**LAPORAN UAS PENGOLAHAN CITRA DIGITAL**

**“SEGMENTASI GAMBAR”**

****

**DISUSUN OLEH :**

**Parhan Pauzi Salim**

**F55120095**

**Kelas C**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

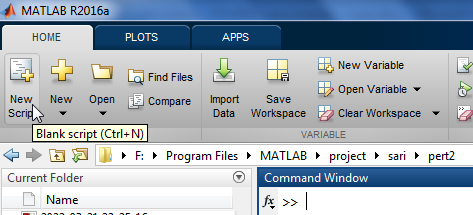
**JURUSAN TEKNOLOGI INFORMASI**

**FAKULTAS TEKNIK**

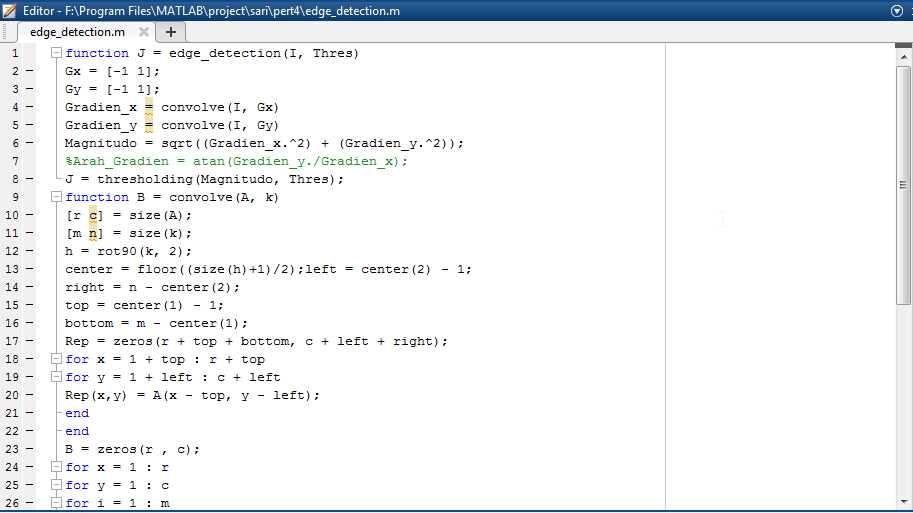
**UNIVERSITAS TADULAKO**

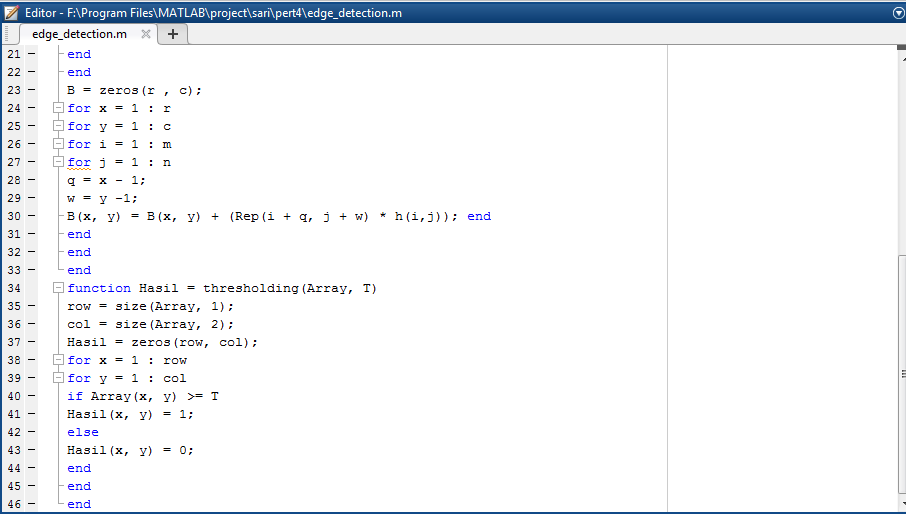
**2022**

1. **LANGKAH KERJA**
2. **Operator Gradien Pertama**
3. **Operator Sobel**
4. Buat *project* baru dengan klik *“New”* kemudian pilih *“Script”.*

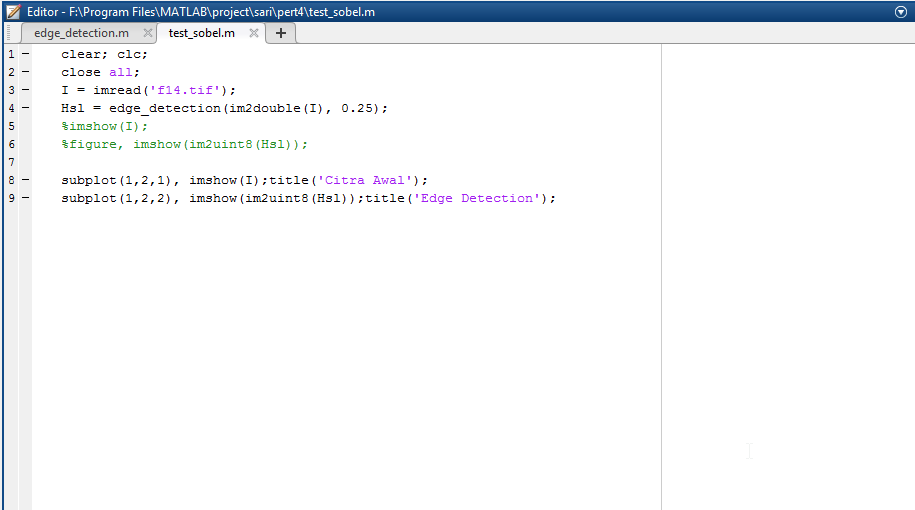


1. Ketikkan *script* berikut ini

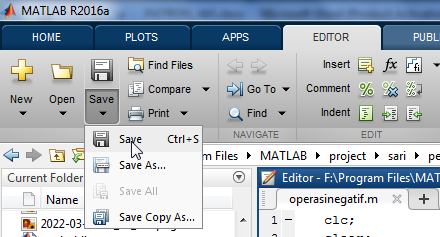




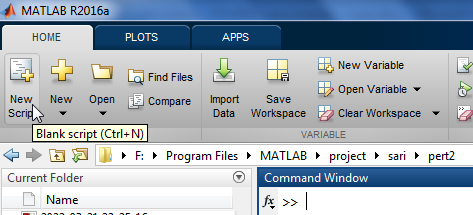
1. Buat lagi satu *file* baru dan ketikkan script berikut ini sebagai *source code* input nya



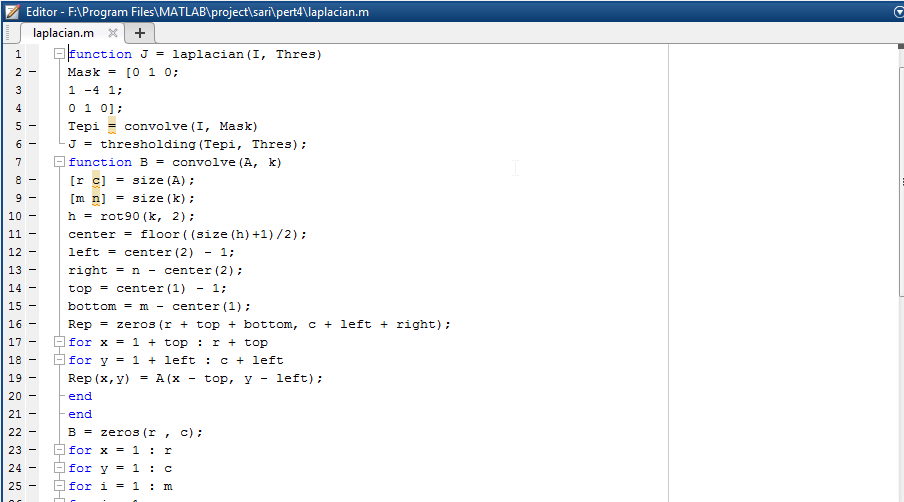
1. Menyimpan kode program dengan format .m.

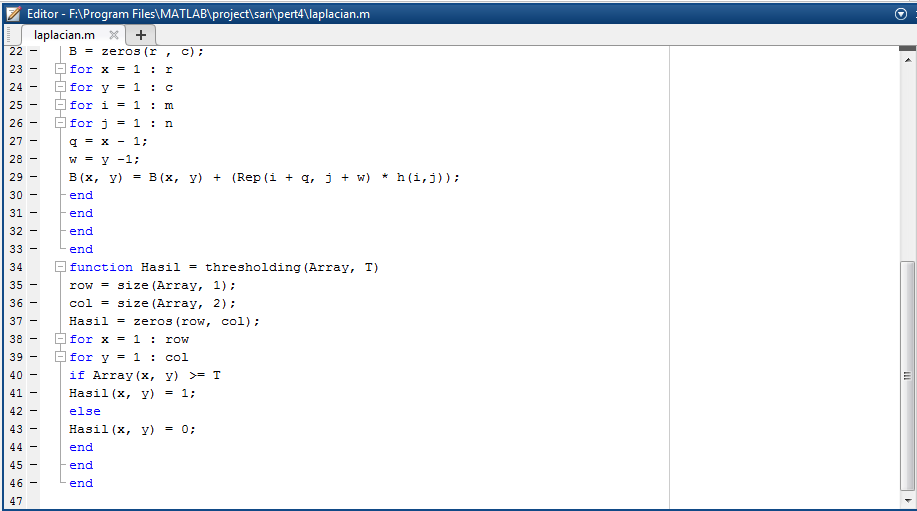


1. **Operator Turunan Kedua**
2. **Laplacian of Gaussian (LOG)**
3. Membuat *project* baru dengan mengklik *“New”* kemudian pilih *“Script”.*

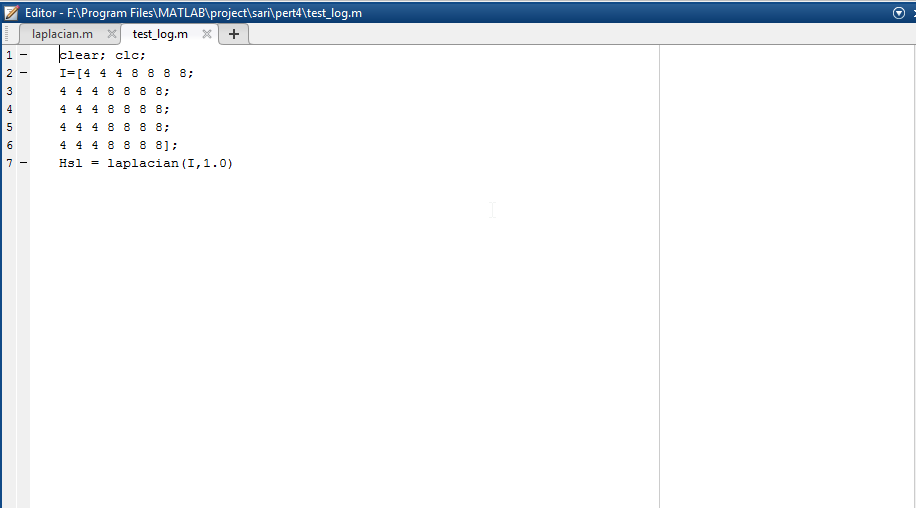


1. Ketikkan kode program untuk *source code* function nya

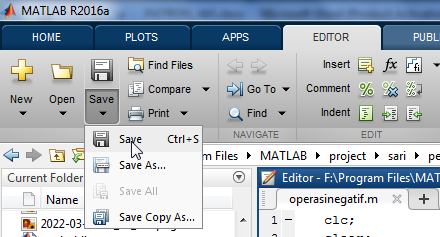




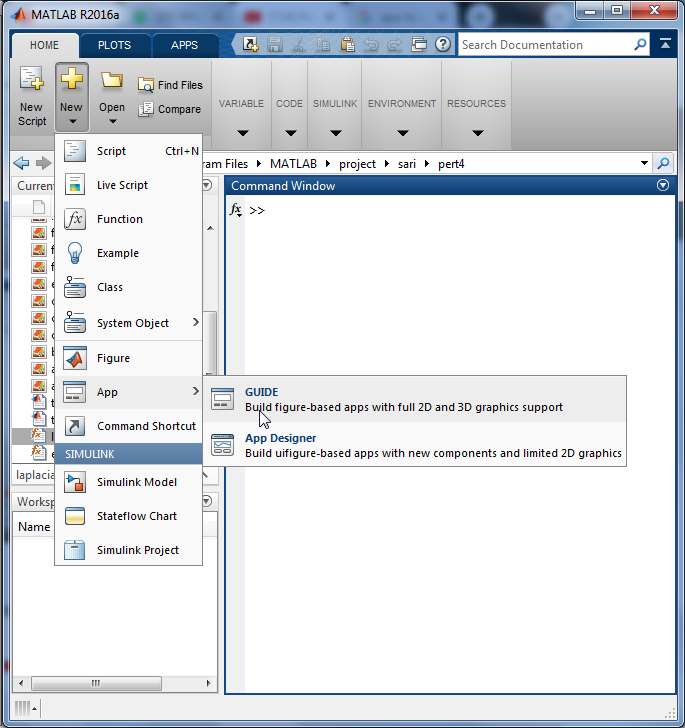
1. Lalu buat lagi satu *file* baru untuk mengetikkan *source code* input nya



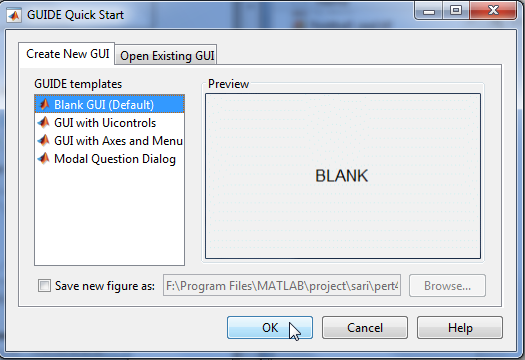
1. Menyimpan *file* kode program dengan format .m.



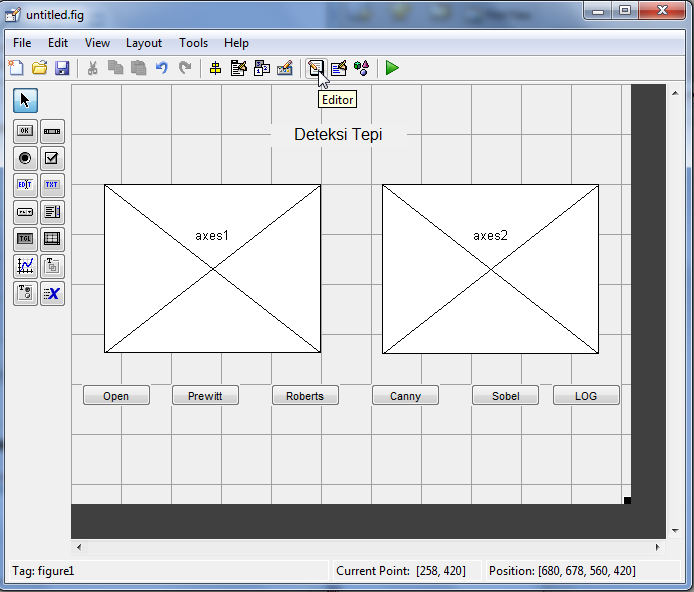
1. **Aplikasi Deteksi Tepi**
2. Buat *app guide* baru dengan cara seperti ini



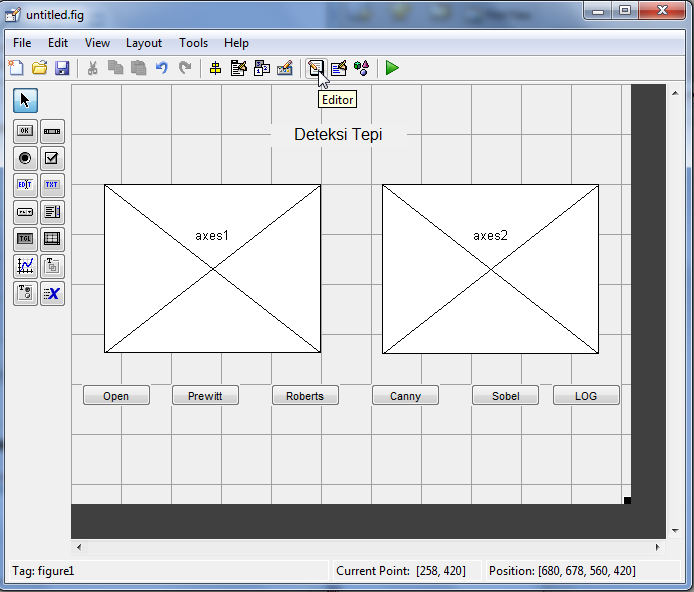
1. Lalu pilih *Blank GUI*



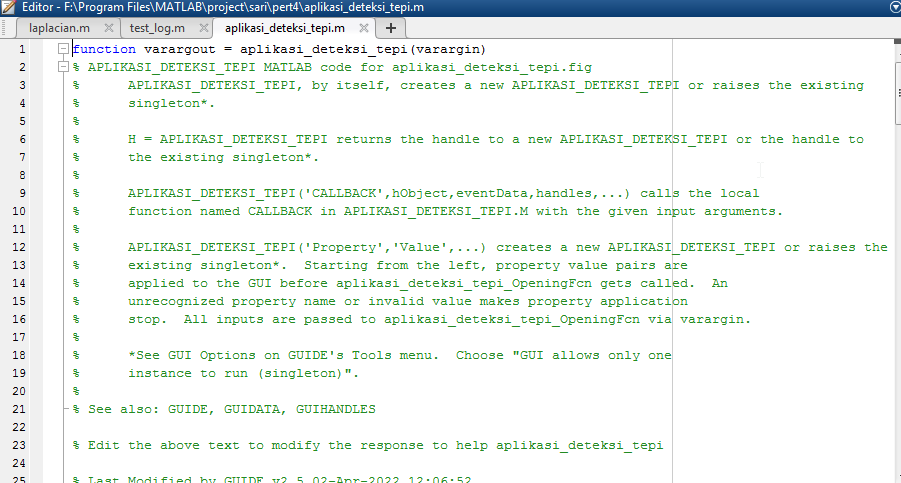
1. Lalu atur tampilan aplikasinya menjadi seperti ini



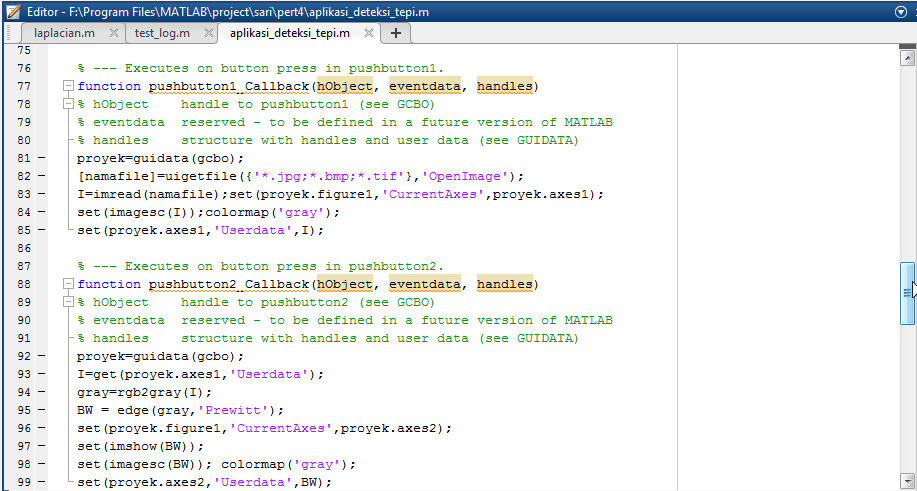
1. Setelah itu klik editor untuk menyimpan desain aplikasi dan sekaligus membuka *source code* nya



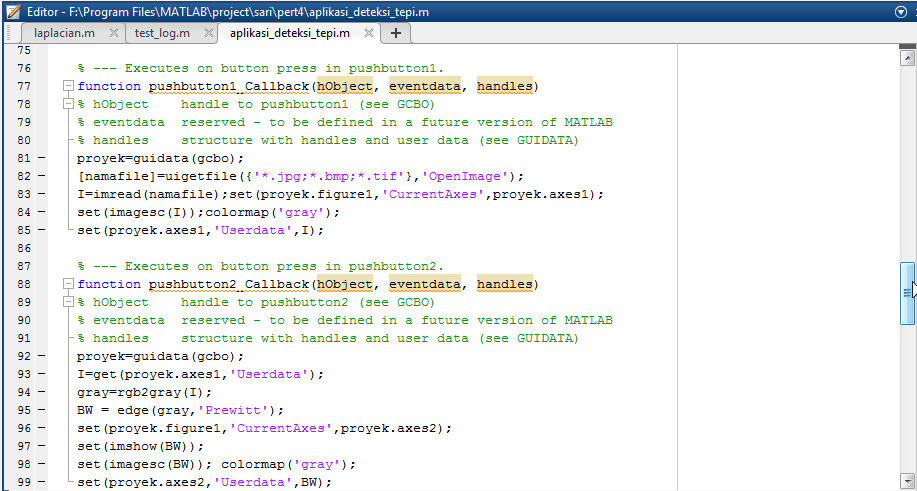
1. Maka akan muncul tampilan seperti ini



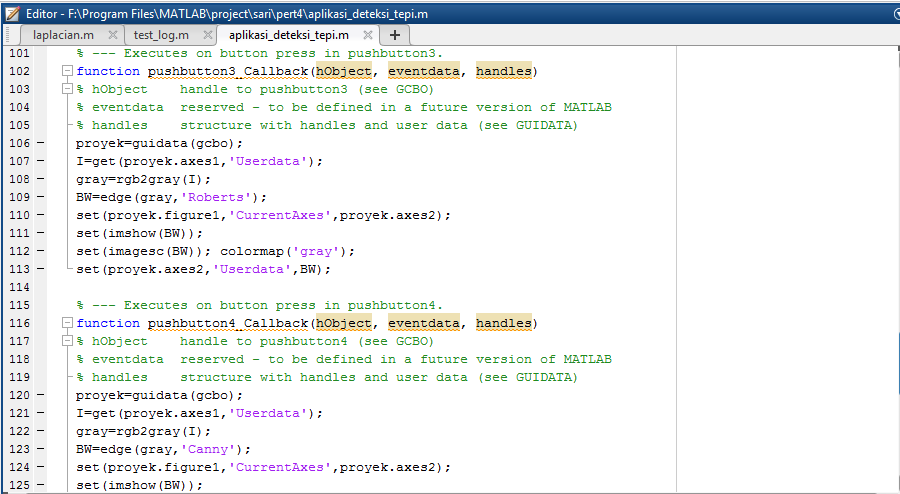
1. Ketikkan *source code* open seperti ini



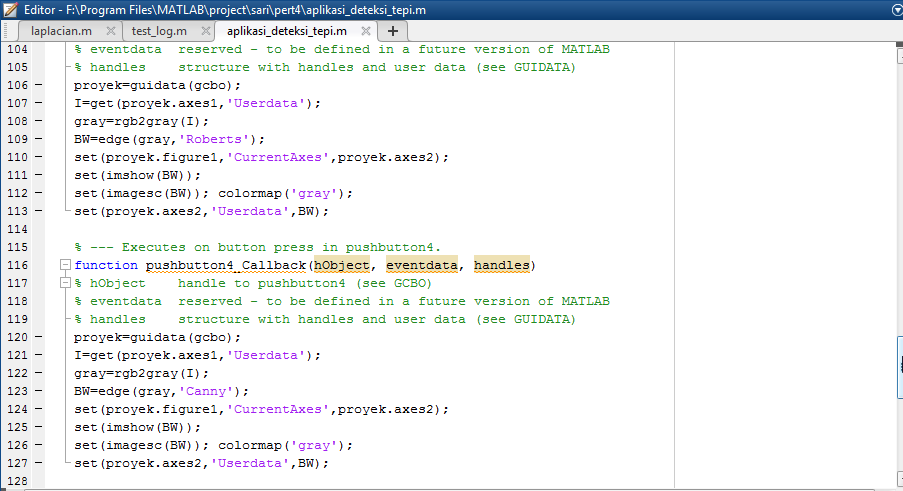
1. Lalu, untuk *source code* operator *Prewitt* ketikkan *script* ini



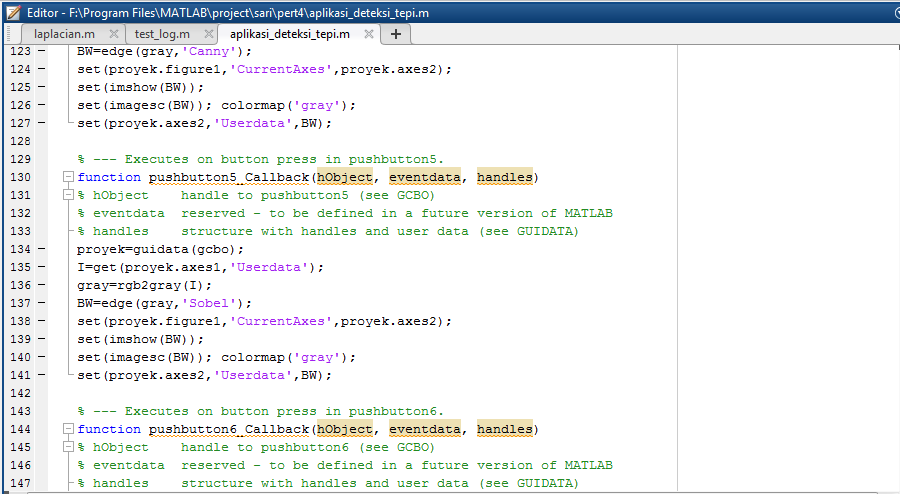
1. Kemudian, ketikkan *script* berikut ini untuk *source code* *Roberts*



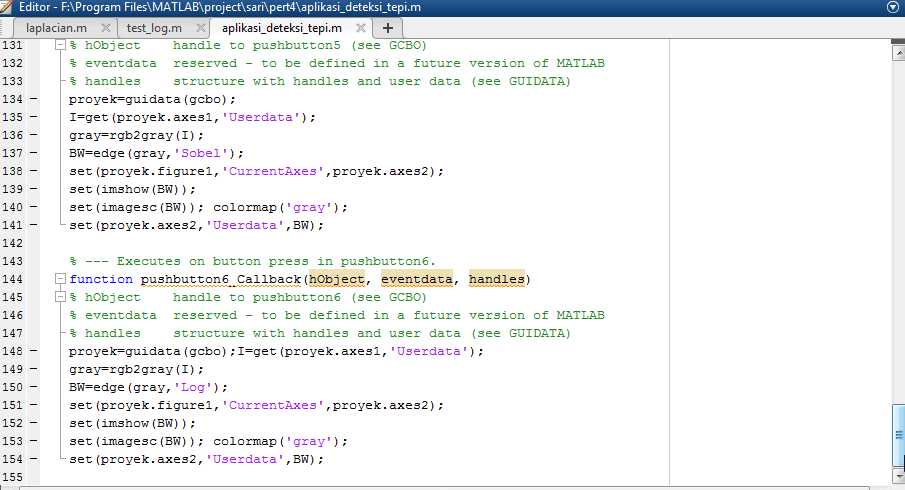
1. Untuk *source code* operator *Canny*, ketikkan *script* berikut ini



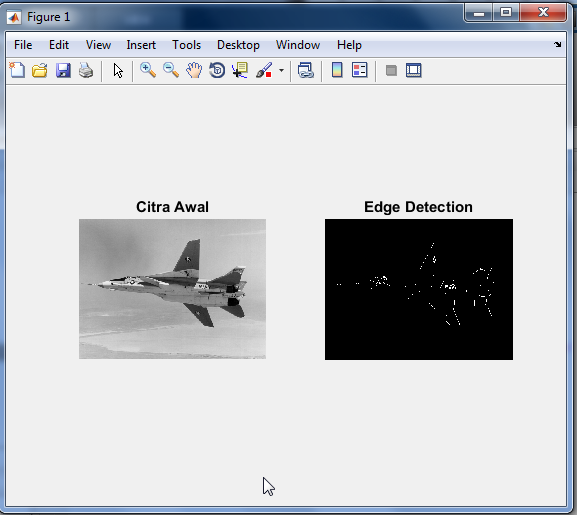
1. Selanjutnya, untuk operator *Sobel* ketikkan *script* berikut ini



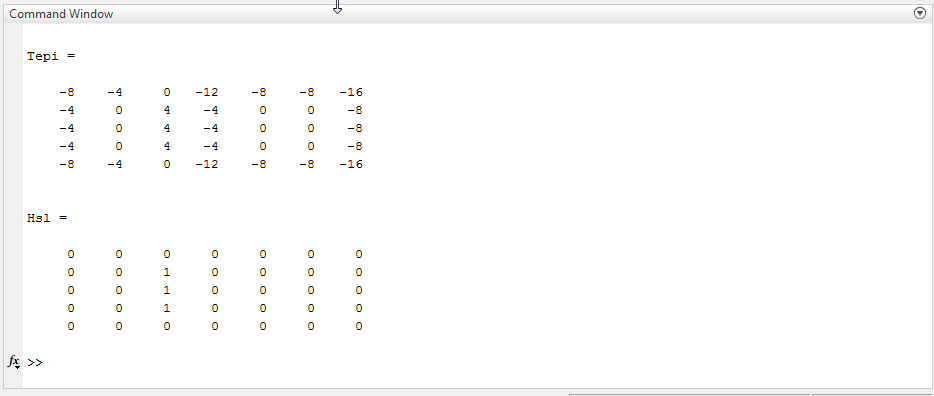
1. Terakhir, untuk operator *LOG* ketikkan *script* berikut ini



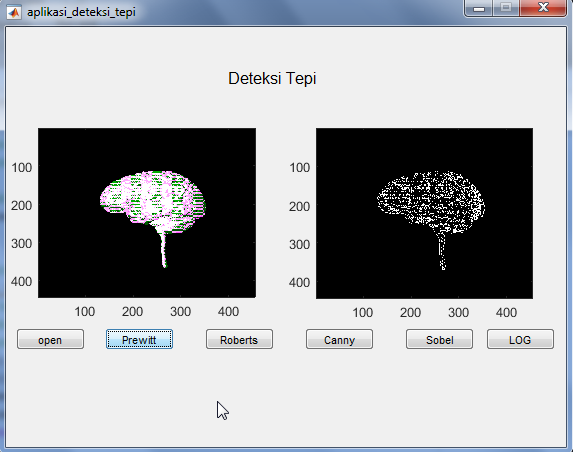
1. **HASIL PERCOBAAN**
2. **Operator Gradien Pertama**
3. **Operator Sobel**



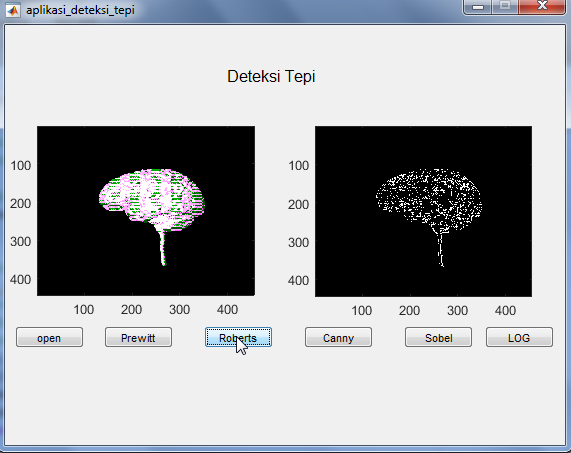
1. **Operator Turunan Kedua**
2. **Operator Laplacian of Gaussian (LOG)**

****

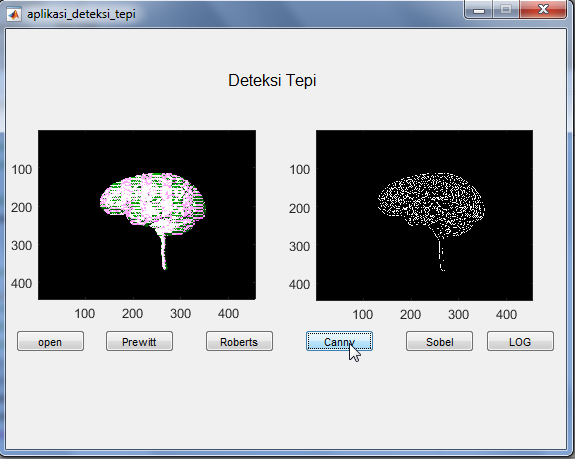
1. **Aplikasi Deteksi Tepi**
2. **Prewitt**

****

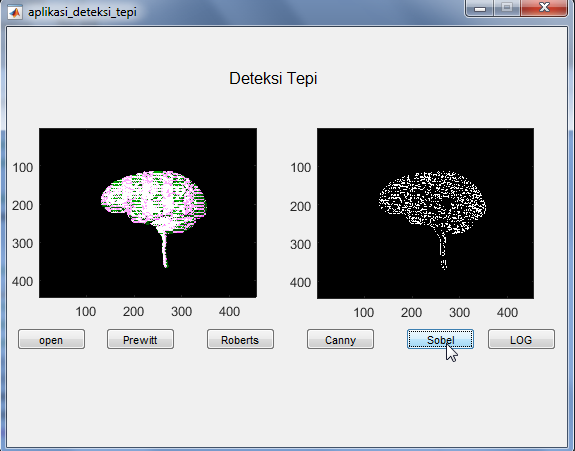
1. **Roberts**

****

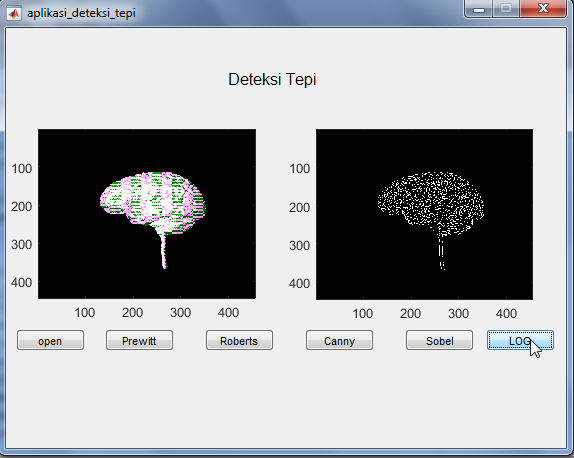
1. **Canny**

****

1. **Sobel**

****

1. **Laplacian of Gaussian (LOG)**

****

1. **ANALISIS**

Berdasarkan percobaan tersebut dapat dianalisis bahwa dalam *edge detection* ada beberapa operator namun yang pertama dipraktikan di sini adalah operator *sobel* dan operator *LOG*. Lalu berikutnya juga akan dibuat sebuah aplikasi *GUI* sederhana tentang deteksi tepi ini yang menyajikan operator-operator lainnya juga.

Yang pertama dilakukan adalah tentunya menyiapkan sebuah gambar, dan gambar yang digunakan disini merupakan gambar dengan ekstensi *.tif* dengan resolusi 640x480 piksel. Gambar ini di berasal dari laptop lalu kemudia di paste ke file projek ini agar gambarnya dapat keluar. Gambar ini disiapkan untuk praktik operator *sobel*. Untuk operator *sobel* ini dibuat dua *file* dulu, yang pertama adalah *file* *edge\_detection.m* yang berfungsi sebagai *source code functionnya*. Di sini dideklarasikan “*function J=edge\_detection(I, Thres)”* yang maksudnya adalah mendeteksi tepi pada gambar I dengan menggunakan *edge-detection* dengan metode yang spesifik. Kemudian, terdapat “*Gx=[1 -1]*” dan “*Gy[1 -1]*” yang mana kode ini maksudnya adalah berupa matriks yang diberi konvolusi *kernel* dari operator *sobel* ini. Lalu kemudian terdapat kode “*Magnitudo=sqrt((Gradien\_x.^2)+(Gradien\_y.^2));*” yang mana kode ini merupakan penentu atas arah gradiennya. Berikutnya, diberikan “*function B=convolve(A,k)*” yang merupakan pendefinisian konvolusinya. Di sini juga dibandingkan antara tengah, kanan, kiri, atas dan bawah. Setelah itu terdapat juga “*function Hasil=thresholding(Array, T)”* yang memberikan segmentasi *thresholding* pada hasil akhirnya. Setelah membuat *file* untuk *function* nya berikutnya adalah membuat *file* baru untuk *source code* inputnya. Di sini dilakukan pemanggilan terhadap gambar yang sudah disiapkan sebelumnya. Yaitu dengan kode “*I=imread()*”. Lalu, diberikan kode dengan variabel “*Hsl=edge\_detection(im2double(I), 0.25);*” yang memberikan deteksi tepi pada gambar yang dipanggil tadi.

Pada operator yang kedua yaitu operator turunan kedua (*LOG*), di sini juga disiapkan dua *file* yaitu *file* untuk *function* dan *file* untuk *source code* inputnya. Untuk *function*, hampir sama seperti seperti program sebelumnya yaitu diberikan pendefinisian “*function J=laplacian(I, Thres)*” terlebih dahulu. Lalu kemudian diberikan lagi pendefinisian “*function B=convolve(A, k)*” untuk definisi konvolusinya. Di sini juga dilakukan perbandingan pada tengah, atas, kiri, kanan, dan bawah nya juga. Dan berikutnya juga diberikan “*function Hasil=thresholding(Array, T)”* yang berfungsi untuk memberikan segmentasi *thresholding*. Berikutnya, dibuat *file* untuk *source code* input nya dengan memberikan kode yang merupakan matriks dan untuk menentukan hasilnya diberikan kode “”*Hsl=laplacian(I,1,0)*” pada *source code* nya.

Lalu, untuk aplikasi deteksi tepi yang pertama dibuat adalah desain aplikasinya. Disini ditentukan dua *axes* sebagai wadah muncul gambar asli yang akan lakukan *edge detection*. Kemudian, diberikan enam buah *push button* yang terdiri dari satu *button* untuk membuka gambar dan lima *push button* untuk tombol operatornya. Setelah itu, diberikan lagi *source code* untuk masing-masing tombolnya. Untuk tombol pertama yaitu tombol *open*, di sini diberikan kemampuan tombol tersebut untuk bisa menginputkan gambar dengan beberapa ekstensi seperti *.jpg, .bmp*, dan *.tif*. Adapun kodenya yaitu *“[namafile]=uigetfile({‘\*.jpg;\*.bmp;\*.tif’}, ‘OpenImage’);*”. Lalu, untuk operator-operatornya secara garis besar mempunyai *script* yang sama yaitu pertama gambarnya diberikan citra *gray* dengan adanya kode “*gray=rgb2gray(I);*” lalu kemudian yang membedakannya adalah pada kode “*BW=edge(gray,’’)”.* Pada kode tersebut tinggal dimasukkan nama operatornya. Resolusi gambar yang dimasukkan di sini haruslah sesuai dengan resolusi dari *axes* nya agar operatornya bisa bekerja dengan baik.

1. **KESIMPULAN**

Berdasarkan percobaan tersebut dapat disimpulkan bahwa pada citra gambar bisa dilakukan teknik deteksi tepi atau *edge detection*. Dimana terdapat dua garis besar yaitu operator gradien pertama dan operator turunan kedua. Dan pada percobaan tersebut bisa diketahui bahwa operator *sobel* dan *LOG* dan juga beberapa operator lainnya sangatlah penting dan menarik untuk dipelajari dan dipahami. Olehnya, diharapkan praktikan bisa mempelajari dan memahami materi tersebut dengan baik.