

PENGUNAAN QR-CODE PADA MOBILE SYSTEM UNTUK PROSES PEMBELIAN PULSA LISTRIK

Hidayatulah Himawan (1), Setyo Agung R. (2), Wilis Kaswidjanti (3)

Jurusan Teknik Informatika, UPN "Veteran" Yogyakarta

Jl. Babarsari No 2, Tambakbayan, Yogyakarta

e-mail : if.iwan@gmail.com

Abstrak

Listrik telah menjadi sumber energi utama dalam setiap kegiatan, baik dalam kegiatan rumah tangga maupun industri. Saat ini PLN menyediakan fasilitas listrik pintar, listrik pintar merupakan layanan baru dari PLN untuk pelanggan dalam mengelola konsumsi listrik melalui meter elektronik prabayar. Dengan Listrik Pintar, pelanggan bisa lebih mudah mengoptimalkan konsumsi listrik, disamping itu pelanggan tidak perlu berurusan dengan pencatatan meter setiap bulan, dan tidak perlu terikat dengan jadwal pembayaran listrik bulanan. Namun saat ini banyak keluhan dari konsumen karena jika menggunakan listrik pintar harus selalu melihat sisa pulsa token karena jika lupa mengecek dan habis di malam hari konsumen mengalami kendala untuk membeli token pulsa dikarenakan outlet di malam hari tutup dan menyulitkan konsumen harus pergi ke outlet pada malam hari. Oleh karena itu, dibutuhkan sebuah aplikasi untuk mempermudah pembelian token listrik. Aplikasi ini untuk melakukan pembelian token listrik menggunakan QR-Code Berbasis sistem operasi android. Metode yang digunakan dalam perancangan dan pembuatan aplikasi ini adalah GRAPPLE (Guidelines for Rappid Application Engineering) dan bahasa pemodelan menggunakan UML (Unified Modelling Language). Aplikasi ini dibuat dengan menggunakan Android Studio, Atom, MYSQL, Zing Barcode Scanner, Bahasa pemrograman java dan PHP.

Kata Kunci : Token, QR-Code, Listrik Pintar, Grapple

1. PENDAHULUAN

Listrik telah menjadi sumber energi utama dalam setiap kegiatan, baik dalam kegiatan rumah tangga maupun industri. Barang-barang elektronik sekarang ini diciptakan dengan ketergantungan menggunakan energi listrik seperti kompor listrik, *rice cooker*, motor listrik dan lain-lain.

Selama ini pelanggan PLN mendapatkan layanan listrik pascabayar yaitu pelanggan menggunakan energi listrik dulu dan membayar belakangan pada bulan berikutnya. Setiap bulan PLN harus mencatat meter, menghitung, dan menerbitkan rekening yang harus dibayar pelanggan, melakukan penagihan kepada pelanggan yang terlambat atau tidak membayar, dan memutuskan aliran listrik jika konsumen terlambat atau tidak membayar rekening listrik setelah waktu tertentu. Mekanisme tersebut tidak dilaksanakan pada sistem listrik pintar atau prabayar. PLN menyediakan fasilitas listrik pintar, listrik pintar merupakan layanan baru dari PLN untuk pelanggan dalam mengelola konsumsi listrik melalui meter elektronik prabayar. Dengan Listrik Pintar, pelanggan bisa lebih mudah mengoptimalkan konsumsi listrik, disamping itu pelanggan tidak perlu berurusan dengan pencatatan meter setiap bulan, dan tidak perlu terikat dengan jadwal pembayaran listrik bulanan. Karena persediaan kWh dapat ditambah berapa saja dan kapan saja sesuai kebutuhan dan keinginan pelanggan.

Latar belakang diluncurkannya program ini sebagai wujud komitmen PLN untuk terus meningkatkan pelayanannya kepada pelanggan dengan berbagai bentuk inovasi. Namun saat ini banyak keluhan dari konsumen karena jika menggunakan listrik pintar harus selalu melihat sisa pulsa token karena jika lupa mengecek dan habis di malam hari konsumen mengalami kendala untuk membeli token pulsa dikarenakan outlet di malam hari tutup dan menyulitkan konsumen harus pergi ke outlet pada malam hari.

Salah satu aplikasi yang bisa digunakan untuk menjawab masalah ini adalah aplikasi pembelian token listrik secara *online* dengan teknologi mobile memanfaatkan QR-Code. Dengan aplikasi akan memudahkan konsumen untuk membeli token pulsa listrik. QR-Code adalah sebuah evolusi dari kode batang dari satu dimensi menjadi dua dimensi. QR-Code merupakan singkatan dari *Quick Response Code* dimana tujuan awal pembuatan QR-Code adalah untuk mendapatkan sejumlah data dengan cepat dan akurat. Permasalahan diatas dapat di pecahkan dengan Aplikasi *Mobile* Pembelian Token Listrik menggunakan QR-Code Berbasis Android.

2. TEORI





QR Code adalah image berupa matriks dua dimensi yang memiliki kemampuan untuk menyimpan data di dalamnya. QR Code merupakan evolusi dari kode batang (*barcode*). *Barcode* merupakan sebuah simbol penandaan objek nyata yang terbuat dari pola batang-batang berwarna hitam dan putih agar mudah untuk dikenali oleh

komputer . QR-Code dikembangkan oleh Denso Wave sebuah divisi dari Denso Corporation yang merupakan perusahaan asal Jepang dan diperkenalkan pada 1994. Berikut beberapa kelebihan QR-Code :

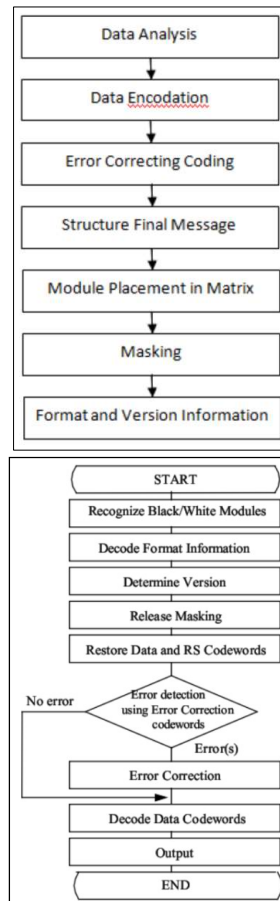
1. Memiliki kapasitas tinggi dalam penyimpanan data pengkodean, QR-Code dapat menyimpan data numerik sebanyak 7089 karakter, alphanumerik 4296 karakter, kode binari 2953 byte.
2. Secara umum ukuran QR-Code bisa hanya sepersepuluh dari ukuran *barcode* konvensional.
3. Tahan terhadap kerusakan karena dapat memperbaiki kesalahan dengan toleransi sampai 30%.
4. Dapat dibaca dengan hasil yang sama dari sudut manapun sejauh 360 derajat.

Perbandingan QR-Code dengan *barcode matriks* lainnya dapat dilihat pada tabel 1.

Tabel 1. Tabel Perbandingan QR-Code dengan *barcode matriks* lain

	QR-Code	PDF417	DataMatrix	Maxi Code
				
Developer (country)	DENSO (Japan)	Symbol Technologies (USA)	RVSI Acuity CiMatrix (USA)	UPS (USA)
Type	Matrix	Stacked Bar Code	Matrix	Matrix
Data capacity	Numeric	7089	2710	3116
	Alphanumeric	4296	1850	2355
	Binary	2953	1018	1556
	Kanji	1817	554	778
Main Features	Large capacity, small printout size, high speed scan	large capacity	small printout size	high speed scan
Main Usages	all categories	OA	FA	Logistics
Standarization	AIM International JIS ISO	AIM International ISO	AIM International ISO	AIM International ISO

Proses pembacaan QR-Code menjadi sebuah teks merupakan kebalikan dari proses *encode*, dapat dijelaskan menggunakan diagram alir pada gambar 1.



Gambar 1. Diagram alir proses pembacaan QR-Code (ISO/IEC 18004, 2000)

Seiring berkembangnya QR-Code, semakin banyak penelitian yang dilakukan mengenai kode simbol ini. Berbagai penelitian terus dilakukan, baik untuk menambah jumlah data yang dapat disimpan dalam QR-Code, menambah resistensi terhadap kerusakan, dan lain-lain. Beberapa penelitian yang telah dilakukan diantaranya adalah (Nugraha, 2011):

1. Pembuatan aplikasi pembacaan QR-Code menggunakan perangkat *mobile* berbasis J2ME(Rahayu, 2006).

Pada penelitian ini dibuat aplikasi pembaca QR-Code menggunakan perangkat *mobile* berbasis sistem operasi J2ME dengan kamera digital yang terintegrasi pada *mobile* tersebut. Pada penelitian ini juga dianalisis tingkat akurasi pembaca QR-Code tersebut sehingga ditemukan parameter yang mempengaruhi tingkat akurasinya.

2. QR-Code untuk tandatangan digital(Adha, 2010).

Pada penelitian ini, peneliti menggunakan QR-Code untuk tanda tangan digital. Data yang di-*encode* adalah *message digest* dari artikel atau tulisan yang akan dibuat tanda tangan digitalnya. Pada penelitian ini juga dibuat sebuah perangkat lunak yang langsung men-*generate* QR-Code dari masukan berupa tulisan atau artikel yang akan dibuat tanda tangan digitalnya.

3. QR-Code untuk autentikasi novel *user*(Kuan, 2010).

Pada penelitian ini, QR-Code digunakan sebagai autentikasi *user* pada sebuah jaringan internet untuk *mobile phone*.

4. QR-Code untuk edukasi (Law, 2007)

Pada penelitian ini dijelaskan manfaat QR-Code untuk edukasi, karena peneliti mengungkapkan bahwa selama ini QR-Code kebanyakan hanya digunakan untuk kepentingan komersil. Contohnya adalah penggunaan QR-Code untuk katalog perpustakaan.

3. METODE PENELITIAN

GRAPPLE merupakan metodologi yang fleksibel dan memberikan panduan yang jelas dalam proses pengembangan sistem. Metode ini terdiri dari lima bagian yaitu (Schmuller, 1999) :

1. Requirement gathering

Pada tahap pertama yang dilakukan oleh pengembang perangkat lunak adalah mengambil informasi lengkap dari pengguna tentang sistem yang akan dibangun dengan wawancara dan kuisioner. Wawancara dilakukan langsung dengan pengguna yang menginginkan adanya sistem ini dan dengan pengguna yang berhubungan langsung dengan sistem. Tahap ini menyarankan untuk mewawancarai pengguna yang memiliki kemampuan teknis. Analisis masalah, fungsi dan kebutuhan sistem termasuk dalam tahap *requirement gathering*.

2. Analysis

Di tahap *analysis* yang dilakukan adalah menggali lebih dalam hasil yang diperoleh dalam tahap sebelumnya. Tahap ini mengkaji permasalahan pengguna dan menganalisis solusinya. Pada tahap ini yang termasuk didalamnya antara lain adalah pengembangan data dan informasi dari *requirement gathering* serta pembuatan diagram.

3. Design

Tahap *design* dilakukan untuk merancang solusi yang dihasilkan pada tahap *analysis* dan *design* dapat berjalan dua arah saling menyesuaikan sampai diperoleh rancangan yang tepat. Pada tahap ini yang termasuk didalamnya adalah diagram yang telah dianalisis serta dibuat rancangannya.

4. Development

Tahap ini ditangani oleh pengembang program untuk membangun kode program dan *user interface*. Pengujian program dan dokumentasi sistem dilakukan pada tahap ini.

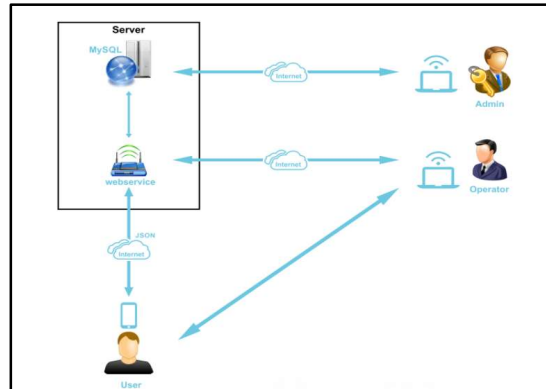
5. Deployment

Tahap *deployment* adalah tahap pendistribusian produk yang dihasilkan kepada pengguna. Tahap ini mencakup instalasi dan perencanaan *backup* data bila diminta oleh pengguna sesuai dengan perjanjian sebelumnya.

4. HASIL DAN PEMBAHASAN

Arsitektur sistem

user yang sudah meng-*install* aplikasi, bisa melakukan pembelian secara *online* hanya dengan memindai QR Code. Selain itu, *user* juga dapat melihat riwayat transaksi, profil dan mengubah data profil. Arsitektur dari aplikasi *mobile* pembelian token listrik berbasis sistem operasi *android* dapat dilihat pada gambar 2.



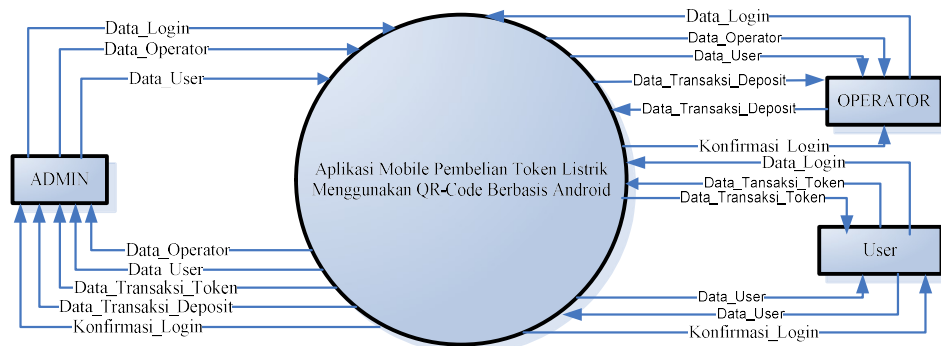
Gambar 2. Arsitektur Sistem Aplikasi

Aplikasi ini membutuhkan koneksi internet untuk mengambil data yang ada di server. *Admin* mempunyai hak akses untuk mengolah data *user*, *operator*, data transaksi token dan data transaksi deposit, tetapi untuk lalu lintas transaksi yang sedang terjadi *admin* tidak dapat merubah ataupun menghapusnya. Sedangkan *operator* pada sisi *server* mempunyai hak akses untuk mengolah data isi deposit

Data Flow Diagram (DFD)

Data Flow Diagram Level 0

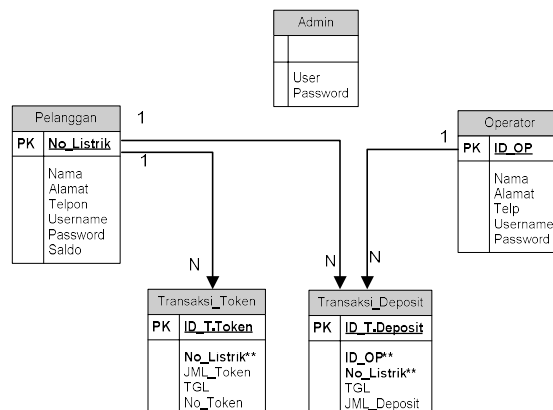
Pada DFD level 0 membahas mengenai entitas *admin* sebagai pengelola sistem, entitas *operator* sebagai pengelola dalam pengisian deposit dan entitas *user* sebagai pengguna aplikasi. Pada *admin* terdapat alir data yaitu data *login*, data transaksi deposit, data *operator*, data *user* dan data transaksi token. Pada *operator* terdapat alir data yaitu data *login*, data *user*, data *Operator* dan data transaksi deposit. Dan pada *user* terdapat alir data *login*, data transaksi token dan data *user*. Data Flow Diagram Level 0 dapat dilihat pada gambar 3.



Gambar 3. Data Flow Diagram Level 0

3. Perancangan RAT

Perancangan relasi antar tabel aplikasi *mobile* pembelian token listrik berbasis android dapat dilihat pada gambar 4.

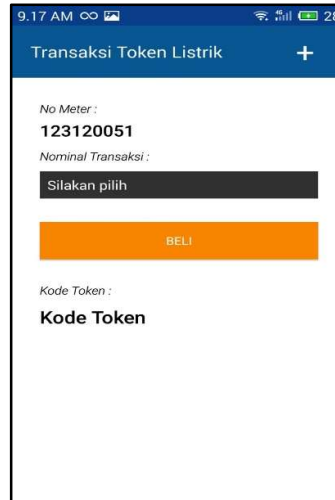


Gambar 4. Relasi Antar Tabel

Dalam bab implementasi ini akan membahas salah satu tahapan dalam metodologi *Guidelines for Rapid APPLication Engineering* (GRAPPLE) yaitu tahapan pengembangan (*development*). Segmen ini bertujuan untuk menerjemahkan keperluan perangkat lunak dalam bahasa yang dimengerti oleh *smartphone*. Aksi konstruksi kode program, pengujian kode program dan konstruksi antar muka termasuk koneksinya serta pengujian merupakan tiga aksi yang terdapat pada segmen ini. Pembahasan lebih rinci dari penelitian ini akan dilakukan kelas per kelas.

Kelas beli.java

Kelas beli.java adalah kelas untuk menampilkan halaman pembelian secara manual. Pada halaman beli terdapat *form* yang berisikan nominal yang akan dipilih oleh pelanggan itu sendiri. Tampilan kelas beli.java dapat dilihat pada gambar 4.6, potongan modul program kelas beli.java dapat dilihat pada modul program 5.



Gambar 5. Tampilan Halaman beli

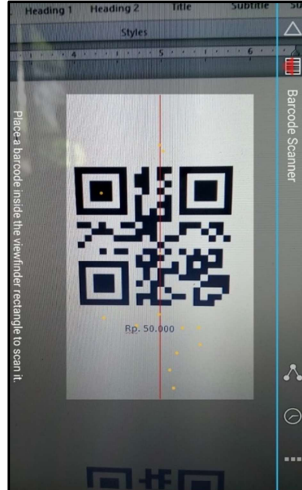
Modul program kelas beli dapat dilihat pada Modul Program 1, yaitu :

```
public class CekLogin extends AsyncTask<String, String, String> {
    ArrayList<HashMap<String, String>> contactList = new ArrayList<HashMap<String, String>>();
    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        nomor = no.getText().toString();
        String trim1 = saldo.getSelectedItem().toString().replace("Rp ", "");
        jml = trim1.replace(".", "");
    }
    @Override
    protected String doInBackground(String... arg0) {
        String url = "http://192.168.137.1/JSON-PLN/proses-beli.php";
        JSON jParser = new JSON();
        List<NameValuePair> params = new ArrayList<NameValuePair>();
        params.add(new BasicNameValuePair("id", nomor));
        params.add(new BasicNameValuePair("jumlah", jml));
        JSONObject json = jParser.makeHttpRequest(url, "POST", params);
        Log.d(" Cek JSON1: ", json.toString());
        try {
            sukses = json.getInt("sukses");
            //int sukses1 = json1.getInt(TAG_MEMBER);
            if (sukses == 1) {
                cek = json.getJSONArray("data");
                JSONObject c = cek.getJSONObject(0);
                token = c.getString("token");
            } else {
                Log.d("Gagal Transaksi ", json.toString());
                pesan = json.getString("pesan");
                return json.getString("pesan");
            }
        } catch (JSONException e) {
            e.printStackTrace();
        }
        return null;
    }
}
```

Modul Program 1. Kelas beli.java

Kelas QRscan.java

Kelas QRscan.java adalah kelas untuk menampilkan halaman Beli. Pada halaman Beli terdapat aplikasi pihak ketiga untuk melakukan proses pindai QR-Code dan melakukan proses transaksi pembelian token. Tampilan kelas QRscan.java dapat dilihat pada gambar 6.



Gambar 6. Tampilan Halaman Beli

Modul program kelas QRscan dapat dilihat oada Modul Program 2.

```
Protected String doInBackground(String... args) {
    List<NameValuePair> params1 = new ArrayList<NameValuePair>();
    params1.add(new BasicNameValuePair("id", user));
    params1.add(new BasicNameValuePair("jumlah", upc));
    JSONObject json = jsonParser.makeHttpRequest(url_proses,"POST", params1);
    Log.d("Respon tambah transaksi", json.toString());
    try {
        int sukses = json.getInt("sukses");
        if (sukses == 1) {
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    final AlertDialog.Builder builder=new AlertDialog.Builder(QRscan.this);
                    builder.setTitle("Terima Kasih :");
                    builder.setMessage("Pembelian token telah berhasil");
                    builder.setIcon(android.R.drawable.ic_dialog_alert);
                    builder.setNegativeButton("OK", new OnClickListener() {
                        @Override
                        public void onClick(DialogInterface dialog, int which) {
                            // TODO Auto-generated method stub
                            QRscan.this.finish();
                            integrator.initiateScan(QRscan.this);
                            Intent i = new Intent(getApplicationContext(), Riwayat.class);
                            startActivity(i);
                        }
                    });
                    builder.show();
                }
            });
        }
        return json.getString("pesan");
    } else {
        if (sukses == 2) {
            runOnUiThread(new Runnable() {
                @Override
                public void run() {
                    final AlertDialog.Builder builder=new AlertDialog.Builder(QRscan.this);
                    builder.setTitle("Berita :");
                    builder.setMessage("Saldo anda tidak mencukupi untuk melakukan pembelian ini. Segera isi ulang di outlet terdekat");
                    builder.setIcon(android.R.drawable.ic_dialog_alert);
                    builder.setNegativeButton("Keluar", new OnClickListener() {
                        @Override
```

```

        public void onClick(DialogInterface dialog, int which) {
            // TODO Auto-generated method stub
            QRscan.this.finish();
            integrator.initiateScan(QRscan.this);
            Intent i = new Intent(getApplicationContext(), MainActivity.class);
            startActivity(i);
        }
    });
    builder.show();
}
});
return json.getString("pesan");
} else {
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            final AlertDialog.Builder builder=new AlertDialog.Builder(QRscan.this);
            builder.setTitle("Berita :");
            builder.setMessage("Terjadi kesalahan pada transaksi anda. Coba ulangi beberapa saat lagi");
            builder.setIcon(android.R.drawable.ic_dialog_alert);
            // Tombol untuk coba lagi/keluar
            builder.setNegativeButton("Ok", new OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    // TODO Auto-generated method stub
                    QRscan.this.finish();
                    integrator.initiateScan(QRscan.this);
                    Intent i = new Intent(getApplicationContext(), MainActivity.class);
                    startActivity(i);
                }
            });builder.show();
        }
    });
    return json.getString("pesan");
}
} catch (JSONException e) {
    e.printStackTrace();
}
return null;
}
}

```

Modul Program 2. Kelas QRscan.java

4.1.1 Halaman data transaksi token

Halaman data transaksi token merupakan halaman yang digunakan oleh *admin* untuk melihat data riwayat transaksi token. Halaman data transaksi token dapat dilihat pada gambar 7.

AdminPLN Pelanggan Operator Info Transaksi ▾				
ADMIN				
Data Transaksi Penjualan Token				
Show 10 ▾ entries		Search: <input type="text"/>		
No	No.Listrik	Tanggal	Nominal Token	
1	123456789	2016-05-02	25000	
2	123456789	2016-05-02	50000	
3	123456789	2016-05-02	50000	
4	123456789	2016-05-02	50000	
5	123120051	2016-05-10	50000	

Gambar 7. Tampilan halaman data transaksi token

5. KESIMPULAN

Berdasarkan hasil pengembangan yang dilakukan maka diperoleh kesimpulan pada penelitian ini, antara lain :

1. Telah berhasil dibangun sebuah aplikasi yang dapat mempermudah pelanggan dalam melakukan transaksi pembelian token listrik melalui sebuah aplikasi *mobile* pembelian token listrik menggunakan QR-Code yang berjalan pada platform android.
2. Aplikasi ini diharapkan mempermudah pelanggan, karena tidak perlu keluar rumah saat kehabisan token listrik saat malam hari.

DAFTAR PUSTAKA

- Adha, Ripandy. 2010. *Message Digest dalam bentuk QR Code Sebagai Tanda Tangan Digital*. Bandung: Program Studi Teknik Informatika STEI ITB.
- Kuan, Cieh Liao & Hsun Lee Wei. 2010. *A Novel User Authentication Scheme Based on QR-Code*. Taichung, Taiwan: Journal of Networks, Vol.5, No.8. pp937-941.
- Law, Ching-Ying & Simon So. 2007. *QR Code in Education*. Hongkong: Journal of Educational Technology Development and Exchange, 3(1), 85-100.
- Moren, Yani, 2013, *Media Pembelajaran Matematika Untuk Balita Dengan Metode Glenn Doman Berbasis Android*, Teknik Informatika, Fakultas Teknik, Universitas Widyatama, Bandung.
- Nugraha, M Pasca, 2011, *Pengembangan Aplikasi QR-Generator dan QR-Code Reader Dari Data Berbentuk Image*, Institut Teknologi Bandung, Bandung.
- Rahayu, Yeni Dwi, dkk. 2006. *Pembuatan Aplikasi pembacaan Quick Responsse Code Menggunakan Perangkat Mobile Berbasis J2ME Untuk Identifikasi Suatu Barang*. Surabaya: Politeknik Elektronika Negeri Surabaya Institut Teknologi Sepuluh Nopember
- Schmuller, Joseph, 1999, *Teach Yourself UML in 24 Hours*, Sams Publishing, IndianaPolis.
- Saputra, Agus, 2011, *Trik Dan Solusi Jitu Pemrograman PHP*.
- Saputra, Alfian dan Yuliani, Febri, *Analisis Kualitas Layanan Program Listrik Pintar (Prabayar di PT. PLN (PERSERO) Pekanbaru)*, Jom FISIP, Vol 1 no. 2, Oktober 2014.