PERANCANGAN DAN REALISASI MOBIL REMOTE CONTROL MENGGUNAKAN FIREBASE

DESAIN AND REALIZATION OF REMOTE CONTROL CAR USING FIREBASE

Dadan Nur Ramadan[1], Agus Ganda Permana[2], Hafidudin[3]
[1,2,3]Diploma 3 Teknik telekomunikasi, Fakultas Ilmu Terapan, Universitas Telkom
[1]dadan.nr@tass.telkomuniversity.ac.id, [2]agusgandapermana@ymail.com
[3]hafid@tass.telkomuniversity.ac.id

Abstrak
Mobil Remote Control adalah miniatur mobil dengan ukuran yang lebih kecil, yang dikendalikan secara langsung oleh remote atau *joystick* dengan menggunakan modul transmitter sebagai media komunikasi, pada jurnal ini dibahas perancangan dan realisasi mobil remote control berbasis Internet of things menggunakan Firebase *Cloud Messaging*, data dari aplikasi *joystick* pada *smartphone* dikirim ke *realtime database* terlebih dahulu, dan NodeMCU pada mobil RC melakukan singkronisasi data ke *realtime database*. Hasil pengujian diperoleh nilai rata-rata *delay* dari *realtime database* ke NodeMCU adalah 1.191 second, sedangkan nilai pengiriman data dari aplikasi *joystick* ke firebase adalah 3,13 Kb/s untuk *upload* dan *download* sebesar 2,28 Kb/s.

Kata kunci: *Internet of Things*, Firebase, *Realtime Database*, NodeMCU

Abstract
Car Remote Control is a smaller size miniature car, which is controlled directly by the remote or joystick using transmitter module as a communication, in this journal propose design and realization of car remote control based on Internet of things using Firebase Cloud Messaging, first data from joystick application sent to the realtime database, and NodeMCU on car RC synchronizes the data from realtime database. The results obtained average value of delay from realtime database to NodeMCU is 1.191 second, while the value of sending data from the joystick application to Firebase is 3.13 Kb/s for upload and 2.28 Kb/s for download.

Keywords: Internet of Things, Firebase, Realtime Database, NodeMCU

1. PENDAHULUAN

Mobil Remote Control atau Mobil RC, adalah miniatur mobil dengan ukuran yang lebih kecil dibandingkan aslinya, mobil RC dikendaliakan langsung oleh remote atau *joystick* dengan menggunakan modul transmitter. Beberapa penelitian sudah dilakukan pada perangkat RC, diantaranya pada metode pengontolnya yang menggunakan modul Bluetooth[1,2,3] dan wireless [4], dan suara menggunakan metode *Dual Tone Multi-Frequencies* (DTMF)[5,6].

Pada penelitian ini, mobil RC memanfaatkan teknologi berbasis *Internet of things* (IoT), dimana *remote control* menggunakan aplikasi *joystick* yang diinstall pada *smartphone* yang terhubung ke jaringan seluler, untuk dapat mengendalikan sebuah mobil RC yang didalamnya terdapat sebuah NodeMCU, dengan menggunakan *Firebase Cloud Messaging* (FCM) data dari aplikasi *joystick* dikirim ke *cloud* terlebih dahulu, dan NodeMCU pada mobil RC melakukan singkronisasi data, jadi data yang berasal dari aplikasi *joystick* tidak langsung dikirim ke mobil RC, akan tetapi disimpan terlebih dahulu di firebase menggunakan fitur firebase *realtime*

database, penggunaan metode IoT bertujuan utuk membuat aplikasi *joystick* pada *smartphone* dapat berkomunikasi menggunakan internet melalui jaringan seluler sebagai media transmisinya.

2. DASAR TEORI

2.1 *Internet of Things* (IoT)

Internet of Things (IoT) dapat didefinisikan sebagai infrastruktur jaringan yang dinamis dengan kemampuan mengkonfigurasi sendiri berdasarkan komunikasi protokol standar, dimana barang fisik dan virtual memiliki identitas dan karakteristik, dengan dukungan *cloud computing*, memungkinkannya untuk mengakses informasi dari Internet, menyimpan dan mengambil data dan selalu terhubung [7].

2.2 Firebase Realtime Database

Google Cloud Messaging (GCM) adalah layanan komunikasi *push cloud-to-device*, sejak terintegrasi dengan Firebase berubah namanya menjadi *Firebase Cloud Messaging* (FCM) biasa disebut Firebase[8].

Firebase memiliki beberapa fitur, diantaranya adalah *realtime database* yang disimpan secara *cloud*, layanan ini menggunakan *Application program interface* (API), data disimpan sebagai JSON dan disinkronkan secara realtime ke setiap klien yang terhubung, apabila ada perubahan pada data yang tersimpan, maka setiap user yang terhubung akan menerima pembaruan data secara otomatis[9]. Format waktu yang dapat dipergunakan pada Firebase adalah TIMESTAMP (*time since the Unix epoch*) dalam *milliseconds*[10].

Gambar 1. Fitur dari layanan *Firebase Realtime Database*[9].

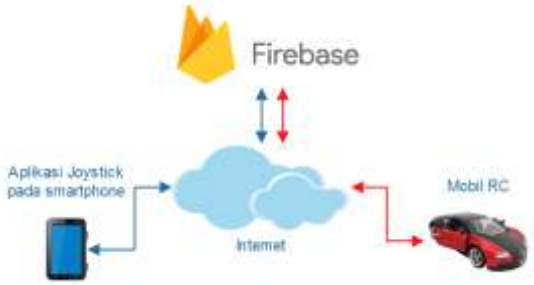
2.3 NodeMCU

NodeMCU adalah *firmware* berbasis *scripting* eLua untuk hardware ESP8266 WiFi SOC buatan Espressif system, *firmware* yang diguanakan berbasis pada Espressif NON-OS SDK dan mengugunakan sistem file berbasis pada spiff (SPI flash file system) untuk *embedded system* [11], NodeMCU dapat dipergunakan sebagai platform IoT karena sifat *firmware*-nya yang *opensource*.

Gambar 2. NodeMCU ESP8266 [12].

3. Mobil Remot Control Menggunakan Firebase

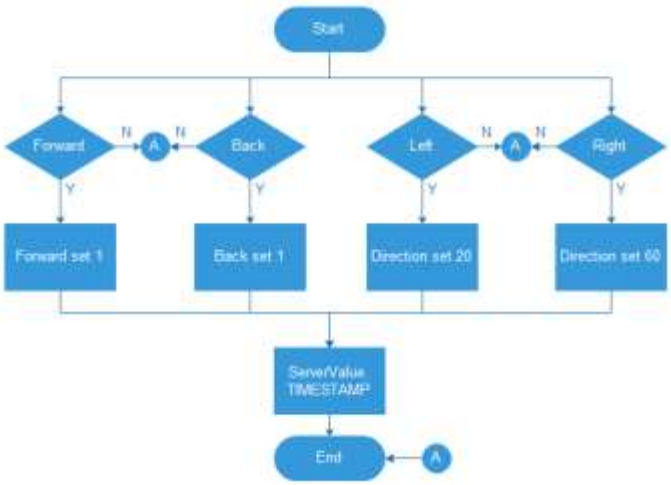
Dalam proses merealisasikan IoT ke dalam sistem Mobil Remot Control (RC), diawali dengan membuat aplikasi *joystick* yang akan dioperasikan pada *smartphone* android, data *output* dari aplikasi *joystick* berupa perintah untuk maju, mundur, belok kanan dan belok kiri. Perintah tersebut dikirim menggunakan jaringan seluler ke firebase, dengan menggunakan salah satu fitur yang dimiliki firebase yaitu *realtime database*, maka setiap perubahan data yang terjadi akan dikirimkan ke NodeMCU yang terdapat pada mobil RC, sehingga mobil RC akan bekerja sesuai dengan perintah yang dikirimkan oleh aplikasi *joystick*, Gambar 1. adalah gambar blok sistem Mobil RC.



Gambar 3. Blok Sistem Mobil Remot Control

Pada Gambar 1. tanda panah berwarna biru, menandakan jalur pengiriman data dari aplikasi *joystick* pada *smartphone* android ke firebase, sedangkan tanda panah berwarna merah adalah jalur data dari firebase ke NodeMCU di mobil RC. Jadi data dari aplikasi joystick tidak langsung dikirim ke mobil RC, akan tetapi disimpan terlebih dahulu di firebase menggunakan fitur realtime database.

Aplikasi *joystick* untuk memberi perintah ke mobil RC dibuat dengan bahasa pemrograman java menggunakan *software* android studio, perintah pada aplikasi *joystick* menggunakan *push button* dengan fungsi *setOnTouchListener*, yaitu fungsi tombol akan memberikan data apabila disentuh atau tidak, perintah yang dikirimkan berupa maju, mundur, belok kanan dan belok kiri, dengan *flowchar* sebagai berikut pada Gambar 2.



Gambar 4. Flowchart Aplikasi Joystick

Penggunaan *push button* dengan fungsi *setOnTouchListener* bertujuan untuk membuat aplikasi *joystick* berjalan seperti tombol pada *remot control* pada umumnya, yaitu apabila tombol ditekan data akan dikirim, sedangkan apabila dilepas atau tidak ditekan maka tidak ada data yang

dikirim sehingga mobil RC diam karena tidak menerima perintah. Script *setOnTouchListener* yang dipergunakan untuk memberikan perintah maju adalah sebagai berikut;

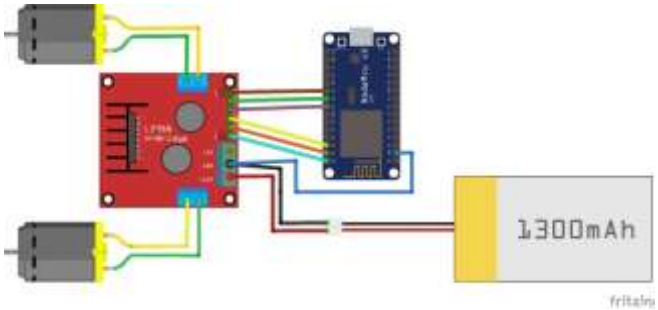
```
forwardImageButton.setOnTouchListener(new View.OnTouchListener() {
   @Override
  public boolean onTouch(View v, MotionEvent event) {
      if (event.getAction() == MotionEvent.ACTION_DOWN) {
        mForward.setValue(1);
        mPost.setValue(ServerValue.TIMESTAMP);
      } else if (event.getAction() == MotionEvent.ACTION_UP) {
         mForward.setValue(0);
      }
      return true;
      }
});
```

Dari script diatas, apabila tombol *forwardImageButton* disentuh atau ditekan aplikasi *joystick* akan mengirimkan data ke *firebase* pada bagian *realtime database* dan mengubah nilai pada *variable* "*Forward*" menjadi 1, sedangkan apabila tombol dilepas atau tidak disentuh maka nilai pada *variable* "*Forward*" menjadi 0. Nilai 0 atau 1 kemudian akan dikirimkan ke NodeMCU pada mobil RC, *variable* "*Forward*" digunakan sebagai perintah, pada "*Forward*" bernilai 1 menandakan HIGH dan 0 menandakan LOW. Jadi apabila *variable* "*Forward*" bernilai HIGH maka mobil RC akan bergerak maju, dan LOW mobil RC akan diam.

Sedangkan *ServerValue.TIMESTAMP* digunakan untuk menyimpan data waktu, yang diperlukan untuk hitungan delay, ketika aplikasi memberikan perintah ke mobil RC maka *Send_Time* akan terisi. *Send_Time* adalah waktu diterimanya perubahan data oleh *realtime database* dan *Received_Time* adalah waktu selesai singkronisasi perubahan data dari NodeMCU ke *realtime database*, Gambar 3. adalah tampilan pada firebase *realtime database* yang dipergunakan;

Gambar 5. Tampilan Firebase Realtime Database

Pada bagian mobil RC, dilakukan penggantian pada bagian *receiver* dan kontrolnya, dengan menggunakan nodeMCU dan modul driver L298N yang dapat secara langsung mengontrol 2 buah motor dc[13], 1 motor sebagi penggerak roda belakang untuk maju dan mundur, dan motor lainnya sebagai pengontrol arah mobil RC. Seperti pada Gambar 4. berikut.



Gambar 6. Wiring NodeMCU dan L298N pada Mobil RC

Kecepatan dan arah putaran motor DC dikendalikan oleh *output* PWM dari NodeMCU, dengan perintah yang diperoleh dari realtime database, perintah untuk singkronisasi data ke *realtime database*, menggunakan *script* sebagai berikut;

```
int x = Firebase.getInt("Forward");
int y = Firebase.getInt("Back");
int z = Firebase.getInt("Direction");

if( x == 1){
       digitalWrite(pinForward, HIGH);
       digitalWrite(pinBack, LOW);
       analogWrite(pinMotor1, 1023);
}else if( y ==1){
       digitalWrite(pinForward, LOW);
       digitalWrite(pinBack, HIGH);
       analogWrite(pinMotor1, 1023);
```

Dari script diatas int *x = Firebase.getInt("Forward");* adalah perintah untuk mengambil data dari realtime database pada *child* "*forward*", dan menyimpan data tersebut pada variable "x", dan nilai dari "x" tersebut akan akan memberikan perintah kepada modul driver L298N untuk menjalankan motor DC sesuai dengan arah dan kecepatan yang diinginkan yaitu mobil RC bergerak maju.

Gambar 7. Mobil Remot Control yang dibuat

Pada Gambar 5. adalah mobil RC yang sudah diganti *receiver* dan kontrolnya menggunakan NodeMCU dan modul driver L298N, untuk sumber tegangan nodeMCU menggunakan sebuah *powerbank*, untuk menghubungkan NodeMCU ke firebase menggunakan jaringan seluler LTE menggunakan sebuah *portable wireless router* yang disimpan pada mobil RC.

4. PENGUJIAN

Pengujian dilakukan pada setiap bagian dari aplikasi *joystick*, komunikasi, pergerakan mobil RC dan kualitas jaringan yang dipergunakan. Pengujian fungsional aplikasi *joystick* pada smartphone android, dengan sebagai berikut pada Tabel 1.

Tabel 1. Pengujian Fungsional Aplikasi *joystick* dan Realtime Database

No	Tombol yang ditekan/ disentuh	Hasil yang diharapkan	Hasil Pengujian	Data yang diharapkan pada *realtime database*	Hasil pengujian pada *realtime database*	Hasil
1	Forward	Mobil bergerak maju	Mobil bergerak maju	Forward 1, Back 0, Direction 45	Forward 1, Back 0, Direction 45	Sesuai
2	Back	Mobil bergerak mundur	Mobil bergerak mundur	Forward 0, Back 1, Direction 45	Forward 0, Back 1, Direction 45	Sesuai
3	Left	Roda depan mengarah ke Kiri	Roda depan mengarah ke Kiri	Forward 0, Back 0, Direction 20	Forward 0, Back 0, Direction 20	Sesuai
4	Right	Roda depan mengarah ke kanan	Roda depan mengarah ke kanan	Forward 0, Back 0, Direction 60	Forward 0, Back 0, Direction 60	Sesuai
5	Forward + Left	Mobil bergerak maju dan belok ke kiri	Mobil bergerak maju dan belok ke kiri	Forward 1, Back 0, Direction 20	Forward 1, Back 0, Direction 20	Sesuai
6	Forward + Right	Mobil bergerak maju dan belok ke kanan	Mobil bergerak maju dan belok ke kanan	Forward 1, Back 0, Direction 60	Forward 1, Back 0, Direction 60	Sesuai
7	Forward + Back	Mobil tidak bergerak	Mobil tidak bergerak	Forward 0, Back 0, Direction 45	Forward 0, Back 0, Direction 45	Sesuai
8	Back + Left	Mobil bergerak mundur ke kiri	Mobil bergerak mundur ke kiri	Forward 0, Back 1, Direction 20	Forward 0, Back 1, Direction 20	Sesuai
9	Right + Right	Mobil bergerak mundur ke kanan	Mobil bergerak mundur ke kanan	Forward 0, Back 1, Direction 60	Forward 0, Back 1, Direction 60	Sesuai
10	Left + Right	Roda depan tetap	Roda depan tetap	Forward 0, Back 0, Direction 45	Forward 0, Back 0, Direction 45	Sesuai

Pengukuran paket data, ketika terjadi proses singkronisasi database, download dan upload, dari aplikasi *joystick* pada *smartphone* ke *realtime database*, menggunakan aplikasi tambahan dari google *playstore* yaitu *Network Monitor Mini*[14], proses pengukuran seperti pada Gambar 8.

Gambar 8. Tampilan aplikasi Joystik beserta nilai *delay*, *download* (D) dan *upload* (U)

Untuk menghitung nilai delay yang pada saat pengiriman data, diperoleh dengan cara mengurangi nilai *Receive_Time* dikurangi S*end_Time*, seperti terlihat pada Gambar 8. nilai delay ditampilkan pada aplikasi *joystick*, nilainya berubah setelah perintah dijalankan oleh NodeMCU. Hasil pengukuran delay, *download* dan *upload* dari aplikasi *joystick* pada *smartphone* ke *realtime database* yang dilakukan menggunakan jaringan LTE, dapat dilihat pada Tabel 2.

Tabel 2. Pengujian Delay, Download dan Upload Aplikasi *joystick* ke Firebase

No	Perintah yg diberikan	Banyaknya percobaan	Rata-rata delay (second)	Rata-rata *download* pada *smartphone* (Kb/s)	Rata-rata *upload* pada *smartphone* (Kb/s)
1	Forward	10 kali	0.98	1.8	2.4
2	Forward + Left	10 kali	1.10	2.6	3.4
3	Forward + Right	10 kali	1.31	2.6	4.4
4	Back	10 kali	1.23	2.2	2.7
5	Back + Left	10 kali	1.22	2.8	3.0
6	Back + Right	10 kali	1.31	1.7	2.9
Rata-rata			1.191	2.28	3.13

Dari hasil pengujian pada Tabel.2, pada aplikasi *joystick* dapat dilihat bahwa data upload lebih besar dari pada data download, perbandingan *data usage* sebelum dan sesudah pengujian dapat dilihat pada Gambar 7.

Gambar 9. Perbandingan *data usage* sebelum dan sesudah dilakukan pengujian

5. KESIMPULAN

Berdasarkan hasil pengujian dapat disimpulkan bahwa, aplikasi *joystick* secara fungsional dapat bekerja sesuai dengan yang diharapkan, data yang tersimpan di *realtime database* sesuai dengan output aplikasi *joystick*. Mobil RC dapat berfungsi sesuai dengan perintah yang dikirim oleh aplikasi *joystick* setelah tersimpan terlebih dahulu di firebase *realtime database*. Ketika menjalankan aplikasi *joystick*, diperoleh nilai *upload* lebih besar dari pada *download*, dengan rata-rata *end to end delay* dari *realtime database* ke NodeMCU adalah 1.191 second, sedangkan rata-rata pengiriman data dari apliakasi *joystick* ke firebase adalah *upload* sebesar 3,13 Kb/s dan *download* sebesar.

DAFTAR PUSTAKA

[1] Widiyanto A, Nuryanto. 2016. *Rancang Bangun Mobil Remote Control Android dengan Arduino*. Citec Journal, Vol. 3, No. 1, November 2015 – Januari 2016, 50-61

[2] Putra M A, Yudho B S, Kurniasari P. 2014.*Pengendali Laju Kecepatan dan Sudut Steering Pada Mobile Robot Dengan Menggunakan Accelerometer Pada Smartphone Android*. Mikrotiga, Vol 1, No. 2 Mei 2014, 19-24

[3] Ardhi S, Sutiksno H. 2016. *Perancangan dan Pembuatan Prototipe Alat Pembersih Lantai dengan Kendali dari Jaringan Bluetooth*. Seminar Internasional dan Konferensi Nasional IDEC 2016, 100-109

[4] Adinandra S, Pangestu W A, Sahroni A. 2014. *Kendali Robot Pemonitor Jarak Jauh Berbasis Smartphone Android Implementasi Sistem Dan Analisis Kualitas Video Streaming*. Seminar Nasional ke – 9: Rekayasa Teknologi Industri dan Informasi, 161-166

[5] Mghawish A J A. 2013. *A Practical Approach For Mobile-Based Remote Control*. European Scientific Journal. June 2013 edition vol.9, No.18, 194-121

[6] Abdiweli Abdillahi Soufi, Abdirasoul Jabar Alzubaidi. 2013. *Remote Control System through Mobile and DTMF*. International Journal of Computational Engineering Research. Vol, 03, 45-52

[7] Towards a definition of the Internet of Things (IoT). Revision 1 – Published 27 May 2015. IEEE Internet Initiative

[8] https://firebase.googleblog.com/2016/05/firebase-expands-to-become-unified-app-platform.html [diakses 1 Mei 2017].

[9] https://firebase.google.com/products/database [diakses 8 Mei 2017]

[10] https://firebase.google.com/docs/reference/js/firebase.database.ServerValue [diakses 10 Mei 2017]

[11] https://nodemcu.readthedocs.io/en/master/en/#nodemcu-documentation, [diakses 8 Mei 2017]
[12] http://nodemcu.com/index_en.html [diakses 8 Mei 2017]
[13] http://www.geeetech.com/wiki/index.php/L298N_Motor_Driver_Board [diakses 8 Mei 2017]
[14] https://play.google.com/store/apps/details?id=info.kfsoft.android.TrafficIndicatorPro&hl=en [diakses 8 Mei 2017]

FORMAT PENULISAN NASKAH ILMIAH JETT

JUDUL ARTIKEL
[JUDUL 14 PTS/BOLD]
[satu baris kosong]

ARTICLE TITLE
[TITLE 14 PTS/BOLD]
[satu baris kosong]

NamaPenulis 1[1], NamaPenulis 2[2], NamaPenulis 3[3] [10 pts]
[satu baris kosong]

[1]Alamat penulis 1
[2]Alamat Penulis 2
[3]Alamat Penulis 3
[1]email1@Penulis 1, [2]email2@Penulis 2, [3]email3@Penulis 3
[10 pts bold underline]

Abstrak [11 pts/Bold]
Abstrak berisi aspek-aspek umum dan kesimpulan utama. Panjang abstrak tidak lebih dari 200 kata dan diketik dalam ukuran huruf 11 pts.
[satu baris kosong]

Kata kunci : Kata kunci sedapat mungkin menjelaskan isi tulisan, dan ditulis dengan huruf kecil, kecuali singkatan. Kata kunci antara 3-6 kata [11 pts/Bold]
[satu baris kosong]

Abstract [11 pts/Bold]
The abstract should state briefly the general aspects of the subject and the main concolusions. The length of abstract should bo no more than 200 word and should be typed be with 11 pts.
[satu baris kosong]

Keywords: keyword should be chosen that they best describe the contents of the paper and should be typed in lower-case, except abbreviation. Keyword should 3- 6 word [11 pts/Bold]
[dua baris kosong]

1. PENDAHULUAN [11 pts/Bold]

Naskah jurnal ditulis di kertas berukuran standar A4 (21 cm x 29.7 cm) dalam jumlah maksimum 8 halaman. Naskah ditulis dalam format font Times New Roman dengan ukuran 11 dan spasi 1.15 . Tambahkan satu spasi untuk setiap antar-bagian (antara judul dan penulis, antara penulis dan abstrak, antara absrak dan kata kunci, antara sub-bab dan isi). Semua margin atas, margin kiri, 30 mm dan margin kanan, margin bawah 25 mm. Margin untuk header dan footer 15 mm. Naskah tidak perlu diberi nomor halaman, header dan footer.

Isi pendahuluan mengandung latar belakang, tujuan, identifikasi masalah dan metoda penelitian, yang dipaparkan secara tersirat (implisit).

Kecuali bab Pendahuluan dan bab Kesimpulan dan Saran, penulisan judul-judul bab sebaiknya eksplisit menyesuaikan isinya. Tidak harus implisit dinyatakan sebagai Dasar Teori, Simulasi Sistem, dan sebagainya.
[satu baris kosong]

2. DASAR TEORI /MATERIAL DAN METODOLOGI/PERANCANGAN [11 pts/Bold]

2.1 Contoh Persamaan Matematika [11 pts/Bold]

Persamaan matematika dinomori dengan angka Arab di dalam tanda kurung buka-tutup pada posisi rata kanan kolom. Persamaan ditulis menjorok ke dalam sejauh satu 7,5 mm. Untuk persamaan yang tidak cukup ditulis dalam lebar 1 kolom, penulisannya dapat melintasi 2 kolom, ditulis di bagian bawah halaman dan diberi nomor urut yang sesuai.

$$C_j = \sqrt{\sum_{i=1}^{n} \left(x_i - w_{ij}\right)^2} \quad (1)$$

2.2 ……

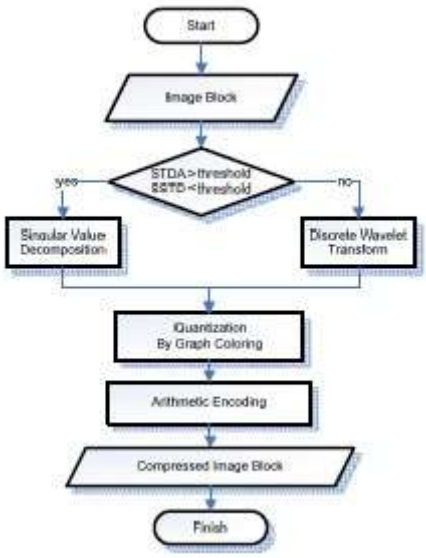
2.3 Keterangan Tabel dan Gambar

Nomor urut tabel ditulis di bagian atas tabel yang dijelaskan, berikut ini contoh penulisan tabel: Tabel 1, Tabel 2(a).

Tabel 1 Range Standar Deviasi dari Gambar [10 pts]

Gambar	Range STDA
Brick	24 – 48
Flowers	8 – 37
Barbara	5 – 32.4

Nomor urut gambar ditulis di bawah gambar yang dijelaskan, contoh: Gambar 1, Gambar 2(a).



Gambar 1. Flowchart Skema yang diusulkan [10 pts]

[satu baris kosong]

3. PEMBAHASAN [11 pts/Bold]

Pembahasan terhadap hasil penelitian dan pengujian, disajikan dalam bentuk uraian teoritik, dapat secara kualitatif maupun kuantitatif. Hasil pengujian sebaiknya ditampilkan dalam berupa grafik atau pun tabel. Untuk grafik dapat mengikuti format untuk diagram dan gambar.
[satu baris kosong]

3.1. Cara Pengajuan dan Pengutipan [11 pts/Bold]

Rujukan dalam pembahasan ditandai nomor pustaka yang dirujuk dalam kurung siku, contoh: [1], [2, 5–7].
[satu baris kosong]

4. KESIMPULAN [11 pts/Bold]

Berisi uraian hasil-hasil penelitian secara jelas serta saran pengembangan untuk penelitian selanjutnya.
Kesimpulan dapat berupa paragraph atau pun *point-point* dengan menggunakan *numbering* atau *bullet.*
[dua baris kosong]

DAFTAR PUSTAKA [11 pts/Bold]
[satu baris kosong]

[6] Ludeman, L. C.. 1987. *Fundamental of Digital Signal Processing*. Singapore : John Wiley & Sons, Inc.
[7] Ochoa H, dan Rao K R. 2003. A Hybrid DWT-SVD Image-Coding System (HDWTSVD) for Color Images. Systemics. *Cybernetics and Informatics*.**1:2** 64-69
[8] Rahardjo, B. 2008. *Pola Akses Internet Yang Bursty*. [Online] Tersedia di http://rahard.wordpress.com/2011/04/04/pola-akses-internet-yang-bursty/ [diakses pada 3 Maret 2011].

Penyusunan rujukan dalam daftar pustaka berurut berdasarkan urutan sitasi yang digunakan (sekuensial) dan diberi nomor angka arab dalam kurung siku. Penulisan unsur-unsur keterangan pustaka mengikuti kaidah dengan urutan: (1) nama pengarang ditulis dengan urutan nama akhir, nama awal dan nama tengah, tanpa gelar akademik. (2) tahun penerbitan. (3) Judul. (4) tempat penerbitan. (5) nama penerbit. Untuk pemisah antar-unsur keterangan pustaka digunakan tanda titik ".". Contoh rujukan [1] adalah untuk buku, sedangkan contoh rujukan [2] adalah untuk jurnal dan rujukan [3] untuk website.

LAMPIRAN
Jika diperlukan, tulisan dapat dilengkapi dengan lampiran.