BAB II

TINJAUAN PUSTAKA

Bab ini akan menjelaskan mengenai teori-teori yang menjadi acuan tim pengembang dalam mengembangkan proyek mahasiswa ini. Bab ini terbagi menjadi enam subbab, yaitu subbab yang menjelaskan tentang definisi citra, jenis-jenis citra digital, ruang warna RGB, transformasi citra, format dan resolusi citra digital, dan tinjauan beberapa algoritma.

2.1 Definisi Citra

Secara umum citra adalah fungsi intensitas cahaya f(x,y), dimana harga x dan y merupakan koordinat spasial dan harga fungsi tersebut pada setiap titik (x,y) merupakan tingkat kecemerlangan citra pada titik tersebut [GONZ02].

Citra digital memiliki definisi yang serupa dengan citra secara umum, namun pada citra digital fungsi f(x,y) mengalami diskritisasi koordinat spasial (sampling) dan diskritisasi tingkat kecermelangan atau tingkat keabuan (kuantifikasi) [GONZ02]. Dapat dikatakan bahwa citra digital merupakan suatu matriks dimana indeks baris dan kolomnya menyatakan suatu titik pada citra tersebut dan elemen matriksnya (yang disebut sebagai elemen gambar/ piksel/ *picture elements*/ *pels*) yang menyatakan tingkat keabuan pada titik tersebut.

Pembentukan citra pada proyek mahasiswa ini menggunakan alat pemindai standar untuk menghasilkan citra digital dari LIK. Untuk selanjutnya, sebutan citra dan citra digital tidak dibedakan, keduanya mengacu kepada citra digital.

Untuk dapat mengenali bundaran yang dihitamkan, citra digital LIK terlebih dahulu perlu diproses. Pada subbab-subbab berikut akan dibahas beberapa istilah pemrosesan citra digital yang terkait.

2.2 Jenis-jenis Citra Digital

2.2.1 Citra Berwarna

Citra berwarna dapat dipandang sebagai sebuah matriks tiga dimensi M dengan ukuran m x n x 3. Setiap elemen pada matriks M menyatakan tingkat kecermelangan suatu citra pada koordinat spasial yang bersesuaian. Untuk setiap kedalaman memiliki matriks dengan ukuran m x n, masing-masing kedalaman menyatakan tingkat kecermelangan untuk komponen warna sesuai dengan kedalamannya [GONZ02]. Untuk lebih jelasnya dapat dilihat pada Gambar 2.1 sebagai berikut:

Gambar 2. 1 Representasi Citra Berwarna dalam Matriks 3 Dimensi

Pada Gambar 2.1 citra berwarna memiliki tiga komponen kedalaman warna. Pada matriks tersebut, detil citra berwarna akan bergantung kepada nilai untuk masing-masing komponen kedalaman warna. Kedalaman warna tergantung kepada ruang warna yang digunakan citra tersebut. Beberapa ruang warna antara lain RGB (Red, Green, Blue), CMYK, L*a*b, HIS, dan sebagainya. Ruang warna yang masih berkaitan dengan sistem ini adalah

ruang warna RGB. Penjelasan lebih detail mengenai ruang warna RGB, akan dibahas pada subbab 2.3.

2.2.2 Citra *Grayscale*

Apabila citra berwarna dapat dipandang sebagai sebuah matriks 3 dimensi, lain halnya untuk citra *grayscale*. Citra *grayscale* dapat dipandang sebagai sebuah matriks 2 dimensi dengan ukuran m x n, dan setiap elemen pada matriks tersebut merepresentasikan intensitas atau tingkat keabuan dalam skala tertentu yang menentukan kecermelangan suatu piksel pada koordinat yang bersesuaian.

Pada citra *grayscale* umumnya setiap piksel dalam matriks citra memiliki nilai intensitas keabuan yang disimpan dengan ukuran 8 bit, sehingga terdapat 256 kemungkinan nilai intensitas, yakni mulai dari 0 hingga 255. Nilai 0 menyatakan intensitas keabuan paling tinggi, yaitu warna hitam. Sebaliknya untuk nilai 255 menyatakan intensitas paling rendah, yaitu warna putih.

2.2.2 Citra *Black & White (Binary)*

Sama seperti citra *grayscale*, citra *black & white* dapat dipandang sebagai sebuah matriks 2 dimensi dengan ukuran m x n, dan setiap elemen pada matriks tersebut merepresentasikan intensitas atau tingkat keabuan suatu piksel dalam skala tertentu pada koordinat yang bersesuaian.

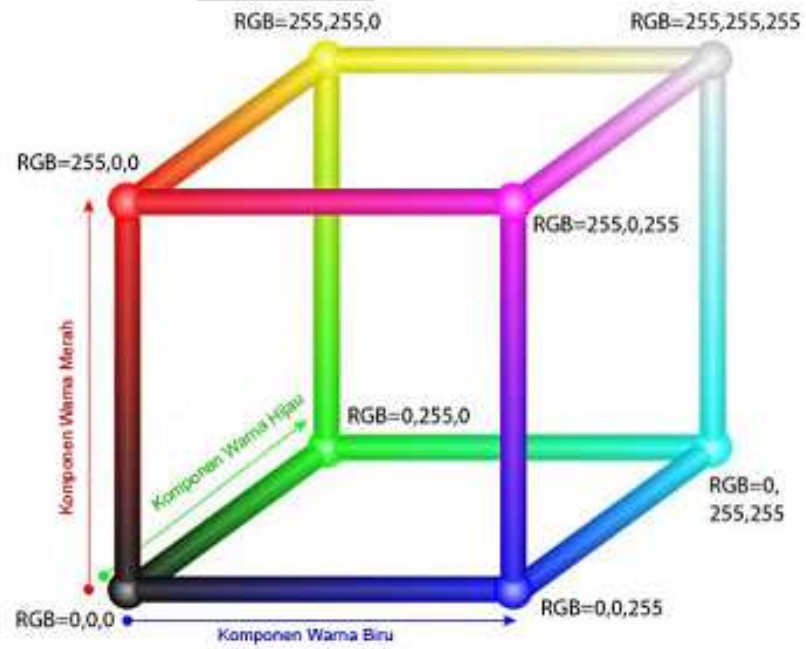
Berbeda dengan citra *grayscale* yang memiliki tingkat keabuan yang bervariasi dari 0 sampai 255, citra *black & white* hanya terdiri dari dua warna yaitu hitam (frekuensi 0) dan putih (frekuensi 255). Citra ini disimpan dengan ukuran 1 bit yang hanya memiliki dua kemungkinan nilai warna. Warna hitam akan bernilai 1 dan sebaliknya warna putih akan bernilai 0.

2.3 Ruang Warna RGB

Seperti yang telah dijelaskan tentang citra berwarna pada bagian 2.2.1, setiap citra berwarna memiliki ruang warna yang bersesuaian. Ruang warna adalah

spesifikasi yang diberikan terhadap suatu sistem koordinat dan ruang yang berada di dalam sistem koordinat tersebut, sehingga setiap titik di dalamnya dapat merepresentasikan sebuah warna [GONZ02].

Dalam ruang warna RGB, setiap warna direpresentasikan sebagai spektrum warna merah, hijau, dan biru. Setiap detil piksel warna ditentukan oleh nilai dari masing-masing komponen warna merah, hijau, biru tersebut. Berikut adalah model warna RGB yang digunakan untuk menspesifikasi sistem koordinat kartesius 3D untuk representasi warna.



Gambar 2. 2 Skema Kubus dari Ruang Warna RGB

Dapat dilihat pada gambar di atas bahwa warna hitam terdapat pada titik awal yaitu pada koordinat (0,0,0,) di mana setiap komponen warnanya bernilai minimal. Sedangkan warna putih direpresentasikan oleh koordinat (255,255,255) di mana setiap komponen warnanya bernilai maksimal.

2.4 Transformasi Citra

Transformasi citra, sesuai namanya, merupakan proses perubahan bentuk citra untuk mendapatkan suatu informasi tertentu [GONZ02]. Transformasi dapat dibagi menjadi dua:

- Transformasi piksel atau transformasi geometris:

  Transformasi piksel masih bermain di ruang atau domain yang sama (domain spasial), hanya posisi piksel yang kadang diubah. Contoh: rotasi, translasi, *scaling*, invers, *shear*, dan lain-lain. Pada proyek mahasiswa ini terdapat fungsi untuk rotasi, yang bertujuan meluruskan LIK yang sedikit miring ketika dipindai.

- Transformasi ruang/domain/space

  Transformasi ruang merupakan proses perubahan citra dari suatu ruang/domain ke ruang/domain lainnya, contoh: dari ruang spasial ke ruang frekuensi.

- Transformasi warna

  Transformasi warna merupakan proses perubahan warna suatu citra, baik transformasi dalam ruang warna yang sama (misalkan dalam ruang warna RGB), seperti operasi komplemen warna, pemotongan citra, dan lain-lain, maupun pengolahan citra dari citra berwarna menjadi citra *grayscale*, citra *black & white* atau sebaliknya.

Suatu LIK mengandung informasi setelah LIK tersebut diisi (dihitamkan). Pada LIK SPMB, UMB, dan SNMPTN isian yang dihtamkan berbentuk lingkaran. Untuk mendapatkan informasi yang terdapat pada LIK, maka perlu diketahui lingkaran mana yang dihitamkan oleh pengisi. Formulir LIK tersebut ada yang berwarna, adapula yang hitam putih (*grayscale*). Pada sistem ini, citra LIK yang berwarna akan ditransformasi menjadi citra *black & white* terlebih dahulu sebelum diproses. Begitu juga dengan citra *grayscale*. Transformasi dapat dilakukan oleh sistem "*Scanner Project*" maupun pada saat pemindaian oleh alat pemindai.

2.5 Format dan Resolusi Citra Digital

Terdapat bermacam format representasi citra dalam berkas, seperti BMP, GIF, PNG, JPG dan sebagainya. Masing-masing format memiliki kelebihan dan kekurangan.

Format BMP yang kaya warna merupakan format yang kurang efisien, karena semua informasi angka dalam baris disimpan seluruhnya. Bagian data citra sebenarnya bisa dikompresi agar ukuran berkas tidak terlalu besar. Salah satu cara kompresi adalah dengan terlebih dahulu mentransformasikan citra ke ruang yang berbeda, sebagai contoh adalah format berkas JPG. Format berkas JPG juga kaya warna, yang dapat mendukung 16 juta warna yang berbeda.

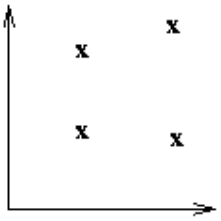
Lain halnya dengan format BMP monokrom 1 bit, yang hanya terdiri dari warna hitam dan putih yang dapat disimpan dengan nilai 0 atau 1 (*binary*). Citra dengan format ini memiliki ukuran yang relatif jauh lebih kecil dibandingkan format yang lain.

Dalam proyek mahasiswa ini, citra digital LIK yang telah dipindai disimpan dalam format berkas BMP monokrom 1 bit. Selain menyimpan citra LIK dalam format berkas jpg, ditentukan pula resolusi gambar dari citra tersebut. Resolusi gambar mengacu kepada jumlah piksel dalam sebuah citra, dan dihitung dengan menggunakan satuan dpi (dot per inci). Semakin besar resolusi gambar suatu citra, maka piksel citra tersebut akan semakin banyak dan gambarnya akan semakin halus. Namun, dapat memperlambat dalam melakukan proses pengolahan untuk ekstraksi informasi dari LIK tersebut serta mengakibatkan diperlukan ruang penyimpanan yang lebih besar. Pada sistem ”*Scanner Project*”, hasil terbaik diperoleh dengan menggunakan citra digital yang resolusinya 100 dpi.

2.6 Tinjauan Algoritma

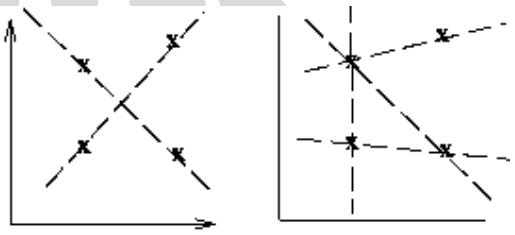
2.6.1 Algoritma *Circle Hough Transform*

Dalam [RFIS03] dijelaskan bahwa algoritma ini berawal dari pembacaan gambar dan dikonversikan ke dalam bentuk *binary* untuk mendapatkan koordinat-koordinat titiknya seperti terlihat pada gambar berikut:



Gambar 2. 3 Koordinat Titik

Lalu dari koordinat-koordinat titik tersebut dicari kemungkinan-kemungkinan garis yang mungkin terjadi dan dicari persamaannya. Hal ini terlihat pada gambar berikut:



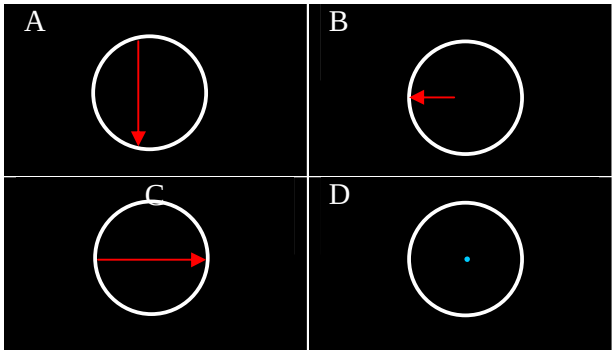
Gambar 2. 4 Kemungkinan Garis

Koordinat titiknya dapat berupa koordinat Kartesius atau koordinat Kutub, dan koordinat tersebut bisa dalam bentuk dua dimensi atau tiga dimensi. Dengan mengetahui persamaan garis tersebut, maka bentuk kurva dari koordinat titik tersebut dapat dibuat sehingga dapat diketahui gambar yang dibentuk dari koordinat titik-titik tersebut.

2.6.2 Algoritma Titik Tengah

Dalam [Albe06] dijelaskan tentang tahapan-tahapan dari algoritma ini yang terdiri dari:

1. Pencarian lingkaran dimulai dengan suatu titik pada gambar yang bukan *background*.
2. Diasumsikan titik tersebut terdapat pada sisi dari suatu lingkaran.
3. Kemudian dilakukan proses pencarian titik tengah dari lingkaran tersebut, yaitu dengan langkah-langkah sebagai berikut:



Gambar 2. 5 Proses Pencarian Titik Tengah

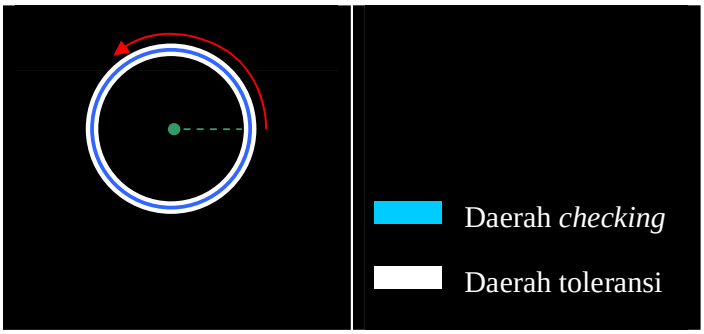
a. Telusuri gambar ke bawah sampai menemukan sisi lingkaran sambil menghitung jarak ke sisi lingkaran tersebut. Jika ada, maka diperoleh informasi mengenai titik tengah dari tinggi lingkaran, yaitu dari titik awal pergerakan ditambah dengan jarak/2. Jika tidak, maka objek bukanlah lingkaran.
b. Titik tengah yang diperoleh pada langkah sebelumnya belum tentu merupakan titik tengah dari lingkaran, tetapi hanya titik tengah dari tinggi lingkaran. Jadi selanjutnya akan dicari titik tengah dari lebar lingkaran.

   Dari titik tengah tinggi lingkaran tadi, telusuri gambar ke kiri sampai menemukan sisi tanpa menghitung jarak ke sisi tersebut. Jika tidak ditemukan, maka objek bukanlah lingkaran.

c. Dari titik yang diperoleh pada tahap (b), telusuri gambar ke kanan sampai menemukan sisi sambil menghitung jarak ke sisi tersebut. Jika ada, maka diperoleh informasi mengenai titik tengah dari lebar lingkaran, yaitu dari titik awal pergerakan ditambah dengan jarak/2. Jika tidak, maka objek bukanlah lingkaran.
d. Diperoleh informasi mengenai titik tengah dan radius dari objek tersebut.

4. Jika radiusnya lebih besar dari toleransi, maka dilakukan identifikasi objek.

5. Identifikasi objek dilakukan dengan langkah-langkah sebagai berikut:



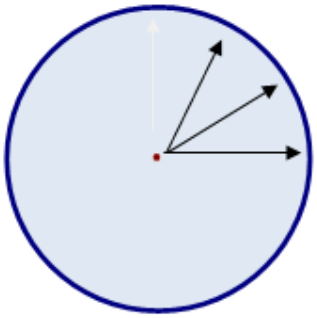
Gambar 2. 6 Identifikasi Objek

a. Dilakukan rotasi 360^{o} berlawanan arah jarum jam, yaitu dengan menggunakan *loop*.
b. Untuk setiap iterasi, dihitung titik (xt,yt), yaitu titik yang berjarak radius dari titik pusat dan memiliki sudut yang bersesuaian dengan iterasi yang sedang dilakukan.
c. Pada titik tersebut dan pada n-tetangga di sekitarnya (n adalah sebuah toleransi ketetanggaan), dilakukan pemeriksaan. Nilai n ini bergantung pada radius objek: semakin besar radius objek, semakin besar nilai n, dan sebaliknya. Jika pada salah satu dari titik-titik tersebut merupakan titik sisi, maka iterasi dilanjutkan. Jika tidak satupun dari titik-titik tersebut yang merupakan titik sisi, maka objek bukan lingkaran.
d. Jika iterasi berakhir dengan sukses, maka objek adalah lingkaran.

2.6.3 Algoritma *Circle Detection*

Dalam [ARIO06] dijelaskan tentang algoritma *Circle Detection* yang diawali dengan deteksi sisi. Saat melakukan deteksi sisi akan kita dapatkan himpunan semua titik-titik yang termasuk sisi dalam suatu citra. Kemudian kita bagi titik-titik tersebut ke dalam kelas ekivalensi dimana setiap kelas ekivalensi merepresentasikan satu ‘komponen yang terkoneksi’ (untuk selanjutnya dalam dokumen ini akan dirujuk dengan sebutan komponen). Pendefinisian komponen itu sendiri dengan menggunakan jendela delapan ketetanggaan.

Setelah mendapatkan komponen tersebut, kita mendapatkan sekumpulan titik yang terkait satu sama lain. Kemudian dari titik-titik tersebut pasti membentuk satu komponen bentuk tertentu yang mungkin adalah lingkaran. Setiap bentuk memiliki ciri yang khusus yang dapat digunakan untuk mengenali bentuk tersebut, begitu juga dengan lingkaran. Ciri khusus ini dapat dilihat pada gambar di bawah ini:



Gambar 2. 7 Ciri Khusus Objek Lingkaran

Tampak pada gambar bahwa jarak antara titik pusat lingkaran dengan titik-titik yang ada pada sisi adalah sama panjang. Namun karena pada citra awal dapat terjadi distorsi gambar yang mungkin diakibatkan oleh berbagai aspek, maka definisi “sama panjang” perlu didefinisikan secara khusus. Untuk menangani hal tersebut, maka kelompok kami memberikan ambang toleransi suatu besaran dikatakan sama panjang.

2.6.4 Algoritma Pusat Massa dan Titik Berat

Dalam [ABAR06] dijelaskan tentang algoritma Pusat Massa dan Titik Berat dengan uraian sebagai berikut:

Ide Dasar

Mencari sebuah pusat lingkaran dengan menggunakan konsep Fisika tentang pusat massa dan titik berat. Sebelumnya akan dipaparkan sekilas penjelasannya.

Pusat Massa dan Titik Berat

Pusat massa dan titik berat suatu benda memiliki pengertian yang sama, yaitu suatu titik tempat berpusatnya massa/berat dari benda tersebut. Perbedaannya adalah letak pusat massa suatu benda tidak dipengaruhi oleh medan gravitasi, sehingga letaknya tidak selalu berhimpit dengan letak titik beratnya.

1. **Pusat Massa**

   Koordinat pusat massa dari benda-benda diskrit, dengan massa masing-masing M_1, M_2,..... , M_i ; yang terletak pada koordinat (x_1,y_1), (x_2,y_2),....., (x_i,y_i) adalah:

$$x = \frac{\left(\sum_{k=1}^{i} M_k \bullet x_k\right)}{\sum_{k=1}^{i} M_k} \text{ dan } y = \frac{\left(\sum_{k=1}^{i} M_k \bullet y_k\right)}{\sum_{k=1}^{i} M_k}$$

Gambar 2. 8 Persamaan Pusat Massa

2. **Titik Berat**

   Koordinat titik berat suatu sistem benda dengan berat masing-masing W_1, W_2, ..., W_i ; yang terletak pada koordinat (x_1,y_1), (x_2,y_2), ..., (x_i,y_i) adalah:

$$x = \frac{\left(\sum_{k=1}^{i} W_k \bullet x_k\right)}{\sum_{k=1}^{i} W_k} \text{ dan } y = \frac{\left(\sum_{k=1}^{i} W_k \bullet y_k\right)}{\sum_{k=1}^{i} W_k}$$

Gambar 2. 9 Persamaan Titik Berat

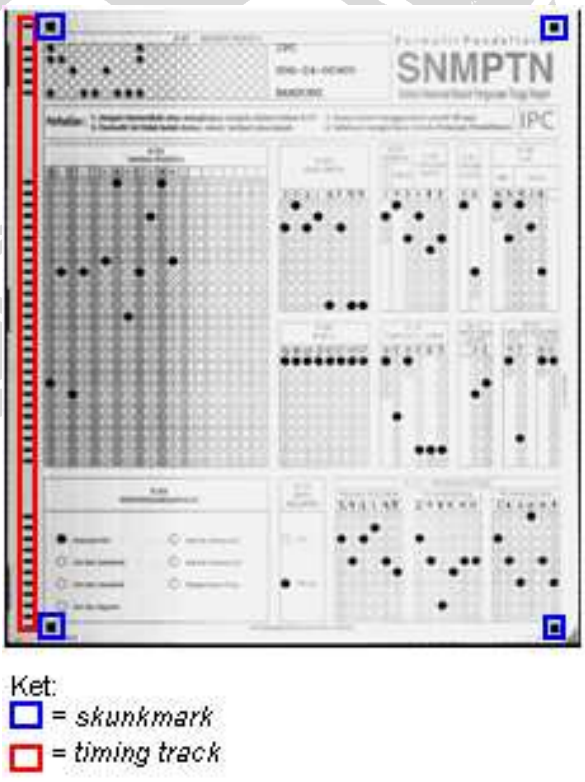
Asumsi dengan menggunakan kosep pusat massa dengan memberikan massa M_i yang bernilai satu atau konsep titik berat dengan memberikan berat W_i yang bernilai satu, maka untuk setiap titik-titik pinggir yang membingkai sebuah lingkaran dengan menerapkan rumus di atas akan didapatkan letak koordinat pusat lingkaran.

Permasalahannya adalah titik-titik yang membingkai sebuah lingkaran pada citra digital tidaklah semulus citra analog. Jadi, apakah dengan didapatkan letak koordinat pusat lingkaran sudah menjamin kebenarannya. Di sinilah dibutuhkan konsep statistika, variansi.

Penerapan variansi digunakan untuk menyelidiki selisih koordinat pusat lingkaran dengan titik-titik yang membingkai sebuah bangun yang diduga lingkaran. Seharusnya sebuah bangun lingkaran memiliki karakteristik variansi dari selisih tersebut yang nilainya tidak besar.

2.6.5 Algoritma Deteksi *Timing Track* dan *Skunk Mark*

Algoritma ini adalah algoritma yang khusus dibuat untuk mendeteksi lingkaran yang dihitamkan pada lembar LIK yang memiliki penanda khusus. Pada formulir SPMB penanda tersebut adalah *timing track* dan *skunk mark*. Untuk lebih jelasnya perhatikan gambar berikut.



Gambar 2. 10 *Skunk mark* dan *timing track*

Algoritma ini mendeteksi *timing track* dan *skunk mark* untuk mendapatkan koordinat-koordinat kolom dan baris dari data pada formulir LIK yang akan diekstrak, kemudian dibuat matriks berdasarkan koordinat-koordinat *timing track* dan *skunkmark* melingkupi daerah yang dibatasi koordinat-koordinat tersebut.

Setelah matriks dibuat, dilakukan pemeriksaan nilai *boolean* dari matriks tersebut (bernilai 0 atau 1). Setelah itu, dilakukan pengelompokkan matriks untuk mempermudah mengekstrak data dari LIK. Pengelompokan matriks dilakukan berdasarkan format kerangka tertentu. Setelah matriks dikelompokkan menurut format tertentu, dilakukan pengekstrakan data dengan memetakan nilai matriks dengan *value* yang telah didefinisikan sesuai format kerangka yang bersesauaian dengan area sel (grid) matriks tersebut.

Algoritma ini pernah diimplementasikan sebelumnya pada sistem "*Scanner Project*" terdahulu, namun belum berfungsi dengan benar. Tim pengembang mencoba menyempurnakan algoritma tersebut agar berfungsi dengan benar sehingga dapat dilakukan perbandingan dengan algoritma-algoritma yang lain untuk menemukan algoritma terbaik yang akan diimplementasikan pada sistem "*Scanner Project*" ini.