight: value`.

Contoh berikut mendefinisikan marker berbasis gambar yang menentukan atribut font atribut (ditampilkan dalam huruf tebal) untuk label teks yang dapat ditarik di atas marker:

```
<?xml version="1.0" standalone="yes"?>
<svg width="lin" height="lin">
<desc></desc>
<g class="marker"
style="width:20;height:18;font-family:sans-serif;font-size:9pt;fill:#f
fffff">
  <image x="0" y="0" width="9999" height="9999" type="gif"
href="dummy.gif"/>
</g>
</svg>
```



Dalam contoh gambar di samping, marker diterapkan pada fitur titik dengan label teks, label teks ditarik berpusat di atas marker, menggunakan jenis dan ukuran huruf yang spesifik, dan dengan warna fill (putih dalam kasus ini) sebagai latar depan teks. Label teks (495) pada gambar memiliki atribut teks spesifik.

Style Garis

Sebuah style garis hanya berlaku untuk fitur linier, seperti jalan, rel kereta api, atau batas daerah. Dengan kata lain, style garis dapat diterapkan hanya untuk grafik Oracle Spasial dan geometri dengan nilai `SDO_GTYPE` berakhir di 2 (jalur) atau 6 (multiline).

Ketika MapViewer menggambarkan fitur linear, style garis memberitahu mesin rendering tentang warna, pola dasbor, dan lebar stroke yang digunakan. Sebuah style garis dapat memiliki unsur garis dasar yang, jika didefinisikan, bertepatan dengan geometri linear aslinya. Hal ini juga dapat menentukan dua sisi sejajar dengan garis dasar. Elemen garis paralel dapat memiliki warna sendiri, pola dasbor, dan lebar stroke. Jika garis paralel yang digunakan, mereka harus berada untuk setiap sisi garis dasar, dengan offset sebesar garis dasar tersebut.

Untuk menggambar seperti garis kereta api, Anda perlu menentukan jenis ketiga elemen garis dalam style garis yang disebut hashmark. Untuk sebuah elemen `<baris>` kelas hashmark, nilai pertama dalam array dasbor menunjukkan gap antara dua tanda hash, dan nilai kedua menunjukkan panjang hash mark untuk kedua sisi garis. Contoh berikut mendefinisikan garis hash mark dengan jarak 8,5 unit layar dan panjang 3 unit layar di setiap sisi garis dasar:

```
<line class="hashmark" style="fill:#003333" dash="8.5,3.0"/>
```

Contoh berikut mendefinisikan style garis secara lengkap :

```
<?xml version="1.0" standalone="yes"?>
```



```
<svg width="1in" height="1in">
  <g class="line" style="fill:#ffff00;stroke-width:5">
    <line class="parallel" style="fill:#ff0000;stroke-width:1.0"/>
    <line class="base" style="fill:black;stroke-width:1.0"
dash="10.0,4.0"/>
  </g>
</svg>
```

Dalam contoh di atas, class = "line" mengidentifikasi gaya sebagai gaya garis. Warna mengisi keseluruhan (#ffff00) digunakan untuk mengisi ruang antara garis sejajar dan garis dasar. Garis lebar keseluruhan (5 piksel) membatasi lebar maksimum yang gaya bisa menempati (termasuk yang dari garis paralel).

Style garis dalam contoh sebelumnya memiliki kedua garis dasar dan elemen garis paralel. Unsur garis paralel (class = "paralel") didefinisikan oleh elemen pertama <line>, yang mendefinisikan warna dan lebar. (Karena definisi tidak menghasilkan pola dasbor, garis paralel atau tepi akan solid.) Elemen garis dasar (class = "base") didefinisikan oleh elemen kedua <line>, yang mendefinisikan warna, lebar, dan pola dasbor.

Sebuah marker (seperti penanda arah) dapat didefinisikan untuk style garis. Parameter `marker-name` menentukan nama style marker, parameter `marker-position` menentukan proporsi (0-1) dari jarak sepanjang garis dari titik awal di mana untuk menempatkan marker, dan parameter `marker-size` menentukan jumlah tampilan unit untuk ukuran marker. Orientasi marker mengikuti orientasi segmen garis dimana marker ditempatkan.

Contoh berikut mendefinisikan line style dengan penanda arah :

```
<?xml version="1.0" standalone="yes"?>
<svg width="1in" height="1in">
  <g class="line" style="fill:#33a9ff;stroke-width:4;
  marker-name:M.IMAGE105_BW; marker-position:0.15;
  marker-size=8; multiple-marker=true">
    <line class="parallel" style="fill:red;stroke-width:1.0"/>
  </g>
</svg>
```

Style Area

Style Area mendefinisikan pola yang akan digunakan untuk mengisi fitur area. Dalam rilis Oracle Map Viewer saat ini, style area harus berbasis gambar. Artinya, ketika Anda menerapkan style area untuk geometri, gambar mendefinisikan gaya yang diplot berulang-ulang sampai geometri terisi penuh. Definisi dari style area hampir sama dengan definisi style marker.

Contoh berikut mendefinisikan style area:

```
<?xml version="1.0" standalone="yes"?>
<svg width="1in" height="1in">
  <g class="area" style="stroke:#000000">
    <image/>
  </g>
</svg>
```

Dalam contoh sebelumnya, class = "area" mengidentifikasi gaya sebagai style area. Warna stroke (style = "Stroke: # 000000") adalah warna yang digunakan untuk menggambarkan batas geometri. Jika warna stroke tidak didefinisikan, geometri tidak akan memiliki garis batas yang terlihat, meskipun interiornya penuh dengan gambar pola.

Anda juga dapat menentukan style garis yang akan digunakan sebagai batas untuk style area. Berikut definisi style area menggunakan kata kunci style garis (ditampilkan dalam huruf tebal dalam contoh) dalam menentukan style garis yang akan digunakan untuk perbatasan fitur :

```
<?xml version="1.0" standalone="yes"?>
```

```
<svg width="1in" height="1in">
  <g class="area" style="line-style:L.DPH">
    <image x="0" y="0" width="9999" height="9999" type="gif"
href="dummy.gif"/>
  </g>
</svg>
```

Seperti style marker gambar, gambar untuk style daerah harus disimpan dalam kolom terpisah (diidentifikasi dalam kolom IMAGE dalam view metadata USER_SDO_STYLES dan ALL_SDO_STYLES)

Style Teks

Sebuah style teks mendefinisikan font dan warna yang akan digunakan dalam label fitur spasial. Kelas atribut harus memiliki nilai "text". Untuk font, Anda dapat menentukan style (polos, miring, dan sebagainya), family font, ukuran, dan lebar. Untuk menentukan warna bagian depan, Anda dapat menggunakan atribut fill.

Contoh berikut mendefinisikan style teks:

```
<?xml version="1.0" standalone="yes"?>
<svg width="1in" height="1in">
  <g class="text" style="font-style:plain; font-family:Dialog;
font-size:14pt;
font-weight:bold; fill:#0000ff">
    Hello World!
  </g>
</svg>
```

Dalam contoh di atas, teks "Hello World!" ditampilkan hanya saat style tersebut dipreview dalam style tool creator, seperti tool Map Builder. Ketika style diterapkan untuk peta, style tersebut akan selalu ditambahkan dengan label teks, dimana sebenarnya MapViewer memperolehnya dari tema.



Sebuah style teks dapat memberikan latar belakang putih di sekitar teks yang ditulis, untuk membuat label lebih mudah dibaca pada peta yang memiliki banyak fitur. Gambar di samping menunjukkan label Vallejo dengan latar belakang putih yang berada di sekitar pada huruf.

Untuk mendapatkan hasil yang ditunjukkan pada gambar di atas, Anda harus menentukan atribut float-width dalam elemen <g> dari definisi style teks. Contoh berikut menggunakan atribut float-width (dalam contoh ditampilkan dalam huruf tebal) untuk menentukan latar belakang putih yang membentang 3,5 piksel dari batas setiap huruf. (Teks Hello World! Diabaikan ketika gaya diterapkan pada tampilan label)

```
<?xml version="1.0" standalone="yes"?>
<svg width="1in" height="1in">
<desc></desc>
<g class="text" float-width="3.5"
style="font-style:plain; font-family:Dialog; font-size:12pt;
font-weight:bold;
fill:#000000">
    Hello World!
</g>
</svg>
```

Style Advance

Style Advance adalah style terstruktur yang terbuat dari style sederhana. Style Advance digunakan terutama untuk pemetaan tematik. Inti dari style advance adalah style bucket (BucketStyle), dan setiap style advance adalah bentuk dari bucket style. Sebuah bucket style

adalah pemetaan satu-ke-satu antara satu set style primitif dan satu set bucket. Setiap bucket berisi satu atau lebih nilai atribut dari fitur yang akan diplot. Untuk masing-masing fitur, salah satu atribut yang digunakan untuk menentukan bucket terkandung dalam fitur tersebut, dan kemudian style diberikan pada bucket yang diterapkan untuk fitur tersebut.

Dua jenis style bucket khusus juga disediakan: skema warna dan variabel marker. Gaya advanced lainnya adalah dot density, bar chart, collection, pie chart variabel, dan heat map.

Style Bucket

Sebuah style bucket mendefinisikan satu set bucket, dan menetapkan satu style primitif untuk setiap bucket. Isi bucket dapat berupa nilai-nilai seperti berikut :

Sebuah koleksi nilai diskrit (misalnya, bucket untuk semua kabupaten dengan kode risiko badai 1 atau 2, bucket untuk semua kabupaten dengan kode risiko badai 3, dan seterusnya).

Berbagai nilai kontinu (misalnya, bucket untuk semua kabupaten dengan pendapatan keluarga rata-rata kurang dari \$ 30.000, bucket untuk semua kabupaten dengan pendapatan rata-rata keluarga dari \$ 30.000 sampai dengan \$ 39.999, dan sebagainya). Dalam hal ini, rentang dari serangkaian bucket dapat didefinisikan secara individual (masing-masing ditentukan oleh nilai batas atas dan nilai batas bawah) atau sama dibagi dengan rentang induknya.

Kode berikut kutipan menunjukkan format dasar dari style bucket:

```
<?xml version="1.0" ?>
<AdvancedStyle>
  <BucketStyle>
    <Buckets>
      . . .
    </Buckets>
  </BucketStyle>
</AdvancedStyle>
```

Berbeda dengan corak style dasar yang lain, advance style selalu memiliki elemen root yang diidentifikasi oleh <AdvancedStyle> tag.

Untuk advance style, sebuah elemen <BucketStyle> adalah satu-satunya anak dari elemen <AdvancedStyle>. Setiap elemen <BucketStyle> memiliki satu atau lebih elemen-elemen anak <Buckets>, yang isinya bervariasi tergantung pada jenis bucket.

Style Skema Warna

Sebuah style skema warna secara otomatis menghasilkan style warna individu dari berbagai kecerahan untuk setiap bucket berdasarkan warna dasar. Tingkat kecerahan yang berada di antara kecerahan penuh dan kegelapan total. Biasanya, bucket pertama diperuntukkan pada bagian yang paling terang dari warna dasar dan bucket terakhir diperuntukkan pada bagian yang paling gelap.

Anda dapat menyertakan warna stroke untuk digunakan oleh style warna pada setiap bucket. Warna stroke bukan bagian dari perhitungan tingkat kecerahan. Jadi, misalnya, jika satu set fitur poligonal yang diberikan menggunakan style skema warna, interior setiap poligon diisi dengan warna (bentuk warna dasar) untuk setiap bucket yang sesuai, namun batas-batas dari semua poligon ditarik menggunakan warna stroke yang sama.

Anda dapat menyertakan nilai opacity (0 sampai 255, dari transparan sampai buram) untuk menjadi warna dasar (menggunakan atribut basecolor_opacity) dan untuk warna stroke (menggunakan atribut stroke_color_opacity).

Contoh berikut mendefinisikan style skema warna dengan warna stroke hitam dan empat bucket terkait dengan berbagai nuansa warna dasar biru.

```
<?xml version="1.0" ?>
<AdvancedStyle>
```

```
<ColorSchemeStyle basecolor="blue" strokecolor="black">
  <Buckets>
    <RangedBucket label="&lt;10" high="10"/>
    <RangedBucket label="10 - 20" low="10" high="20"/>
    <RangedBucket label="20 - 30" low="20" high="30"/>
    <RangedBucket label="&gt;=30" low="30"/>
  </Buckets>
</ColorSchemeStyle>
</AdvancedStyle>
```

Style Variabel Marker

Sebuah style variabel marker menghasilkan serangkaian style marker berbagai ukuran untuk setiap bucket. Anda menentukan jumlah bucket, ukuran awal (terkecil) marker, dan selisih ukuran antara dua marker berturut-turut.

Style variabel marker secara konseptual mirip dengan style skema warna di kedua bucket dasar pada variasi dari objek tersebut: dengan tingkat kecerahan style skema warna dari warna dasar yang bervariasi, dan dengan ukuran style variabel marker yang bervariasi.

Contoh berikut membuat style variabel marker dengan empat bucket, masing-masing terkait dengan ukuran yang berbeda (dengan penambahan sebesar 4) dari marker (m.circle). Marker untuk bucket pertama memiliki radius 10 unit display, marker untuk bucket kedua memiliki radius 14 unit display, dan sebagainya. Contoh berikut mengasumsikan bahwa penanda bernama m.circle telah ditetapkan.

```
<?xml version="1.0" ?>
<AdvancedStyle>
  <VariableMarkerStyle          basemarker="m.circle"          startsize="10"
  increment="4">
    <Buckets>
      <RangedBucket label="&lt;10" high="10"/>
      <RangedBucket label="10 - 20" low="10" high="20"/>
      <RangedBucket label="20 - 30" low="20" high="30"/>
      <RangedBucket label="&gt;=30" low="30"/>
    </Buckets>
  </VariableMarkerStyle>
</AdvancedStyle>
```

Style Dot Density Marker

Sebuah style dot density marker, bila diterapkan pada fitur daerah seperti negara atau kabupaten, secara acak akan menggambar satu set titik-titik dalam daerah tersebut. Jumlah titik ditarik dalam setiap daerah ditentukan oleh nilai count terkait dengan daerah tersebut. Ketika anda mendefinisikan style dot density (titik kepadatan) , Anda harus menentukan style marker yang akan digunakan untuk masing-masing titik.

Contoh berikut menampilkan definisi XML dari style dot density sederhana:

```
<?xml version="1.0" ?>
<AdvancedStyle>
  <DotDensityStyle          MarkerStyle="M.STAR"          DotWidth="8"
  DotHeight="8">
  </DotDensityStyle>
</AdvancedStyle>
```

Dalam contoh di atas, style marker M.STAR digunakan untuk setiap titik, dan ukuran masing-masing titik adalah 8 pixel lebar dan tinggi.

Ketika Anda menggunakan style dot density, Anda harus "menskala" nilai hitungan ke kisaran yang tepat. Sebagai contoh, jika Anda ingin menerapkan style titik kepadatan berdasarkan pada hitungan populasi untuk setiap kabupaten, Anda tidak akan ingin menggunakan penghitungan penduduk secara langsung (satu titik untuk setiap orang), karena ini akan menghasilkan jumlah

yang tidak dapat diterima dari titik-titik yang ditarik (misalnya, jika daerah memiliki 15.000 orang). Sebaliknya, tentukan nilai yang diperkecil skalanya, seperti populasi / 1000, ketika Anda menentukan aturan styling untuk tema. (MapView tidak melakukan penurunan skala internal apapun, sehingga Anda harus melakukannya di tingkat query SQL.)

Style Bar Chart Marker

Sebuah style bar chart marker mirip dengan style pie chart, kecuali bahwa ia menggambarkan grafik batang untuk setiap fitur yang menggunakannya. Contoh berikut menunjukkan definisi XML dari style bar chart:

```
<?xml version="1.0" ?>
<AdvancedStyle>
  <BarChartStyle width="30" height="25" show_x_axis="true">
    <Bar name="1990" color="#FF0000" />
    <Bar name="1995" color="#FFC800" />
    <Bar name="1998" color="#0000FF" />
    <Bar name="2000" color="#00FF00" />
    <Bar name="2002" color="#00FFFF" />
  </BarChartStyle>
</AdvancedStyle>
```

Dalam contoh di atas, lebar dan tinggi menentukan ukuran keseluruhan dari bar chart, termasuk semua individu bar di dalamnya.

Ketika bar chart digunakan pada fitur berdasarkan seperangkat nilai-nilai yang berhubungan dengan fitur itu, ketinggian setiap batang dapat ditentukan oleh salah satu dari dua pendekatan: skala lokal atau skala global. Sebuah skala lokal bar chart menentukan ketinggian setiap batang hanya dari nilai-nilai yang terkait untuk fitur tersebut; dan dengan demikian, misalnya, Anda tidak dapat membandingkan bar kedua dalam satu grafik ke bar kedua pada grafik lain dengan tema yang sama. Sebuah skala global bar chart menggunakan skala bar yang sama untuk semua grafik pada peta; dan dengan demikian, misalnya, Anda dapat membandingkan bar kedua dalam satu grafik ke bar kedua pada grafik lain dengan tema yang sama.

Jadi, jika Anda ingin membandingkan bar tidak hanya dalam grafik yang sama, tetapi juga di antara semua grafik yang ditunjukkan pada peta, Anda harus menggunakan style skala global bar chart dengan menentukan `share_scale = "true"` dalam definisi style bar chart, seperti yang ditunjukkan dalam contoh berikut:

```
<?xml version="1.0" ?>
<AdvancedStyle>
  <BarChartStyle width="40" height="30" share_scale="true"
    min_value="0.0" max_value="100">
    <Bar name="1990" color="#FF0000" />
    <Bar name="1995" color="#FFC800" />
    <Bar name="1998" color="#0000FF" />
    <Bar name="2000" color="#00FF00" />
    <Bar name="2002" color="#00FFFF" />
  </BarChartStyle>
</AdvancedStyle>
```

Ketika style bar chart dalam contoh di atas diterapkan pada tema, MapViewer mempertimbangkan berbagai nilai global dari semua fitur dalam tema itu, dan kemudian menentukan ketinggian setiap bar berdasarkan nilai tertentu mana dalam kisaran global dari nilai minimum ke nilai maksimum.

Style Collection

Sebuah style advance collection hanyalah kumpulan jenis gaya yang diterapkan bersama-sama

untuk sebuah fitur. Hal ini dapat mengakibatkan proses render dari tema collection lebih cepat dibandingkan dengan menggunakan beberapa tema berdasarkan style yang berbeda.

Sebagai contoh, sebuah style bar chart, bila diterapkan ke tampilan sebuah daerah, akan menampilkan hanya bar chart di satu area di dalam daerah, tetapi daerah itu sendiri (batas dan wilayah interior) tidak ditampilkan. Namun, Anda mungkin ingin melihat batas-batas dari kabupaten, untuk melihat bar chart mana yang menjadi milik daerah yang mana. Untuk melakukan hal ini tanpa style collection, Anda harus menentukan tema kedua di mana setiap daerah akan dikaitkan dengan style warna atau daerah. Pendekatan ini akan menghasilkan dua kali proses render (karena dua tema yang terlibat) untuk menghasilkan kelompok fitur yang sama.

Namun, dengan menggunakan style collection dalam contoh ini, Anda dapat menetapkan satu style yang mengacu pada kedua bar chart dan style warna atau daerah, dan kemudian menerapkan style collection dengan tema untuk kabupaten. Tema ini, ketika diberikan oleh MapViewer, akan menampilkan grafik bar dengan batas-batas pada peta.

Contoh penggunaan lain dari style collection adalah untuk rendering koleksi jenis fitur topologi, yang masing-masing dapat berisi beberapa jenis geometri, seperti poligon (area), poin, dan garis. Dalam kasus tersebut, style collection dapat mencakup gaya yang paling tepat untuk setiap jenis geometri di fitur koleksi topologi.

Contoh berikut menunjukkan definisi XML dari style collection:

```
<?xml version="1.0" standalone="yes"?>
<AdvancedStyle>
  <CollectionStyle>
    <style name="C.COUNTIES" shape="polygon" />
    <style name="L.PH" shape="line" />
    <style name="M.CIRCLE" shape="point" />
  </CollectionStyle>
</AdvancedStyle>
```

Style Variabel Pie Chart

Sebuah variabel pie chart variabel menghasilkan serangkaian lingkaran pie dari berbagai ukuran untuk setiap bucket. Anda yang menentukan informasi pie slice, ukuran jari-jari awal (terkecil) untuk lingkaran pie, dan ukuran kenaikan radius antara dua lingkaran secara berturut-turut.

Style variabel pie chart secara konseptual mirip dengan style marker variabel. Dengan style marker variabel ukuran marker dasar bervariasi, sedangkan dengan gaya pie chart variabel radius lingkaran bervariasi.

Contoh berikut membuat definisi untuk style variabel pie chart dengan empat bucket, masing-masing terkait dengan ukuran yang berbeda (dengan penambahan sebesar 4) dari lingkaran dengan jari-jari awal 5. Jari-jari lingkaran untuk bucket pertama memiliki radius 5 unit display, lingkaran untuk bucket kedua memiliki radius 9 unit display, dan sebagainya.

```
<?xml version="1.0" ?>
<AdvancedStyle>
  <VariablePieChartStyle startradius="5" increment="4">
    <PieSlice name="WHITE" color="#FFFFFF"/>
    <PieSlice name="BLACK" color="#000000"/>
    <PieSlice name="HISPANIC" color="#FF0000"/>
  <Buckets>
    <RangedBucket seq="0" label="0 - 6194757.2" low="0" high="6194757.2" />
  </Buckets>
  <RangedBucket seq="1" label="6194757.2 - 1.23895144E7" low="6194757.2" high="1.23895144E7"/>
  <RangedBucket seq="2" label="1.23895144E7 - 1.85842716E7" low="1.23895144E7" high="1.85842716E7"/>
</AdvancedStyle>
```

```

    <RangedBucket seq="3" label="1.85842716E7 - 2.47790288E7"
low="1.85842716E7" high="2.47790288E7"/>
    <RangedBucket seq="4" label="2.47790288E7 - 3.0973786E7"
low="2.47790288E7" high="3.0973786E7"/>
  </Buckets>
</VariablePieChartStyle>
</AdvancedStyle>

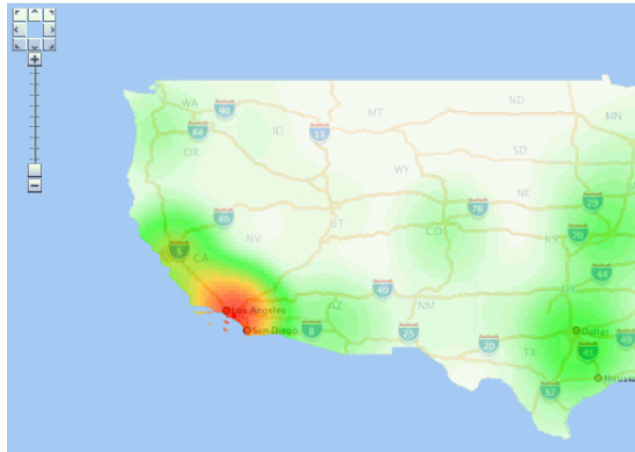
```

Style Heat Map

Sebuah style heat map dapat digunakan untuk menghasilkan warna peta dua dimensi (2D) dari setiap jenis poin data set. Warna-warna akan mewakili kepadatan distribusi atau pola poin atau peristiwa di seluruh wilayah. Secara internal, MapViewer menciptakan matriks 2D dan memberikan nilai untuk setiap sel grid yang didasarkan pada hasil dari jarak-tertimbang algoritma yang dijalankan terhadap poin data set.

Anda dapat membuat style heat map menggunakan tool Map Builder, dan menetapkannya sebagai gaya render untuk tema geometri jenis-titik. Anda kemudian dapat menambahkan

tema ini untuk peta dasar, atau menambahkannya sebagai FOI lapisan berdasarkan tema ke aplikasi interaktif Oracle Maps. Gambar di bawah menunjukkan peta yang ditampilkan menggunakan tema berdasarkan style heat map. Peta di samping menunjukkan konsentrasi restoran pizza: daerah merah memiliki konsentrasi tertinggi restoran pizza, dengan konsentrasi semakin rendah untuk oranye, kuning, hijau tua, hijau terang, pucat daerah hijau, dan putih.



Menentukan style label untuk sebuah bucket

Untuk style bucket berbasis koleksi dan individu style bucket berdasarkan jarak, Anda dapat menentukan gaya pelabelan dengan menggunakan atribut label_style di setiap elemen bucket. Contoh 2-3 menciptakan gaya bernama canggih V.RB1 di mana setiap kotak diberi label style teks (menggunakan atribut label_style), dengan beberapa style yang digunakan untuk beberapa bucket.

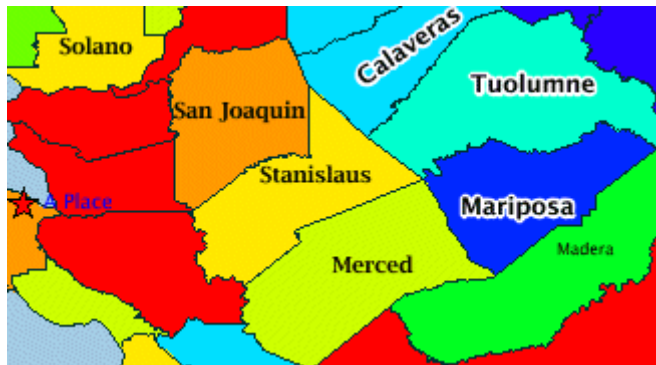
```

<?xml version="1.0" ?>
<AdvancedStyle>
  <BucketStyle>
    <Buckets>
      <RangedBucket seq="0" label="10k or less" high="10000"
style="c.rb13_1" label_style="T.AIRPORT NAME"/>
      <RangedBucket seq="1" label="10k - 20k" low="10000" high="20000"
style="c.rb13_2" label_style="T.CITY NAME"/>
      <RangedBucket seq="2" label="20k - 30k" low="20000" high="30000"
style="c.rb13_3" label_style="T.CITY NAME"/>
      <RangedBucket seq="4" label="30k - 40k" low="30000" high="40000"
style="c.rb13_4" label_style="T.CITY NAME"/>
      <RangedBucket seq="5" label="40k - 50k" low="40000" high="50000"
style="c.rb13_5" label_style="T.CITY NAME"/>
      <RangedBucket seq="6" label="50k - 75k" low="50000" high="75000"
style="c.rb13_6" label_style="T.ROAD NAME"/>
      <RangedBucket seq="7" label="75k - 100k" low="75000" high="100000"

```

```
        style="c.rb13_7" label_style="T.PARK NAME"/>
    <RangedBucket seq="8" label="100k - 125k" low="100000"
high="125000"
        style="c.rb13_8" label_style="T.RED STREET"/>
    <RangedBucket seq="9" label="125k - 250k" low="125000"
high="250000"
        style="c.rb13_9" label_style="T.ROAD NAME"/>
    <RangedBucket seq="10" label="250k - 450k" low="250000"
high="450000"
        style="c.rb13_10" label_style="T.ROAD NAME"/>
    <RangedBucket seq="11" label="450k - 650k" low="450000"
high="650000"
        style="c.rb13_11" label_style="T.ROAD NAME"/>
    <RangedBucket seq="12" label="650k up" low="650000"
style="c.rb13_13"/>
</Buckets>
</BucketStyle>
</AdvancedStyle>
```

Untuk individual bucket berdasarkan jarak, nilai yang lebih rendah-terikat inklusif, sedangkan nilai yang lebih tinggi-terikat eksklusif (kecuali untuk rentang yang memiliki nilai lebih besar



dari nilai apapun dalam rentang lainnya; nilai atas-terikat adalah inklusif). Tidak ada rentang yang diperbolehkan untuk memiliki rentang nilai yang tumpang tindih dalam rentang lainnya.

Berikut akan ditampilkan peta yang mungkin terlihat seperti tampilan pada Gambar di samping, di mana nama-nama kabupaten ditunjukkan dengan label yang mencerminkan berbagai gaya teks (dalam hal ini

kasus tergantung pada jumlah penduduk di kabupaten itu).

Mendapatkan Gambar Sampel dari Setiap Style

Untuk mendapatkan gambar sampel untuk didefinisi awal dari setiap style yang disimpan dalam database, Anda dapat mengeluarkan permintaan HTTP sederhana untuk server MapViewer. Permintaan ini dapat menentukan ukuran gambar sampel, warna latar belakang, dan format gambar. Permintaan tersebut berguna jika Anda ingin menampilkan daftar visual style pada halaman Web, untuk membuat legenda peta yang dapat diedit, atau hanya untuk melihat bagaimana berbagai style akan muncul. Permintaan HTTP memiliki parameter berikut, yang semuanya opsional kecuali sty:

- sty (diperlukan) menentukan nama style.
- ds menentukan sumber data mana style dapat diakses. Secara default, sumber data MapViewer default adalah yang digunakan.
- w menentukan lebar sampel gambar dalam piksel. Nilai defaultnya adalah 20.
- h menentukan ketinggian sampel gambar dalam piksel. Nilai defaultnya adalah 20.
- f menentukan format gambar sampel. Nilai yang tersedia adalah png untuk streaming langsung gambar PNG, png_url untuk URL dari gambar PNG, gif untuk streaming gambar GIF secara langsung, atau gif_url untuk URL dari gambar GIF. Nilai default

png, yang berarti server MapViewer akan langsung mengalirkan data gambar PNG yang dihasilkan kembali ke client tanpa terlebih dahulu menyimpannya ke disk server.

- bg menentukan warna latar belakang gambar sampel. Format harus string heksadesimal dalam bentuk 0xrrggbb, seperti 0x808080 untuk warna abu-abu. Nilai default adalah 0xffffffff (putih).

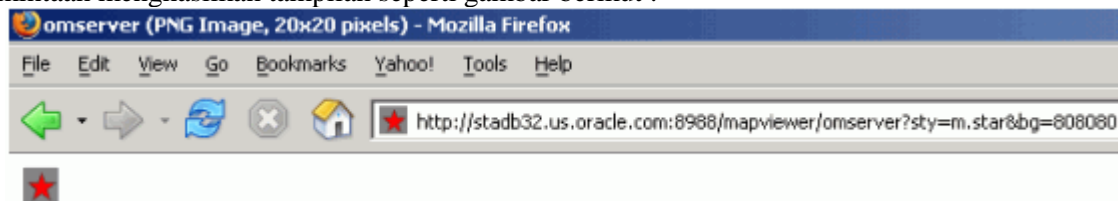
Untuk latar belakang transparan, tentukan bg sebagai string heksadesimal diperpanjang untuk memasukkan nilai-nilai alpha, dalam format 0xaarrggbb. Misalnya, 0x00ffffff akan membuat latar belakang gaya gambar itu benar-benar transparan, sementara 0x55ffffff adalah latar belakang putih dengan nilai transparansi 0x55 (nilai desimal 80). Nilai alpha dapat berkisar dari 0x00 (benar-benar transparan) ke 0xff (benar-benar buram).

- aa menentukan apakah gambar sampel harus diberikan dalam mode antialiasing. Nilai default adalah string yang benar. Tentukan string false jika Anda tidak ingin menggunakan antialiasing.

Contoh berikut menghasilkan PNG antialiased dengan latar belakang abu-abu dengan ukuran default 20x20 pixel, menampilkan gaya marker bernama M.STAR dari MapViewer sumber data default:

<http://www.mycorp.com/mapviewer/omserver?sty=m.star&bg=808080>

Permintaan menghasilkan tampilan seperti gambar berikut :



Mengatur Orientasi Style Teks

Untuk mengarahkan label teks suatu titik dalam arah sebuah orientasi vektor, Anda dapat menentukan suatu titik sebagai titik Oracle Spasial berorientasi pada permintaan pada peta. Ketika MapViewer memberi label titik orientasi, maka secara otomatis Pusat label teks pada posisi titik, dan menyelaraskan label agar mengarah ke arah vektor orientasi.

Untuk setiap fitur yang akan diberi label, Anda harus menentukan lokasi untuk titik orientasi. Anda dapat mengelompokkan titik-titik berorientasi di satu tabel dan membuat indeks spasial pada kolom yang berisi geometri titik. Anda kemudian dapat membuat tema berdasarkan atas tabel tersebut, menentukan gaya teks yang diinginkan sebagai label, dan menentukan gaya warna transparan sebagai gaya render sehingga titik tersebut tidak ditampilkan pada peta.

Contoh 2-4

```
<map_request
  title="Labeling Oriented Points"
  datasource="my_datasource" width="400" height="300"
  antialias="true"
  format="PNG_STREAM">

<themes>
  <theme name="themel">
    <jdbc_query
      spatial_column="geom" jdbc_srid="8265"
      render_style="transparent_color"
      label_column="label" label_style="t.street name"
```

```

datasource="tilsmenv">
SELECT SDO_GEOMETRY(2001, 8265, NULL,
                SDO_ELEM_INFO_ARRAY(1, 1, 1, 3, 1, 0),
                SDO_ORDINATE_ARRAY(12, 14, .3, .2))
geom, 'Oriented Point' label FROM dual
</jdbc_query>
</theme>
</themes>

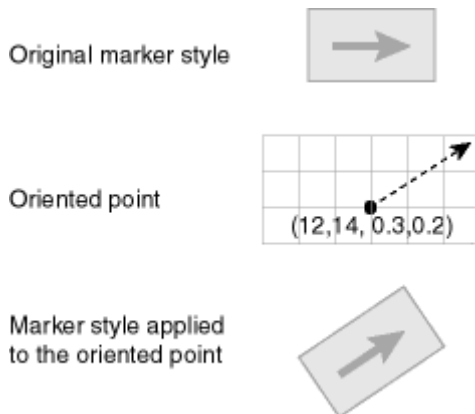
<styles>
<style name="transparent_color">
<svg width="lin" height="lin">
<g class="color" style="stroke:#ff0000;stroke-opacity:0">
<rect width="50" height="50"/>
</g>
</svg>
</style>
</styles>
</map_request>

```

Contoh berikut adalah permintaan peta dengan label satu titik berorientasi pada koordinat (12,14, 0.3,0.2), di mana (12,14) merupakan X dan Y koordinat titik dan (0.3,0.2) merupakan orientasi vektor. Hal itu membuat titik menggunakan gaya warna didefinisikan secara dinamis transparan (bernama transparent_color) untuk memastikan bahwa teks ditampilkan tetapi titik yang mendasari tidak ditampilkan, terlihat pada gambar di samping.



Mengatur Orientasi Style Marker



Ketika style marker diterapkan ke titik orientasi, MapViewer otomatis memutar style marker agar mengarah ke vektor orientasi. Setiap rotasi diperlukan oleh style marker akan berada di sekitar pusat marker.

Gambar di samping menunjukkan bagaimana Anda dapat menggunakan jalur berorientasi untuk mengontrol orientasi style marker. Dalam gambar ini, style marker asli pertama kali ditampilkan tanpa rotasi apapun. Namun, ketika marker diterapkan ke titik berorientasi sama ditunjukkan pada Contoh berikutnya, style marker diputar sesuai (dalam hal ini sekitar 34 derajat berlawanan jarum jam) untuk

mencerminkan vektor orientasi.

Tema

Tema adalah konsep yang paling penting dalam MapViewer. Tema adalah representasi visual dari lapisan data tertentu. Secara konseptual, tema adalah kumpulan fitur geografis yang berbagi atribut yang sama, ditambah rendering dan pelabelan aturan yang memberitahu MapViewer tentang style yang digunakan untuk membuat dan memberi label fitur. Agar lebih tepat, ketika Anda menentukan tema, Anda benar-benar memberikan MapViewer dengan informasi berikut: di mana dan bagaimana untuk mendapatkan data, dan bagaimana untuk membuat dan memberi

label pada data.

Tergantung pada bagaimana tema yang dibuat, hal tersebut juga dapat dikategorikan sebagai tema yang telah ditentukan atau tema (JDBC) dinamis. Untuk tema yang telah ditentukan, definisi tema ditentukan dalam tool Map Builder standalone dan disimpan dalam database. Untuk tema yang dinamis, definisi tema ini (XML) diciptakan secara real time oleh aplikasi. Tema dinamis biasanya menggunakan query SQL yang dijalankan oleh aplikasi untuk mendapatkan data.

Biasanya, data untuk tema berasal dari tabel spasial diaktifkan, yaitu, tabel database atau view dengan kolom jenis SDO_GEOMETRY. Misalnya, tema dengan nama US_STATES mungkin didasarkan pada tabel STATES yang memiliki kolom bernama GEOMETRY, ditambah kolom atribut nonspatial lainnya. Jenis tema sering disebut tema geometri. Selain data geometris, jenis data geografis database yang dikelola dapat dikaitkan dengan jenis yang sesuai tema; sebagai contoh:

- Gambar georeferensi yang disimpan dalam BLOB (tema gambar)
- Data Oracle Spasial GeoRaster (tema GeoRaster)
- Oracle data model spasial jaringan (tema jaringan)
- Oracle data model spasial topologi (tema topologi)
- Teks kartografi anotasi (tema anotasi)

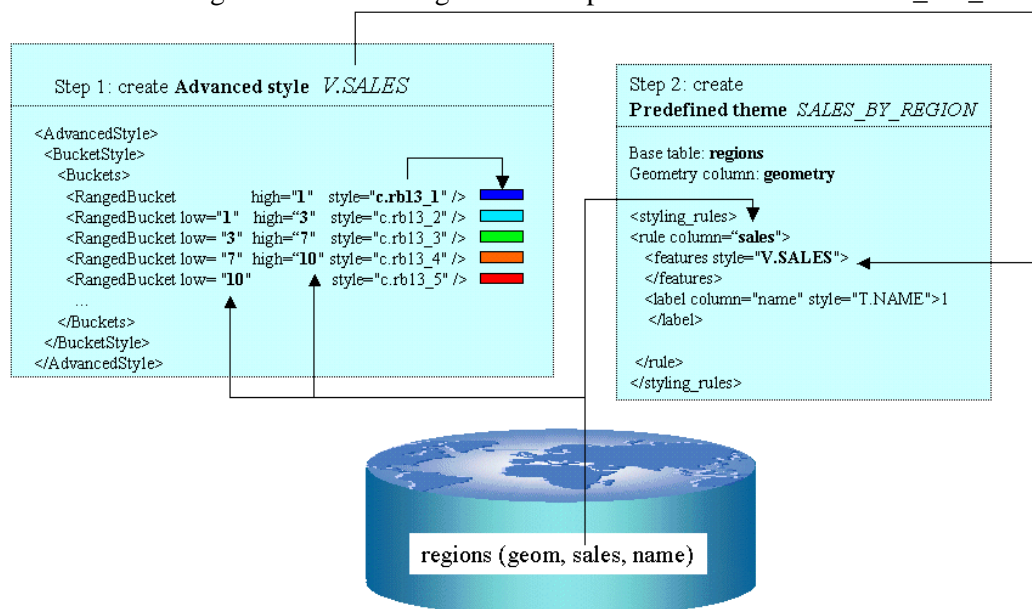
Tema MapViewer dapat digunakan untuk membuat tidak hanya data geografis yang tersimpan dalam database, tetapi juga data yang berasal dari sumber lain, seperti layanan Web (WFS dan WMS) atau sistem file lokal (melalui spasial antarmuka penyedia data). Terlepas dari jenis data apa yang terkait dengan tema (kecuali untuk tema WMS, yang mewakili lapisan peta eksternal terrender), pengaturan styling MapViewer styling masih perlu didefinisikan untuk setiap tema, dan style yang direferensikan oleh aturan styling harus ada dan disimpan dalam database sebagai bagian dari metadata pemetaan

Salah satu jenis tema dalam Oracle Map Viewer adalah pemetaan tematik (thematic mapping). Pemetaan tematik mengacu pada gambar fitur spasial berdasarkan nilai atribut mereka. MapViewer menggunakan pemetaan tematik untuk membuat peta di mana warna atau simbol yang digunakan untuk fitur untuk menunjukkan atribut mereka. Misalnya, tema Counties dapat ditarik menggunakan warna dengan warna yang berbeda yang memetakan langsung ke kepadatan penduduk masing-masing kabupaten, atau Gempa tema dapat diplot dengan lingkaran penuh yang ukuran peta dengan skala atau kerusakan setiap gempa.

Untuk membuat pemetaan tematik, Anda harus terlebih dahulu membuat style advance yang cocok untuk jenis peta tematik, dan kemudian membuat tema untuk fitur menentukan style advance sebagai gaya rendering. Dalam aturan styling untuk tema, Anda juga harus menentukan atribut kolom di tabel atau view yang nilai akan digunakan untuk menentukan dengan tepat bagaimana fitur akan diberikan tematis oleh style advance.

Sebagai contoh, asumsikan bahwa Anda ingin menampilkan peta di mana warna yang digunakan untuk masing-masing daerah mencerminkan tingkat penjualan untuk produk tertentu. Untuk melakukan ini, Anda harus membuat style advance yang mendefinisikan serangkaian bucket individu berdasarkan jarak, di mana setiap bucket berisi berbagai standar dari nilai penjualan untuk produk, dan setiap bucket memiliki gaya render terkait. (Masing-masing daerah akan diberikan dengan menggunakan style yang terkait dengan rentang di mana nilai penjualan di wilayah yang rendah tingkat penjualannya) Juga menentukan nama kolom atau kolom yang memberikan nilai atribut untuk diperiksa terhadap rentangnya. Dengan kata lain, style advance menentukan cara untuk memetakan daerah berdasarkan nilai penjualan mereka, dan styling aturan tema yang mengikat bersama style advance dan kolom atribut yang berisi nilai-nilai penjualan aktual.

Gambar berikut menunjukkan hubungan antara style advance dan tema, dan bagaimana style dan tema berhubungan dengan tabel dasar. Dalam gambar ini, gaya canggih bernama V.SALES mendefinisikan serangkaian bucket. Yang telah ditetapkan tema bernama SALES_BY_REGION



ditentukan gaya V.SALES dalam aturan styling. Tema juga mengidentifikasi kolom SALES dalam tabel REGIONS sebagai kolom yang nilainya harus dibandingkan dengan bucket berkisar dalam style. (Setiap bucket bisa dikaitkan dengan style label di samping atau bukan style rendering) Sebagai bagian dari metadata pemetaan.

Penutup

Layer, Style dan Tema adalah bagian konsep penting dari aplikasi Oracle Maps Viewer. Dengan mempelajari bagian-bagian layer pada Oracle Maps Viewer, Anda dapat lebih memahami komposisi serta konten yang ada pada tampilan suatu peta. Dengan mengetahui karakteristik tiap-tiap style, Anda akan dapat menggunakannya secara tepat dalam tampilan peta yang Anda susun. Sedangkan Tema seringkali bersinergi dengan style membentuk satu kesatuan tampilan yang dapat mendukung detail suatu peta terlihat lebih maksimal. Diperlukan referensi-referensi yang lebih mengupas detil dari aplikasi seperti Oracle Maps Viewer sehingga pengguna dapat menggunakannya secara lebih maksimal.

Referensi

Oracle, "How Map Content Is Organized", 2015, Oracle Corp.

http://docs.oracle.com/cd/E12839_01/web.1111/e10145/vis_omaps.htm#JIMPV9786

Oracle, "Text Styles", 2015, Oracle Corp.

http://docs.oracle.com/cd/E12839_01/web.1111/e10145/vis_xml_styles.htm#JIMPV9838

Oracle, "Thematic Mapping", 2015, Oracle Corp.

http://docs.oracle.com/cd/E12839_01/web.1111/e10145/vis_concepts.htm#JIMPV9563

Oracle, " XML Format for Styles, Themes, Base Maps, and Map Tile Layers", 2015, Oracle Corp.

http://docs.oracle.com/cd/E12839_01/web.1111/e10145/vis_xml_styles.htm#JIMPV9000

Biografi Penulis



Yuafanda Kholfi Hartono. Menyelesaikan S1 di Universitas Indonesia. Sebelumnya menamatkan pendidikan di Sekolah Tinggi Akuntansi Negara, minat dan ketertarikan pada bidang IT membuat pada akhirnya ditempatkan pada Direktorat Jenderal Bea dan Cukai Kementerian Keuangan sebagai Database Administrator (DBA). Pengembangan Executive Information System yang menggunakan tools Oracle Map Viewer sebagai pendukung tampilan membuatnya mempelajari lebih detail mengenai aplikasi tersebut.