

# Penentuan Motif Pada Gambar Berpola

M. Albadr Lutan Nasution<sup>1</sup>, 13508011

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

<sup>1</sup>if18011@students.if.itb.ac.id, m.ln.albadr@gmail.com

**Abstract**—Sebuah gambar pada umumnya memiliki pola atau motif. Gambar yang hanya tersusun atas susunan tertentu disebut gambar berpola. Susunan tersebut biasanya merupakan perulangan yang apik yang menyusun gambar. Pelacakan balik pola perulangan atau motif dari gambar tersebut dapat dilakukan dengan merubah representasi gambar jadi teks dan melakukan pencarian perulangan pola teks padanya. Perulangan pola yang berhasil ditemukan ini adalah motif dari gambar. Kemungkinan kita bisa menemukan motif dari sebuah gambar dengan cukup akurat kita bisa menggunakannya untuk karya gambar yang lain yang lebih indah

**Index Terms**— Brute Force, Motif, Pemisahan Komponen Gambar, Pencarian Pola pada Gambar.

## I. PENDAHULUAN

Algoritma pencarian string tampaknya adalah algoritma yang biasa saja. Algoritma ini memiliki banyak sekali variasi dan penelitiannya hingga saat ini masih dilakukan. Inti dari algoritma ini adalah menemukan ada atau tidaknya sebuah pola string dalam sebuah teks. Jika ada, memberitahukan letak pola string tersebut ditemukan di dalam teks.

Namun, meskipun prinsipnya sederhana, algoritma ini memiliki aplikasi yang tidak bisa dipungkiri. Prinsip pencarian pola string dalam suatu teks string besar ini tidak hanya dapat digunakan dalam mencari kata dalam bahasa sehari-hari, seperti dalam *search engine* atau dalam teks editor, tetapi juga dapat diaplikasikan untuk pencarian lain yang lebih kompleks. Contoh-contoh hal ini banyak sekali. Misalkan pencarian sidik jari, pencocokan suara, pemecahan untaian DNA, meneliti virus dan mutasi dan sebagainya.

Hal-hal tadi dapat dilakukan tidak lain karena manusia dapat menerjemahkan masalah tadi menjadi model yang mirip dengan pencarian string. Masalah-masalah tadi bisa dianggap sebagai pencarian pola string pada teks biasa dengan membuat representasi teks dari domain yang ingin kita pecahkan. Misalkan dalam pencocokan suara, domain utama tempat pencarian dilakukan adalah database gelombang suara. Dengan teknologi digital, segala macam bentuk dalam kehidupan kita dapat diterjemahkan (dan disederhanakan dengan mengurangi detail yang tidak perlu) menjadi digit-digit atau bit-bit. Representasi bit ini

sudah mirip dengan teks yang akan menjadi domain pencarian.

Sama halnya dengan sebuah gambar, representasi digit atau bit dapat diambil terhadapnya. Sebenarnya teks bit itulah yang benar-benar disimpan dalam era komputer dan media digital yang ada dalam sekarang ini.

Makalah ini menceritakan suatu contoh penggunaan sederhana algoritma pencarian string terhadap gambar. Contoh yang digambarkan dalam makalah ini berhubungan dengan pola pada gambar. Seperti yang kita ketahui, pola bukan hanya istilah yang terbatas pada teks atau string saja, tetapi terdapat pula pada gambar. Pencarian pola di dalam gambar tentu saja dapat dilakukan melalui representasi gambar berupa bit tadi. Akan tetapi, contoh yang dibicarakan dalam makalah ini tidak (hanya) berkisar tentang pencarian keberadaan sebuah pola gambar tertentu dalam sebuah gambar yang lebih besar, tetapi lebih kepada pencarian sebuah pola terkecil atau atomik yang merupakan penyusun sebuah gambar berpola.

## II. GAMBAR DAN MOTIF

Kata pola sendiri secara natural didapatkan oleh manusia secara visual. Pola sendiri berarti suatu objek atau kejadian yang terjadi secara berulang-ulang. Pola mingguan, pola terbitnya matahari, pola susunan meja di dalam kelas, atau pola makan teman sekosan merupakan contoh yang mudah dipahami. Hal paling prinsip dari pola ini adalah repetisi atau berperiodik.

Dalam konteks gambar yang dengan kata lain objek visual hal ini juga sangat mudah dicerna karena setiap hari tentu kita melihatnya. Misalkan sebuah ubin di lantai. Penyusun utamanya hanyalah sebuah ubin yang diduplikasi beberapa buah hingga cukup untuk menutupi lantai. Atau pola bunga pada kemeja yang berulang beberapa kali mengelilingi kerah.

Beberapa orang menyebut bunga pada kemeja tadi dengan kata **motif**, yang mungkin bisa digeneralisasi menjadi istilah singkat dari sebuah pola gambar penyusun atomik tunggal dari gambar berpola. Istilah motif akan dipakai dalam makalah ini meskipun jarang kita mendengar orang berkata “lantai ini bermotif ubin abu-abu”.

Saat mendengar kata motif, mungkin kita bisa membayangkan sebuah gambar polygon (misalkan zigzag,

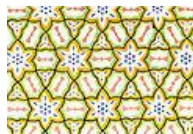
atau segienam) yang bisa digambar atau digambar berulang-ulang secara apik dalam sebuah kanvas. Akan tetapi, motif yang dibahas dalam makalah ini diasumsikan selalu berbentuk kotak persegi panjang. Hal ini berdasarkan asumsi bahwa setiap gambar yang pada asalnya memiliki motif yang bukan berbentuk persegi panjang dapat diambil amlop (batas kotak dari dari objek) untuk motif tersebut sehingga motif bisa dianggap kotak ataupun gambar utama yang disusun dari motif asal tersebut dapat dicari motif kotaknya yang mungkin sedikit berbeda dari motif asal. Hal ini juga mempertimbangkan fakta bahwa pada gambar digital, gambar terkuantisasi dengan pixel yang berbentuk kotak.



Gambar 1 Contoh Gambar Berpola



Gambar 2 Motif dari Gambar 1



Gambar 3. Gambar bermotif yang motif aslinya bukan kotak

Beberapa gambar bukan disusun dari motif-motif. Gambar-gambar ini bisa dibilang tanpa perulangan didalamnya. Walaupun begitu, gambar utama ini sendiri sebenarnya memiliki motif yakni dirinya sendiri. Motif pada Gambar 2 adalah dirinya sendiri.

Dengan mata manusia, ketika kita melihat sebuah gambar berpola seperti lantai berubin atau kemeja berkerah bunga tadi, kita dapat dengan mudah memutuskan pola terkecil yang menyusun gambar besar tadi. Sebuah ubin, atau gambar setangkai bunga. Akan tetapi, komputer tidak bisa langsung begitu saja menentukan bagian mana dan seluas apa yang menjadi motif atau penentu gambar besar tersebut.

Meskipun manusia cukup mampu untuk membedakan atau memilih gambar motif yang menjadi penyusun dari suatu gambar berpola, manusia tidak cukup teliti untuk memotong sebuah gambar tersebut menjadi sebuah motif yang jika diulang secara periodik akan menjadi gambar utuh asal.

Untuk beberapa pekerjaan,  *mungkin saja* penentuan pola dari sebuah gambar berpola ini cukup penting untuk dilakukan dengan lebih akurat. Mungkin saja hal ini dapat digunakan untuk kompresi gambar yang lebih baik, karena

kita tahu bahwa sebuah gambar besar bendera kenya adalah gambar yang hanya terdiri dari sebuah motif berukuran 1x1 pixel berwarna hijau. Atau mungkin ke depannya dengan perkembangan teknologi yang ada, beberapa teknologi akan memerlukannya.

### III. ALGORITMA PENCARIAN STRING DAN BRUTE FORCE

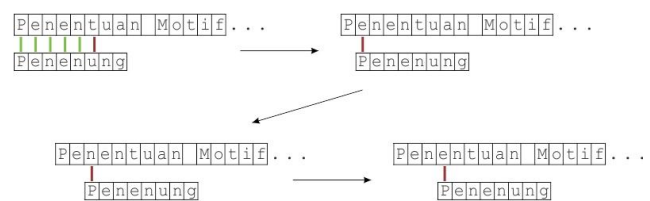
Ada banyak algoritma pencarian string yang telah dikembangkan. Yang paling sederhana adalah dengan cara *brute force*.

Teknik *brute force* adalah teknik yang mengandalkan solusi primitif langsung yang diketahui manusia. Dengan demikian teknik ini bersifat lempang yang langsung didasarkan pada pernyataan masalah atau definisi konsep yang dilibatkan tanpa melakukan trik-trik cerdas tambahan. Teknik atau algoritma ini menyelesaikan masalah dengan cara sederhana, karenanya terkadang algoritma ini disebut algoritma naif (*naïve algorithm*).

Contoh penggunaan dari *brute force* misalkan dalam pencarian faktorial. Definisi matematis sebuah faktorial  $n$  adalah  $n! = n \cdot (n-1) \cdot \dots \cdot 1$ . Dengan definisi konsep ini algoritma *brute force* melakukan penghitungan secara faktorial secara langsung dengan mengkalikan nilai  $n$  dengan  $(n-1)!$  dan seterusnya secara rekursif.

Dengan menggunakan *brute force*, pencarian dilakukan secara langsung juga. Misalkan untuk mencari sebuah pola “Penenung” dalam dokumen ini, hal yang pertama dilakukan adalah mencocokkan huruf pertama pada pola dengan huruf pertama pada dokumen. Bila cocok, pengecekan dilakukan pada huruf ke dua pola dengan huruf kedua teks, dan seterusnya sampai huruf terakhir pola juga ditemukan sekuens dalam teks.

Bila ada yang tidak cocok, huruf pertama pada pola disejajarkan dengan huruf kedua pada dokumen dan pencocokan huruf per huruf pada pola diulang. Bila seluruh huruf pada pola cocok, pola resmi telah ditemukan dalam teks.



Gambar 4 Ilustrasi pencocokan string dengan *brute force*

Teknik atau algoritma *brute force* pada umumnya tidak disukai karena memiliki ketidakmangkusannya. Akan tetapi algoritma ini dapat diterapkan pada sebagian besar masalah. Karena sederhana, algoritma ini juga mudah diimplementasikan dan dengan kekuatannya dapat menyelesaikan masalah dengan kepastian, tidak seperti algoritma canggih dan cerdas yang sulit diimplementasikan dan terkadang memiliki kasus-kasus lemah.

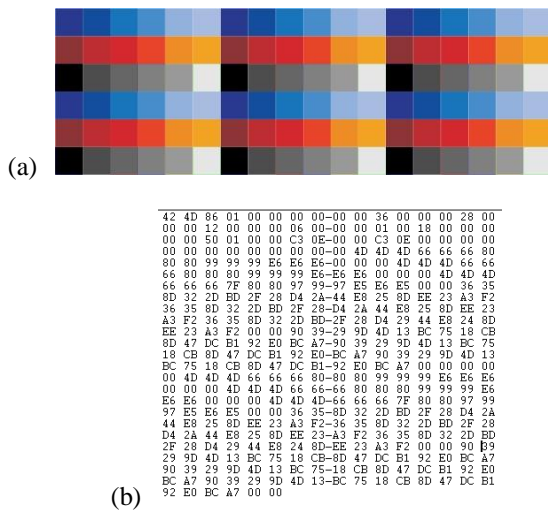
Terkadang, karena kesederhanaannya, teknik *brute force* lebih diperhitungkan untuk masalah yang cukup kecil. Kompleksitas *brute force* bergantung pada konsep masalah yang digunakan. Pada satu masalah pun bisa terdapat beberapa penyelesaian *brute force* yang berbeda bergantung konsep tadi. Terkadang, memberi sedikit bumbu atau trik dalam penggunaan *brute force* cukup membantu mengubah pandangan terhadap masalah.

#### IV. PRINSIP KERJA PENCARIAN MOTIF

##### A. Representasi Teks dari Gambar

Penyesuaian perlu dilakukan agar algoritma pencarian string dapat diterapkan. Penyesuaian ini sudah pasti seputar mengubah representasi pixel pada gambar menjadi teks besar yang dapat dijadikan domain pencarian.

Karena sebuah gambar dalam komputer pasti berbentuk digital, gambar tersebut sudah barang tentu memiliki representasi bit (atau biasanya ditampilkan dalam angka hexadecimal). Representasi bit atau HEX ini dapat kita gunakan sebagai sebuah domain teks tempat kita melakukan pencarian.



Gambar 5 (a) Sebuah Gambar berpola sederhana  
(b) Representasi bit Gambar 5(a) dalam hex

Akan tetapi, ada beberapa hal yang mesti dicatat. Dalam menentukan motif dalam sebuah gambar, kita harus mencari perulangan bagian gambar yang terdapat dalam gambar utama tadi. Dengan kata lain, kita harus dapat melakukan pencarian keberadaan perulangan bagian gambar per baris pixel dari gambar utama.

Dengan memakai representasi bit tadi, hal ini cukup sulit untuk dilakukan. Kesulitan terjadi karena untuk mengetahui representasi bit satu baris pixel teratas dari suatu gambar kita harus menghitung berapa bit yang dipakai oleh satu baris pixel tersebut. Jika hal ini dilakukan pun, kita harus mengambil asumsi bahwa bit-bit tersebut hanyalah merepresentasikan pixel-pixel dari gambar, bukan informasi lain yang mungkin tidak relevan dalam pencarian motif ini.

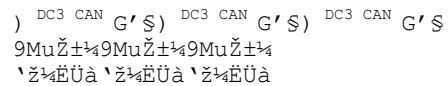
Cara lain untuk mencari representasi adalah dengan mengubah pixel-pixel gambar tadi menjadi teks sedemikian sehingga satu baris pixel dari ujung kiri gambar sampai ujung kanan gambar terdapat pada satu baris teks. Pixel sendiri pada dasarnya tersusun atas tiga komponen warna merah, hijau, biru alias RGB (Red, Green, dan Blue) yang masing-masing memiliki kisaran nilai 0 sampai 255, sebuah kisaran nilai pada representasi huruf dalam *character encoding* dengan format ISO-8859-1 (Latin-1) atau format Windows-1252. Dengan demikian untuk masing-masing pixel kita dapat menentukan tiga *encoding* huruf sesuai RGB-nya.

Tiga huruf untuk satu pixel tampak bukanlah cara yang hemat untuk membuat sebaris teks dari sebaris pixel. Hal ini mungkin dapat diatasi dengan mencari nilai rata-rata dari ketiga nilai RGB dari sebuah pixel sehingga kita memiliki sebuah representasi huruf Latin untuk setiap pixel. Akan tetapi hal ini akan menambah masalah. Karena dengan demikian dapat dimungkinkan dua pixel dengan warna yang berbeda memiliki representasi huruf yang sama.



Gambar 6 Dua pixel berbeda warna dengan rerata RGB sama

Penjumlahan ketiga nilai RGB dan penggunaan format huruf yang lebih kaya dari Latin-1 dan Windows-1252 seperti Unicode juga tidak dapat dilakukan karena kemungkinan menghasilkan huruf yang sama dari dua pixel berwarna jauh berbeda masih ada. Dengan demikian, representasi teks dari sebuah baris pixel melalui konversi nilai ke *encoding character* adalah tiga baris teks Windows-1252 dari nilai masing-masing warna dasar RGB gambar.



Gambar 7 Tiga baris huruf Windows-1252 dari nilai RGB baris pixel pertama Gambar 5 (a)

Sebenarnya beberapa angka pada *encoding character* manapun tidak seluruhnya dipetakan ke huruf yang dapat dicetak. Beberapa adalah karakter kontrol. Misalkan pada contoh di atas <sup>DC3</sup> dan <sup>CAN</sup> adalah singkatan dari fungsi sebuah karakter kontrol. Akan tetapi, demi kesederhanaan bahasan hal ini diabaikan dalam makalah ini.

##### B. Teknik Pencarian

Dasar dari teknik mencari sebuah motif dari sebuah gambar utama tentu saja adalah algoritma pencarian string. Akan tetapi, penggunaan algoritma ini dalam penentuan motif tampaknya sedikit berbeda dibandingkan penggunaannya dalam pencarian seperti mencari kata "Penenung" dalam dokumen ini.

Hal mendasar dan paling awal yang berbeda adalah kita

tidak tahu pola bagaimana yang kita cari. Sebaliknya, dengan algoritma ini kita harus dapat menentukan sebuah pola penyusun (alias motif) dari gambar besar tempat domain pencarian.

Ide paling sederhana untuk melakukan hal ini adalah dengan mencocokkan representasi string dari gambar tadi dengan tetangganya hingga didapati perulangan dengan jumlah tertentu, dengan kata lain *brute force*. Dengan penggunaan algoritma pencarian string paling sederhana juga yaitu *brute force*, kita dapat bilang teknik ini sebagai *brute force* ganda.

Misalkan terdapat sebuah baris teks yang merupakan hasil representasi huruf dari nilai warna biru baris teratas gambar utama seperti pada Gambar 7.

Secara kasat mata kita dapat menebak motif dari teks diatas. Secara prosedural, motif tebakan kita dapat ditentukan dengan beberapa tahapan.

Pertama, ambil karakter pertama dalam baris teks ke dalam sebuah **kantong**. Kemudian, cocokkan karakter kantong dari indeks ke dua dalam teks dengan menggunakan algoritma pencarian, misalkan dengan cara *brute force*. Jika tidak cocok, ulangi dari langkah awal dengan mengambil karakter ke dua dalam baris teks ke dalam kantong. Dengan demikian, di dalam kantong sekarang terdapat dua karakter.

Sama seperti sebelumnya, karakter dalam kantong ini menjadi “pola” pada pencarian string seperti biasa. Akan tetapi, pencarian bukan dilakukan dari depan teks, melainkan dari indeks yang belum dimasukkan karakternya ke dalam kantong sebagai “pola”.

Tahapan ini terus dilakukan berulang sampai ditemukan pola yang sama dalam satu baris tersebut. Setelah pencarian berhenti, kita dapat menentukan motif yang kita cari, yakni pola terakhir dalam kantong sebagai kandidat.



**Gambar 8** Ilustrasi Penentuan Motif pada Sebaris Teks

Hal yang perlu dicatat adalah, keberhasilan menemukan motif pada representasi teks pada nilai RED tidak menjamin motif tersebut adalah motif dari gambar. Kita

juga harus memastikan bahwa motif yang ditemukan pada representasi teks pada nilai BLUE, dan GREEN jatuh pada panjang kantong yang sama dengan motif RED.

Mengapa? Hal ini karena representasi teks tadi adalah berasal dari pixel. Setiap pixel punya tiga nilai RGB, sehingga dalam panjang kantong yang sama dari ketiga nilai RGB adalah pixel sepanjang panjang kantong tadi.

Pada Gambar 8 telah ditemukan sebuah motif dari baris teratas dari pixel gambar. Karena pada gambar dimungkinkan sebuah motif bukan hanya terdiri dari sebaris pixel saja, penentuan pola tidaklah sampai disini. Dengan demikian, gambar diatas hanyalah sebuah kandidat dari motif gambar yang kita cari.

Pencarian bertahap seperti diatas kemudian juga dilakukan pada baris-baris selanjutnya hingga mencapai baris paling bawah. Sama seperti dalam satu baris, setiap motif ditemukan dari masing-masing teks representasi nilai RGB harusnya memiliki panjang yang sama (berbentuk kotak), panjang pixel dari motif tiap baris juga haruslah memiliki panjang yang sama. Apabila ada ketidaksamaan panjang, pencarian harus diulangi dari baris teratas.

Teknik ini dapat dioptimasi dengan menggunakan fakta menyebalkan tadi yaitu motif adalah berbentuk kotak. Setelah pencarian baris pertama dilakukan, sebuah nilai **Len** yang merupakan panjang dari kandidat motif di baris pertama didapatkan. Baris kedua dan seterusnya dapat menggunakan nilai Len ini sebagai inisiasi dari panjang kantong dalam penentuan motifnya. Karena kita tahu bahwa tidak mungkin motif sepanjang kurang dari Len di baris selanjutnya dapat dijadikan kandidat.

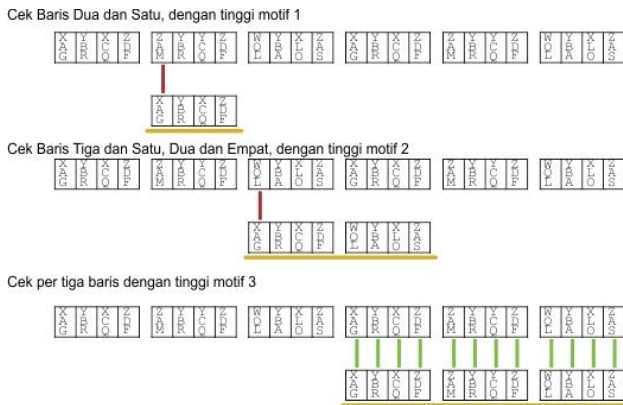
Setelah motif dari baris teratas pixel sampai motif baris terbawah pixel ditemukan, pekerjaan pencarian belumlah juga selesai. Hal ini karena motif atau perulangan gambar tidak hanya terdapat dalam baris-baris pixel, tetapi juga dapat ditemukan dalam kolom. Motif gambar dapat ditemukan dengan mencari lagi motif koloman pada set motif barisan yang ditemukan tadi sebagai gambar utamanya. Tentu saja hal ini sedikit banyak terlihat seperti pemborosan.

Hal ini dapat dioptimasi dengan menambah pekerjaan saat mencari pencarian motif pada baris ke 2 sampai baris ke N. Pekerjaan yang ditambahkan adalah setelah mendapat motif sepanjang Len pada baris ke k, motif ini dicocokkan (dengan algoritma pencarian, cukup secara *brute force*) dengan motif yang ditemukan dalam baris pertama. Jika cocok, pencarian terus dilakukan ke baris k+1 dan motif dicocokkan ke motif yang ditemukan pada baris ke 2. Motif dari gambar utama didapatkan bila motif pada baris ke 1 sampai baris ke k-1 cocok dengan motif pada baris ke k sampai baris ke 2(k-1). Inilah motif dari gambar yang kita cari.

Meskipun begitu optimisasi di atas tidak selalu dapat dilakukan, selain agak membingungkan, untuk gambar dengan jumlah baris ganjil dan beberapa kasus, teknik di atas gagal. Contoh kasus adalah saat ditemukan gambar dengan kandidat motif baris pixel pertama dan ketiganya sama. Jika tinggi gambar hanya 3 pixel, kemungkinan



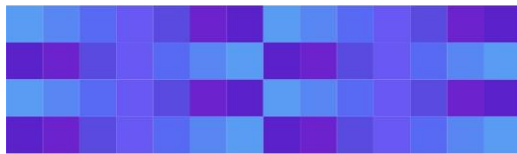
tinggi motif juga 3. Jika tinggi gambar 4 pixel ada peluang bahwa tinggi motif 2 pixel atau 4 pixel. Ketidaktentuan ini sulit diatasi jika kita belum mengetahui tinggi gambar dalam pixel dan bentuk keseluruhan kandidat motif dari seluruh pixel yang ada.



Gambar 9 Ilustrasi Penentuan Motif Antar Baris Pixel

## V. STUDI KASUS

Misalkan gambar yang ingin kita cari motifnya adalah Gambar 10 **Gambar Contoh untuk Studi Kasus** Gambar 10 di bawah yang berukuran 14x4 pixel.



Gambar 10 Gambar Contoh untuk Studi Kasus

Dengan memisahkan nilai RGBnya dan membuat teks berdasarkan encoding character Windows-1252, gambar di atas dapat direpresentasikan sebagai teks-teks seperti pada Gambar 11.

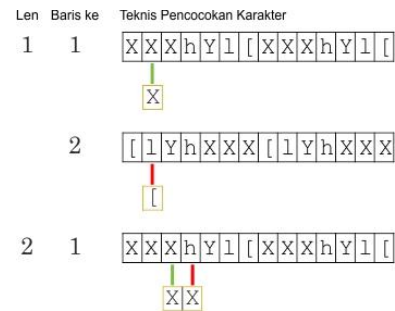
Teks Merah	Teks Hijau	Teks Biru
XXXhYl [XXXhYl [	œþjXJ"œþjXJ"	ïïïïïïïïïïïï
[lYhXXX [lYhXXX	"JXj+œ"JXj+œ	ïïïïïïïïïïïï
XXXhYl [XXXhYl [	œþjXJ"œþjXJ"	ïïïïïïïïïïïï
[lYhXXX [lYhXXX	"JXj+œ"JXj+œ	ïïïïïïïïïïïï

Gambar 11 Representasi Teks Windows-1252 dari Gambar 10

Pencarian dimulai dari teks representasi warna merah. Sebenarnya bisa saja dimulai dari warna lain, atau dari bawah dengan urutan menurut warna, urutan pixel per baris, atau urutan kolom. Pada Gambar 9 pencarian dilakukan dengan urutan pencarian dalam teks per baris pixel, kemudian antar baris pixel. Pada studi kasus ini, dicoba pencarian dengan urutan pencarian dalam representasi teks merah hingga seluruh baris dan mendapat motif koloman, kemudian pindah warna dasar.

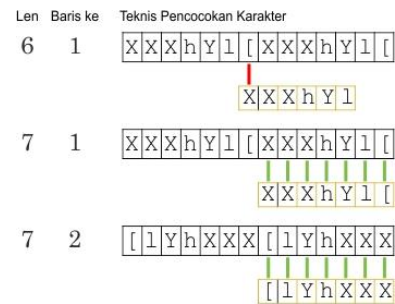
Pada pencocokan pertama di baris pertama teks merah, karakter X (yakni hasil memasukkan kantong dari karakter pertama) dicocokkan ke karakter ke 2 pada teks. Hasilnya

didapatkan kandidat motif yakni X dengan **Len** (panjang kandidat) sama dengan satu. Dengan **Len = 1** digenggaman, dilakukan pencocokan pada baris ke dua. Pada baris kedua tidak terjadi kecocokan, pencocokan pun kembali ke baris satu.



Gambar 12 Pencocokan di ranah teks merah sampai Len 2

Dengan **Len** sepanjang 2, kecocokan di baris kesatu tak ditemukan. Pencocokan pun kemudian dilakukan terus pada baris pertama hingga diperoleh **Len** sama dengan tujuh dan diperoleh kandidat motif baris bertama teks merah XXXhYl [.



Gambar 13 Pencocokan di ranah teks merah dengan Len 7

Dengan **Len = 7** ini ternyata juga cocok pada baris kedua dari teks merah. Sehingga diperoleh kandidat motif dari teks merah seperti pada dua kantong dari baris 1 dan 2 pada Gambar 13 di atas.

Dengan **Len=7**, dengan mudah diperoleh kandidat motif untuk baris ketiga dan keempat teks merah. Sehingga sekarang kita memiliki set lengkap dari kandidat baris dalam satu teks merah. Hal ini menunjukkan kesuksesan nilai **Len=7**, setidaknya dalam teks merah.

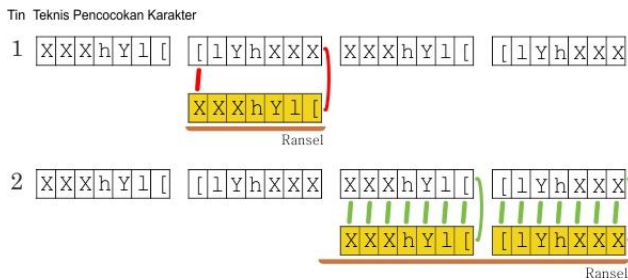
Secara keseluruhan (untuk gambar utama) motif juga memiliki tinggi. Sebut saja variabel **Tin** menyatakan banyak baris dari sebuah motif. Saat kandidat motif dari baris pertama ditemukan, **Tin** diset menjadi 1. Jika baris ke dua tidak sama dengan baris pertama ini, **Tin** pasti lebih dari satu.

Pencocokan kandidat motif antar baris kini dilakukan berdasarkan set lengkap dari kandidat barisan dari teks merah. Prinsip pencarian sama seperti pencarian kandidat motif per baris. Untuk kumpulan kandidat motif antar baris, mari kita gunakan istilah **ransel** dibanding kantong.

Kandidat motif baris bertama dimasukkan ke ransel, **Tin** diset dengan nilai satu. Isi ransel dicocokkan dengan kandidat motif mulai dari baris ke **Tin+1**, alias baris kedua.

Ternyata kandidat motif baris satu dan dua tidak sama, sehingga Tin sekarang bernilai dua dan kandidat motif baris dua juga dimasukkan ke ransel.

Dengan nilai  $Tin=2$ , pencocokan selanjutnya dimulai dari baris ketiga. Isi ransel pertama (yakni kandidat motif baris kesatu) dicocokkan dengan kandidat motif baris ketiga. Karena sama, pencocokan dilanjutkan ke baris selanjutnya. Isi ransel kedua (alias kandidat motif baris kedua) dicocokkan ke kandidat motif pada baris keempat, yang ternyata sama juga. Kebetulan baris waktu itu hanya empat. Hal ini menandakan bahwa **motif sementara** (kandidat motif yang telah lolos berbagai uji lokal, tetapi masih kandidat) teks merah telah ditemukan dengan nilai Len tujuh dan Tin dua.



Gambar 14 Pencocokan kandidat motif dengan ransel

Tahapan selanjutnya adalah mencari motif sementara pada teks hijau dan biru. Jika Len dan Tin nya sama dengan teks merah, motif dari gambar utama resmi ditemukan.

Untuk memudahkan pencarian pada ranah selanjutnya, nilai Len dan Tin pada ranah teks merah dapat digunakan. Pada ranah selanjutnya ini, langsung dicek setiap baris untuk  $Len=7$  apakah kandidat motif ditemukan. Dan langsung dicek apakah nilai  $Tin=2$  juga berlaku untuk ranah tersebut.

Akhirnya, motif dari Gambar 10 ditemukan seperti gambar di bawah ini.



Gambar 15 (a) Teks Motif yang ditemukan (b) Gambar motif ditemukan dari Gambar 10

## VI. BATASAN DAN PELUANG

Meskipun dalam studi kasus di atas motif berhasil ditemukan, teknik untuk menentukan motif seperti yang digunakan pada studi kasus belumlah sempurna. Ada beberapa hal yang sulit untuk ditetapkan.

Hal yang paling sulit adalah untuk menetapkan berapa kali sebuah kantong sepanjang Len dipastikan ada dalam sekuens sebaris teks – kecuali sejumlah Len karakter awal

yang tidak dilakukan pengecekan sama sekali – sehingga isi kantong tersebut bisa dinyatakan sebuah kandidat motif. Dalam semua contoh sebelumnya, jumlah pengecekan kandidat –anggap saja namanya Konstanta Kandidat – hanya sekali. Artinya, saat isi kantong ditemukan sekali – sekali lagi tidak termasuk sejumlah Len karakter awal memang tidak dilakukan pengecekan sama sekali terhadapnya –, isi kantong dianggap sebagai kandidat motif. Pada praktiknya masih terlalu awal untuk memutuskan hal ini.



Gambar 16 Gambar yang motifnya ada di dua baris teratas



Gambar 17 Motif dari Gambar 16



Gambar 18 Potongan dari Gambar 16 yang mungkin salah dianggap motif

Misalkan pada Gambar 16 di atas. Untuk Konstanta Kandidat sekali, jelas bahwa terjadi kesalahan penentuan kandidat motif. Bahkan untuk Konstanta Kandidat sama dengan lima (dicek apakah dua pixel pertama ada di baris pertama pada gambar enam kali termasuk pada awal baris), hasil akan menunjukkan Gambar 18 sebagai motif dari Gambar 16.

Belum lagi jika hampir seluruh gambar berpixel besar adalah berwarna merah dengan garis tipis warna kuning hanya di pinggir kanan dan bawahnya. Berapa banyak pengecekan keberadaan isi kantong (saat itu adalah satu pixel merah) pun jika tidak sampai pinggir kanan dan bawah akan membawa kesimpulan pada motif gambar adapah satu pixel merah. Mungkin benar secara rata-rata statistika, akan tetapi sebenarnya motif gambar itu adalah gambar itu sendiri.

Kesimpulannya, meskipun tampak pemborosan sekali melakukan pengecekan keberadaan isi kantong sampai ujung baris, hal ini sangatlah penting agar tidak salah menentukan motif.

Representasi teks dari gambar juga mungkin menimbulkan masalah. Seperti yang telah disebutkan sebelumnya, beberapa nomor pada *encoding character* termasuk Windows-1252 adalah karakter kontrol yang artinya tidak dapat ditampilkan di layar sehingga tidak dapat dilakukan pencarian terhadapnya. Masalah ini mungkin dapat diselesaikan dengan membuat pemetaan

karakter sendiri dari nilai 0 sampai 255.

Pemakaian algoritma *brute force* pada contoh mungkin cukup aman. Tetapi untuk gambar sangat besar hal ini mungkin tidak diinginkan. Apalagi ada kemungkinan kasus di atas. Pengembangan berikutnya **mungkin** dapat dilakukan dengan menerapkan prinsip *divide and conquer* untuk memecah gambar jadi beberapa bagian dan membuat teknik penentuan motif menjadi lebih mangkus.

Suatu gambar juga tidak mesti memiliki hanya satu motif. Banyak sekali gambar yang terdiri dari beberapa pola perulangan. Gambar 16 misalnya, bisa dianggap sebagai gambar dengan motif Gambar 18 di sebelah kiri, dan gambar dengan motif hijau-merah, hijau-merah muda di tepi kanan. Gambar 2 dan Gambar 3 juga sebenarnya merupakan gambar yang bermotif kompleks. Gambar dari motif juga sebenarnya bisa dibentuk dengan mencerminkan motif, merotasi, dan lain-lain. Kedepannya, mungkin dapat dilakukan pemecahan suatu gambar kompleks yang sebenarnya tersusun dari beberapa motif menjadi motif-motif penyusunnya itu, entah dengan prinsip pencarian string atau dengan cara lain yang lebih pintar.



**Gambar 19** Gambar bermotif yang dibentuk dari mencerminkan motif

Akhirnya, penerapan penentuan motif dari sebuah gambar berpola jatuh pada kualitas gambar. Hal ini ditentukan oleh sudut potret gambar, keseragaman pencahayaan, transparansi, bayangan, dan lain-lain. Seperti pada Gambar 19 di atas, gambar sebenarnya terpotret sedikit miring ke kanan. Mencari motif dari sebuah gambar berpola yang kita buat sendiri dari motif tertentu secara digital dengan hati-hati tentu saja adalah hal yang – tidak bisa diungkapkan dengan kata-kata. Semoga nantinya ada penyelesaian mudah terhadap masalah kualitas gambar ini.

## VII. SIMPULAN

Penentuan motif dari sebuah gambar digital melibatkan penerjemahan representasi gambar menjadi representasi teks per pixel sehingga dapat dilakukan pencarian string padanya. Pencarian string ini dilakukan untuk mencari kandidat motif dari setiap baris pixel pada gambar. Jika sebuah pola acuan (yang diambil N buah dari awal baris) ditemukan lagi pada teks tersebut, bisa jadi pola acuan ini adalah kandidat motif dari baris tersebut. Ada kemungkinan pula kandidat motif ini hanyalah bagian dari kandidat motif yang lebih besar dalam baris. Setelah pencarian dilakukan dari ujung ke ujung baris, barulah motif baris tersebut resmi ditemukan.

Dari motif atau kandidat motif yang telah ditemukan dari semua baris pixel pada gambar, ditentukan motif

utama dari sebuah gambar. Bisa jadi bahwa motif pada setengah tinggi atas gambar berulang di setengah tinggi bawah gambar, sehingga gambar adalah perulangan dua kali secara vertikal dari motifnya. Penentuan ini dilakukan sama seperti penentuan motif per baris, yakni dengan mengecek apakah sebuah pola acuan (yang diambil dari N buah motif atau kandidat motif baris dari atas gambar) berulang beberapa kali secara vertikal sampai di baris paling bawah gambar.

Algoritma *brute force* yang diterapkan dalam penentuan motif sebuah gambar ini cukup membantu dalam memahami masalah yang ada. Dengan algoritma ini, masalah yang ada secara eksplisit dapat dipecahkan dengan mudah, hanya untuk kasus-kasus besar butuh waktu yang besar pula dalam memecahkannya.

## REFERENSI

- [1] Jones, Owen. *The Grammar of Ornament*. Dover Publications, Revised edition (1987).
- [2] Munir, Rinaldi. 2007. *Strategi Algoritmik*. Bandung : Teknik Informatika ITB.
- [3] [http://en.wikipedia.org/wiki/Pattern\\_recognition](http://en.wikipedia.org/wiki/Pattern_recognition)
- [4] <http://www.microsoft.com/globaldev/reference/sbcs/1252.msp>

## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 07 Desember 2010

M. Albadr Lutan N.  
13508011