

REFERENSI CEPAT XPATH

Axis, predikat, fungsi, operator, pemilihan node

Sintaks

| | |
|------------------------------|-----------------------------------|
| Ekspresi Path | |
| <code>/</code> | Node akar (awal path absolut) |
| <code>/bookstore/book</code> | Pemilihan anak langsung |
| <code>//book</code> | Pilih semua node book di mana pun |
| <code>.</code> | Node konteks saat ini |
| <code>..</code> | Induk dari node saat ini |
| <code>@lang</code> | Atribut bernama lang |
| <code>node()</code> | Node apa pun dari tipe apa pun |
| <code>*</code> | Node elemen apa pun |
| <code>@*</code> | Atribut apa pun |

Contoh Dasar

```
<html/body/div # path absolut ke <div>
//input[@type='text'] # semua input teks
//div[@class='main']/* # anak-anak div.main
//a/@href # semua atribut href
```

Menggabungkan Path

```
//book/title | //book/price # gabungan dua path
//h1 | //h2 | //h3 # beberapa tipe elemen
```

Axis

| | |
|-----------------------------------|--|
| Arah Axis | |
| <code>child::</code> | Anak langsung (axis default) |
| <code>parent::</code> | Induk langsung |
| <code>ancestor::</code> | Semua leluhur hingga akar |
| <code>ancestor-or-self::</code> | Leluhur + node saat ini |
| <code>descendant::</code> | Semua keturunan |
| <code>descendant-or-self::</code> | Keturunan + node saat ini |
| <code>following::</code> | Semua node setelah node saat ini dalam dokumen |
| <code>following-sibling::</code> | Saudara setelah node saat ini |
| <code>preceding::</code> | Semua node sebelum node saat ini dalam dokumen |
| <code>preceding-sibling::</code> | Saudara sebelum node saat ini |
| <code>self::</code> | Hanya node saat ini |
| <code>attribute::</code> | Atribut dari node saat ini |
| <code>namespace::</code> | Node namespace |

Contoh Axis

```
//div/child::p # anak <p> dari <div>
//td/parent::tr # induk <tr> dari <td>
//h2/following-sibling::p # <p> setelah <h2>
//li/ancestor::ul # <ul> yang berisi <li>
```

Predikat

Pemfilteran dengan Predikat

```
//book[1] # elemen book pertama
//book[last()] # elemen book terakhir
//book[position() < 3] # dua book pertama
//book[@lang='en'] # book dengan lang="en"
//book[price > 30] # book dengan harga > 30
```

Pola Predikat

| | |
|-------------------------------|------------------------------|
| <code>[n]</code> | Elemen di posisi n (basis 1) |
| <code>[last()]</code> | Elemen terakhir |
| <code>[last()-1]</code> | Kedua dari terakhir |
| <code>[@attr]</code> | Memiliki atribut |
| <code>[@attr='val']</code> | Atribut sama dengan nilai |
| <code>[element]</code> | Memiliki elemen anak |
| <code>[element='text']</code> | Elemen anak berisi teks |
| <code>[not(@attr)]</code> | Tidak memiliki atribut |

Predikat Berantai

```
//div[@class='list']/a[1] # <a> pertama dalam div.list
//input[@type='text'][@name='q'] # kondisi AND
//book[price>30][@lang='en'] # beberapa kondisi
```

Fungsi

Fungsi String

| | |
|-------------------------------------|----------------------------------|
| <code>contains(s, sub)</code> | Benar jika s mengandung sub |
| <code>starts-with(s, pre)</code> | Benar jika s dimulai dengan pre |
| <code>string-length(s)</code> | Panjang string |
| <code>normalize-space(s)</code> | Pangkas dan rapat-padatkan spasi |
| <code>concat(a, b, ...)</code> | Gabungkan string |
| <code>substring(s, pos, len)</code> | Ambil substring (basis 1) |
| <code>translate(s, from, to)</code> | Ganti karakter per karakter |

Fungsi Numerik

| | |
|------------------------------|------------------------------|
| <code>sum(node-set)</code> | Jumlah nilai numerik |
| <code>count(node-set)</code> | Jumlah node |
| <code>floor(n)</code> | Bulatkan ke bawah |
| <code>ceiling(n)</code> | Bulatkan ke atas |
| <code>round(n)</code> | Bulatkan ke integer terdekat |
| <code>number(val)</code> | Konversi ke angka |

Contoh Fungsi

```
//div[contains(@class, 'active')]
//a[starts-with(@href, 'https')]
//p[string-length(text()) > 100]
//ul[count(li) > 5]
```

Operator

Operator Perbandingan

| | |
|--------------------|------------------------------|
| <code>=</code> | Sama dengan |
| <code>!=</code> | Tidak sama dengan |
| <code><</code> | Kurang dari |
| <code><=</code> | Kurang dari atau sama dengan |
| <code>></code> | Lebih dari |
| <code>>=</code> | Lebih dari atau sama dengan |

Logika dan Aritmatika

| | |
|------------------|------------|
| <code>and</code> | AND logika |
| <code>or</code> | OR logika |

not()

| | |
|------------------|-------------------|
| <code>+</code> | Penjumlahan |
| <code>-</code> | Pengurangan |
| <code>*</code> | Perkalian |
| <code>div</code> | Pembagian |
| <code>mod</code> | Modulus |
| <code> </code> | Gabungan node set |

Contoh Operator

```
//book[price > 20 and price < 50]
//item[@type='a' or @type='b']
//span[not(contains(@class, 'hidden'))]
```

Tes Node

Jenis Node

| | |
|---------------------------------------|---|
| <code>node()</code> | Node apa pun (elemen, teks, komentar, PI) |
| <code>text()</code> | Hanya node teks |
| <code>comment()</code> | Hanya node komentar |
| <code>processing-instruction()</code> | Node instruksi pemrosesan |
| <code>*</code> | Node elemen apa pun |
| <code>@*</code> | Node atribut apa pun |
| <code>element-name</code> | Elemen dengan nama spesifik |

Contoh Tes Node

```
//p/text() # konten teks dari <p>
//div/comment() # komentar dalam <div>
//body/node() # semua node anak dari <body>
//div/* # semua anak elemen dari <div>
```

Fungsi Boolean

| | |
|----------------------------|----------------------------|
| <code>true()</code> | Boolean benar |
| <code>false()</code> | Boolean salah |
| <code>boolean(expr)</code> | Konversi ke boolean |
| <code>not(expr)</code> | Negasi boolean |
| <code>lang(code)</code> | Benar jika lang node cocok |

Singkatan

Bentuk Pendek vs Panjang

| | |
|--------------------|------------------------------|
| (tidak ada) | child:: (axis default) |
| @ | attribute:: |
| // | /descendant-or-self::node()/ |
| . | self::node() |
| .. | parent::node() |
| [n] | position()=n |

Contoh Singkatan

```
# Pasangan-pasangan ini setara:
child::div → div
attribute::href → @href
/descendant-or-self::node()/p → //p
self::node() → .
parent::node() → ..
```

Pola Singkatan Umum

```
//div[@id='main'] # div dengan id="main"
//table/td # semua <td> dalam <table> mana pun
../sibling # saudara via induk
//span # keturunan span dari konteks
```

Pola Umum

Web Scraping / Testing

```
//input[@name='username'] # input form berdasarkan nama
//button[text()='Submit'] # tombol berdasarkan teks
//div[contains(@class, 'error')] # elemen dengan class parsial
//a[contains(@href, 'login')] # tautan dengan href parsial
```

Pemilihan Kondisional

```
//div[@class='item'] [./span[@class='price']]
//tr[td[1]='Active'] # baris di mana sel ke-1 = Active
//*[contains(text(), 'Warning')] # elemen apa pun dengan teks
```

XPath di Python (lxml)

```
from lxml import html
tree = html.fromstring(page_content)
links = tree.xpath('//a/@href')
titles = tree.xpath('//h2/text()')
```

XPath di Selenium

```
driver.find_element(By.XPATH, "//input[@id='search']")
driver.find_elements(By.XPATH, "//li[@class='result']")
```