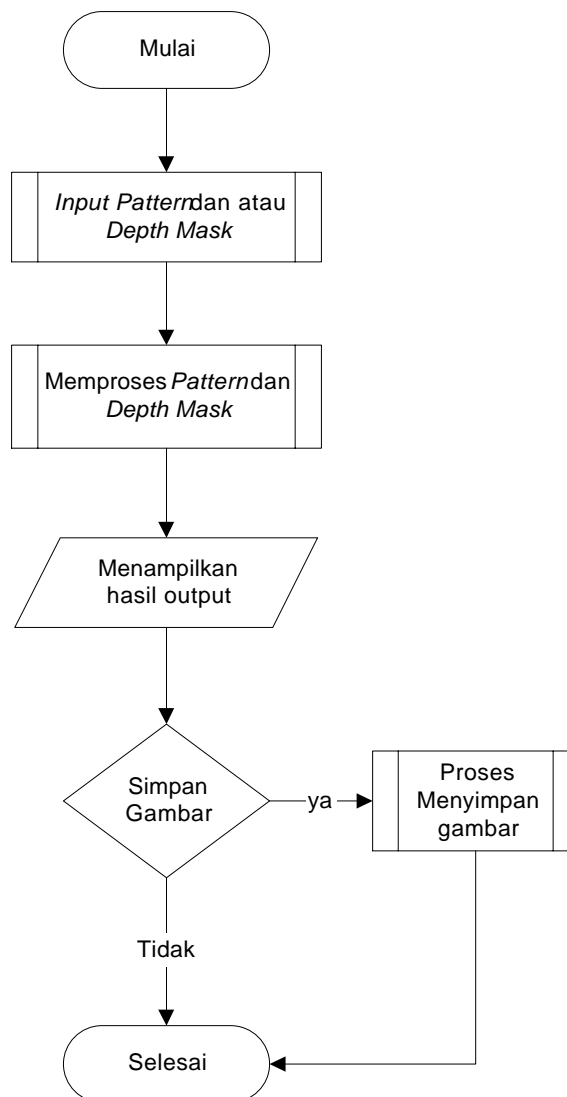


3. DESAIN SISTEM

Pada bab ini akan dibahas perencanaan dan sistem kerja dari perangkat lunak untuk menghasilkan gambar *stereogram*. Secara garis besar, sistem terdiri dari bagian *input* yang terdiri dari *pattern* dan *depth mask*. Setelah selesai melakukan *input*, *user* bisa memulai proses untuk menghasilkan gambar *stereogram*.

3.1. Garis Besar Sistem Kerja Perangkat Lunak

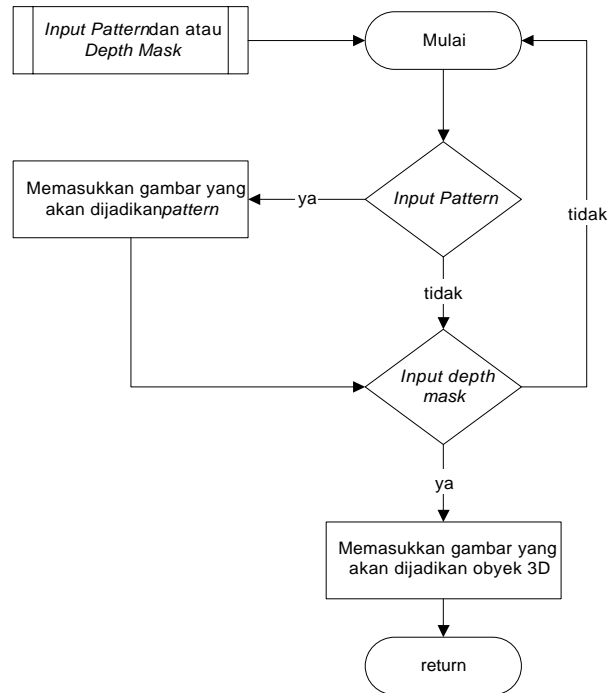


Gambar 3.1. Garis Besar Sistem Kerja Perangkat Lunak

Keterangan:

- ✍ Gambar 3.1 diatas menunjukkan garis besar sistem yang dibuat dan juga garis besar dari semua proses yang terjadi pada perangkat lunak yang dibuat.
- ✍ Di awal sistem, *user* harus meng-*input*-kan minimal hanya *depth mask*-nya saja. Bila tidak, maka program tidak akan dapat berjalan.
- ✍ Sistem untuk menghasilkan gambar *stereogram* dengan titik *random* sebagai *backgorund* hanya akan berjalan, jika *user* telah memberi *depth mask* sebagai *input*.
- ✍ Sistem untuk menghasilkan gambar *stereogram* dengan *pattern* sebagai *backgorund* hanya akan berjalan, jika *user* telah memberi 2 gambar yang terdiri dari *pattern* dan *depth mask*.
- ✍ Setelah kedua *input* dimasukkan, maka program akan memproses dengan otomatis hingga *output* dihasilkan.
- ✍ *Output* akan benar-benar selesai diproses bila ada pemberitahuan berupa *showmessage* yang memberikan informasi bahwa proses sudah selesai.
- ✍ Setelah gambar *output* ditampilkan, *user* dapat menyimpan gambar dalam *format* BMP atau *format* JPEG.
- ✍ Bila *user* tidak ingin menyimpan gambar, *user* dapat menutup program atau memulai proses dari awal lagi.

3.2. Proses memasukkan *pattern* dan atau *depth mask*

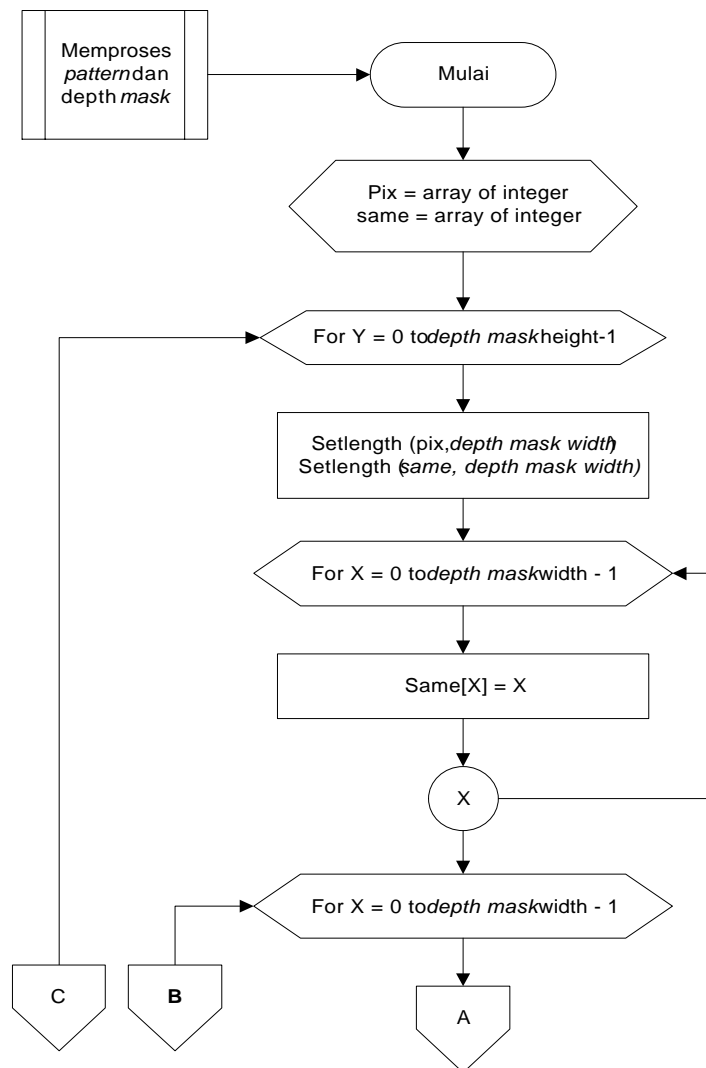


Gambar 3.2. Proses memasukkan *input* dan atau *pattern*

Keterangan:

- ✍ Gambar 3.2 diatas menunjukkan proses yang terjadi pada saat memasukkan *depth mask* dan atau *pattern*.
- ✍ User dapat memilih untuk memasukkan gambar berupa *pattern* yang nantinya akan digunakan sebagai *background* dan pola dari obyek yang akan ditampilkan pada gambar *output* atau tidak memasukkan *pattern*.
- ✍ Meskipun *user* tidak memasukkan *pattern*, *user* tetap harus memasukkan *depth mask* karena *depth mask* tersebut akan digunakan untuk menampilkan obyek 3D pada gambar *output*.

3.3. Proses Pengolahan *Pattern* dan *Depth mask*

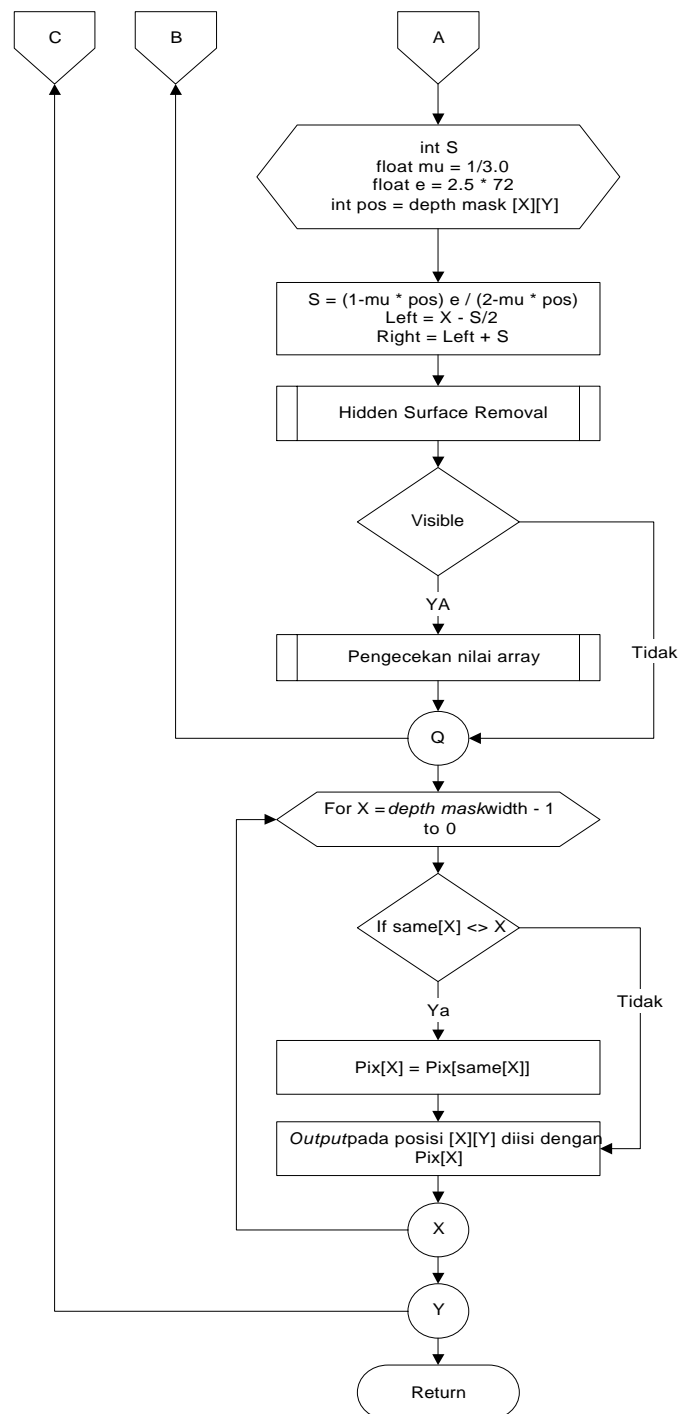


Gambar 3.3. Proses Pengolahan *Pattern* dan *Depth mask*

Keterangan:

- ✎ Gambar 3.3 diatas menunjukkan apa yang terjadi pada proses pembuatan gambar *stereogram* dengan menggabungkan *pattern* dan *depth mask*.
- ✎ Variabel *pix* adalah *array* yang akan digunakan untuk menyimpan *background* yang berupa *pattern* pada gambar *output*.
- ✎ Variabel *same* adalah *array* yang akan digunakan untuk menyimpan posisi *pixel* yang seharusnya digambar pada gambar *output*

- ✍ Pertama kali, variabel *pix* dan *same* diubah ukuran panjangnya agar sesuai dengan lebar dari gambar *depth mask*.
- ✍ Tiap *index* dari variabel *same* diinisialisasikan dengan nilai yang sama dengan posisi *index*-nya.

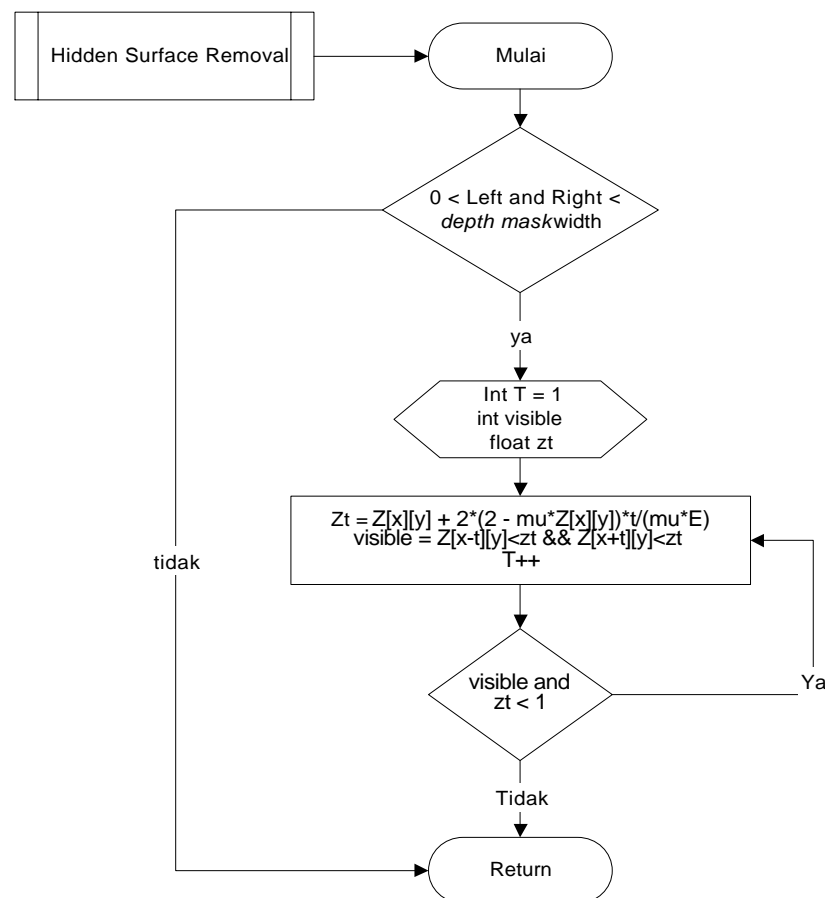


Gambar 3.4. Lanjutan Proses Pengolahan *Pattern* dan *Depth mask*

Keterangan :

- ✍ Gambar 3.4 menunjukkan urutan kerja dari sistem perangkat lunak untuk menghasilkan gambar *stereogram*.
- ✍ Pertama kali dilakukan inialisasi untuk variabel awal yang nantinya akan digunakan untuk proses selanjutnya.
- ✍ Variabel E adalah variabel yang diperoleh dari jarak antar mata (diasumsikan 2,5 *inchi*) dan nilai 72 adalah jumlah pixel di tiap *inchi*
- ✍ Di dalam keseluruhan proses tersebut, terdapat beberapa *sub*-proses untuk menghasilkan gambar *stereogram*. *Sub*-proses tersebut adalah proses *hidden surface removal* dan pengecekan nilai *array*.
- ✍ Setelah proses selesai dilakukan maka dilakukan hasilnya akan terlihat di gambar *output*.

3.4. Proses *Hidden Surface Removal*

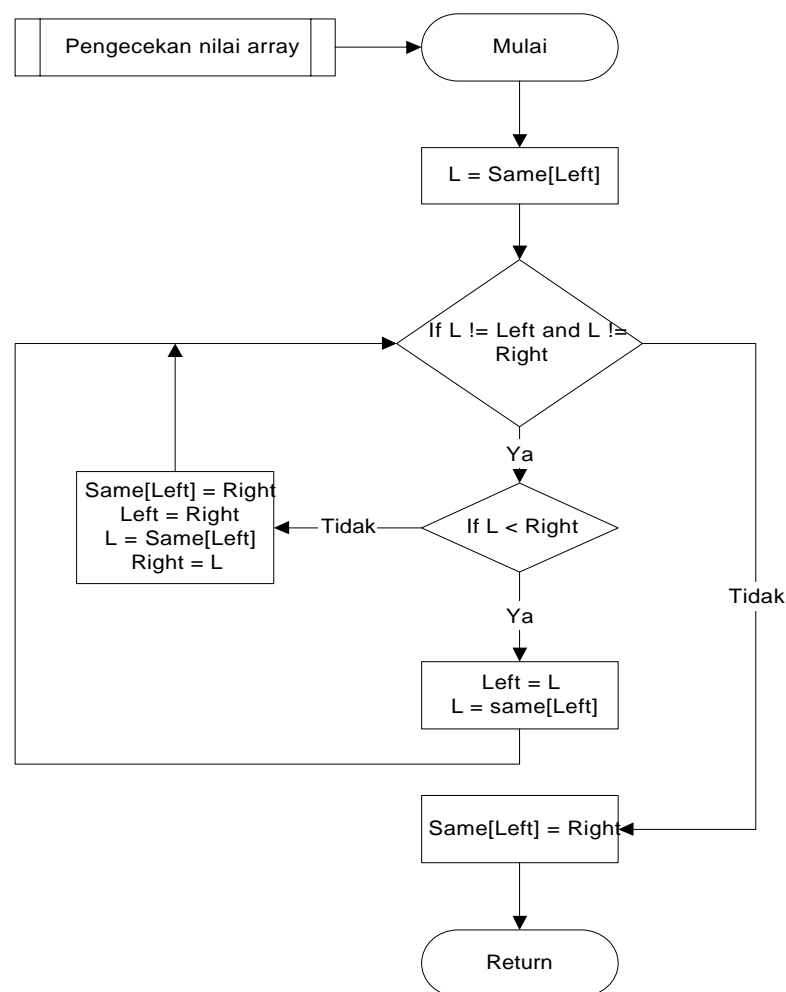


Gambar 3.5. Proses *Hidden Surface Removal*.

Keterangan :

- ✍ Proses *Hidden Surface Removal* adalah proses yang cukup penting di sistem perangkat lunak ini. Proses ini akan mem-*filter pixel* yang seharusnya tidak ditampilkan.
- ✍ Di dalam proses ini terdapat perhitungan secara matematis sesuai yang telah dijelaskan di bab 2.

3.5. Proses Pengecekan Nilai Array

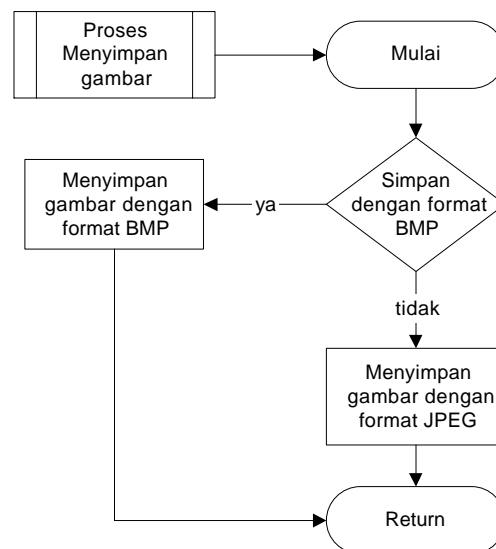


Gambar 3.6. Proses Pengecekan Nilai Array

Keterangan :

- ✍ Pada proses ini dilakukan pengecekan nilai *array* dimana pada *array* tersebut adalah nilai *pixel* yang nantinya akan menjadi *output* di tiap barisnya.
- ✍ Nilai dari *array* tersebut diperiksa pada posisi *left* dan *right* karena nilai pada posisi *left* hanya akan dilihat oleh mata kiri dan nilai pada posisi *right* hanya akan dilihat oleh mata kanan.

3.6. Proses Menyimpan gambar



Gambar 3.7. Proses Menyimpan Gambar

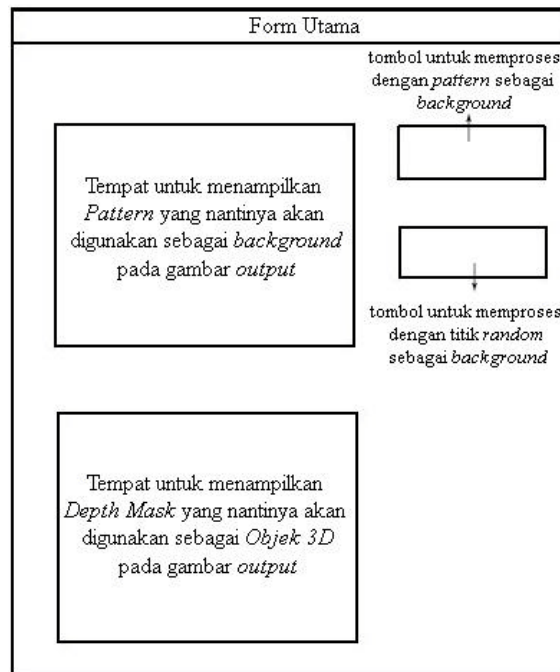
Keterangan :

- ✍ Pada proses ini *user* dapat menyimpan gambar yang telah diproses ke dalam format BMP atau JPEG.

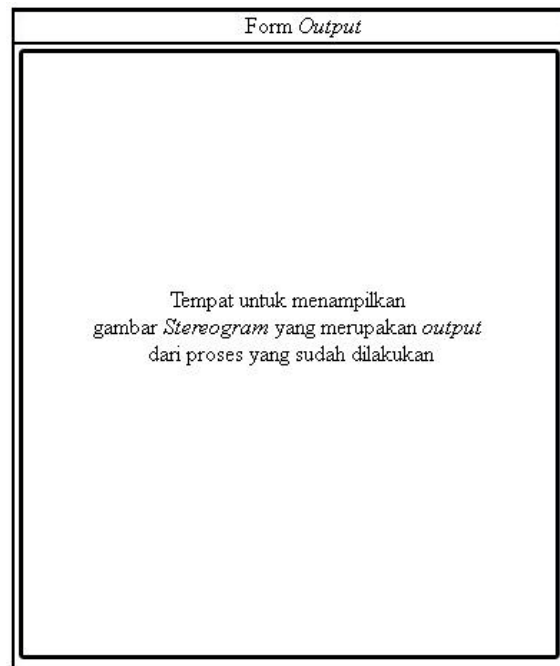
3.7. Desain Interface

Pada aplikasi ini terdapat dua *form*, yaitu *form* utama yang akan digunakan untuk meng-*input*-kan gambar *pattern* yang akan digunakan untuk sebagai *background* pada gambar *output* dan gambar *depth mask* yang akan dijadikan

obyek 3D pada gambar *output*. *Form* kedua adalah *form* yang hanya akan berisi gambar *output* dari program yang telah diproses.



Gambar 3.8. Desain *Interface Form Utama*



Gambar 3.9. Desain *Interface Tampilan Output*

Pada *form utama* dan *form output*, tidak terdapat *menu* karena program ini memasukkan gambar *input* dengan cara melakukan *click* pada bagian yang dapat dilihat dengan jelas. Setelah memasukkan *input*, user dapat langsung memproses untuk menghasilkan gambar *output* dengan cara melakukan *click* pada tombol yang tersedia.

Form output akan muncul pada saat proses. Form ini akan menampilkan gambar *output* dan setelah gambar *output* selesai di-*generate*, *user* dapat melakukan proses penyimpanan gambar dengan melakukan *click* kanan pada mouse dan menyimpan sesuai dengan *format* yang diinginkan (BMP atau JPEG).