

# ASPEK KOMPUTASI PLATFORM SOFTWARE-DEFINED RADIO (SDR)

Eko Marpanaji, Bambang Riyanto T., Armein Z.R. Langi, Adit Kurniawan, Andri Mahendra

Teknik Elektro STEI ITB

ekoaji332@students.itb.ac.id, briyanto@lskk.ee.itb.ac.id, langi@lss.ee.itb.ac.id  
adit@ltrgm.ee.itb.ac.id

## Abstrak

Sejak diluncurkan istilah *Software-Defined Radio (SDR)* pada tahun 1991, paradigma desain radio telah berubah kedalam sebuah platform radio yang dapat diprogram ulang dan dapat dikonfigurasi ulang sehingga menghasilkan sistem komunikasi nirkabel yang fleksibel. Tingkat fleksibilitas ditentukan oleh desain arsitektur SDR dengan memperhatikan isu-isu penting dalam desain sebuah platform SDR, yaitu tantangan komponen ADC/DAC dan kompleksitas komputasi.

Makalah ini akan menguraikan tentang arsitektur SDR, isu-isu penting dalam desain arsitektur SDR, perspektif matematis kebutuhan komputasi dalam mendesain platform SDR, serta kajian awal dari riset yang sedang dilakukan dalam mengembangkan platform SDR. Aplikasi SDR sangat dinantikan untuk pengembangan komunikasi seluler dan bergerak untuk generasi mendatang, *Next Generation Networks (NGN)*, serta *Rural Next Generation Networks (R-NGN)*. Dengan menguasai teknologi SDR diharapkan dapat menjadi sarana dalam pengembangan teknologi komunikasi nirkabel di tanah air di masa datang.

**Kata Kunci:** *software-defined radio, reconfigurable platform, arsitektur, sinyal IF, modulasi, demodulasi, perangkat keras, perangkat lunak, kapasitas komputasi, pengolahan sinyal digital.*

## 1. PENDAHULUAN

Sejalan dengan perkembangan teknologi komunikasi, standar komunikasi juga berubah begitu cepat. Hal ini menjadi masalah yang serius, karena sebagian besar standar komunikasi diimplementasikan dalam bentuk perangkat keras. Perubahan standar ini akan selalu mendorong investasi perangkat keras yang baru dan akan mengakibatkan mahalnya biaya langganan, sehingga masyarakat tingkat ekonomi menengah kebawah tidak mampu menikmati layanan yang diberikan.

*Software-Defined Radio (SDR)* merupakan teknologi komunikasi berbasis nirkabel yang fungsinya ditentukan oleh perangkat lunak. SDR memiliki sifat fleksibel dan dapat dikonfigurasi ulang sehingga perubahan standar dapat dilakukan pada perangkat lunak tanpa harus mengganti perangkat kerasnya. Teknologi SDR ini sangat diharapkan untuk pengembangan komunikasi nirkabel di masa datang.

Kehadiran teknologi SDR sangat cocok dengan perkembangan teknologi generasi mendatang yang tetap mengutamakan komunikasi nirkabel dan bergerak. Kedepan, sistem komunikasi nirkabel berbasis SDR sangat diperlukan dalam pengembangan sistem komunikasi seluler generasi ke-3 (3G) seperti WCDMA dan UMTS, dan *Generation Networks (NGN)* termasuk *Rural Next Generation Networks (R-NGN)*.

Makalah ini akan memaparkan arsitektur SDR dan isu-isu desain platform SDR ditinjau dari aspek komputasinya, serta kajian awal dari riset yang sedang

dilakukan tentang platform SDR yang akan menjawab pertanyaan: Bagaimana merancang platform SDR untuk pengembangan R-NGN. Sistematika makalah ini adalah: Bagian 2 membahas tentang arsitektur SDR beserta sifat-sifat yang dimiliki SDR, Bagian 3 menjelaskan isu-isu penting dalam desain arsitektur SDR beserta aturan-aturannya, sedangkan Bagian 4 membahas gambaran matematis desain platform SDR, dan terakhir Bagian 5 adalah kesimpulan.

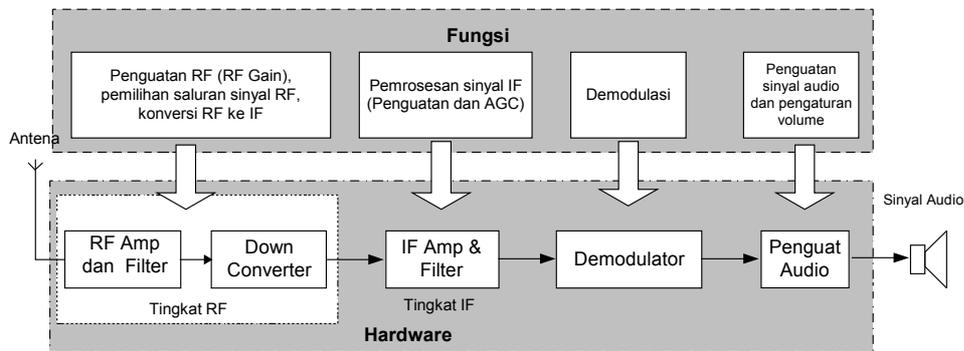
## 2. ARSITEKTUR SDR

*Software-Defined Radio (SDR)*, ada yang menyebut juga *Software Radio (SWR)*, diperkenalkan pertama kali pada tahun 1991 oleh Joseph Mitola[1][2]. Istilah SDR ini digunakan untuk menunjuk sebuah kelas radio yang dapat dikonfigurasi ulang atau diprogram ulang[3], sehingga menghasilkan sebuah jenis perangkat komunikasi nirkabel dengan mode dan band frekuensi yang ditentukan oleh fungsi perangkat lunak. Secara ideal, SDR menawarkan fleksibilitas (*flexibility*), dapat dikonfigurasi ulang (*reconfigurability*), memungkinkan untuk perluasan (*scalability*), dan bahkan memiliki mode sebanyak mungkin (*multi mode*).

Arsitektur SDR dikembangkan berdasarkan fungsi-fungsi radio konvensional. Sebagai ilustrasi dapat dicermati blok diagram dan fungsi sebuah radio penerima konvensional seperti yang ditunjukkan Gambar-1. Semua fungsi pemrosesan sinyal pada radio konvensional dilakukan sepenuhnya oleh perangkat

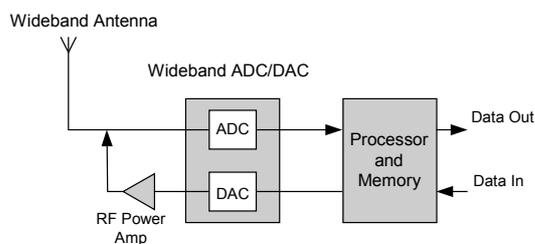
keras. Kunci utama dalam membangun SDR adalah penempatan komponen ADC dan DAC sebagai pemisah

antara sinyal analog dan digital yang akan diproses dalam sebuah SDR.

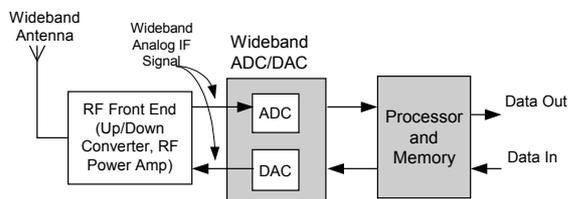


Gambar-1. Radio penerima konvensional

Arsitektur SDR yang ideal akan menempatkan ADC/DAC sedekat mungkin dengan antena seperti yang ditunjukkan pada Gambar-2 Namun demikian cara ini akan membutuhkan wideband ADC/DAC dengan kecepatan sampling yang sangat tinggi untuk melakukan konversi analog ke digital atau sebaliknya terhadap sinyal *Radio Frequency* (RF). Fungsi-fungsi radio dilakukan oleh perangkat lunak yang dijalankan pada prosesor, sehingga lebih fleksibel karena perubahan standar komunikasi dapat diantisipasi pada perangkat lunaknya saja [1][3]. Namun demikian, keterbatasan teknologi dan mahalnya wideband ADC/DAC mendorong untuk sedikit mengubah arsitektur SDR dalam menempatkan ADC/DAC sehingga menjadi lebih realistis seperti yang ditunjukkan pada Gambar-3.



Gambar-2. Arsitektur SDR Ideal



Gambar-3. Arsitektur SDR Realistis

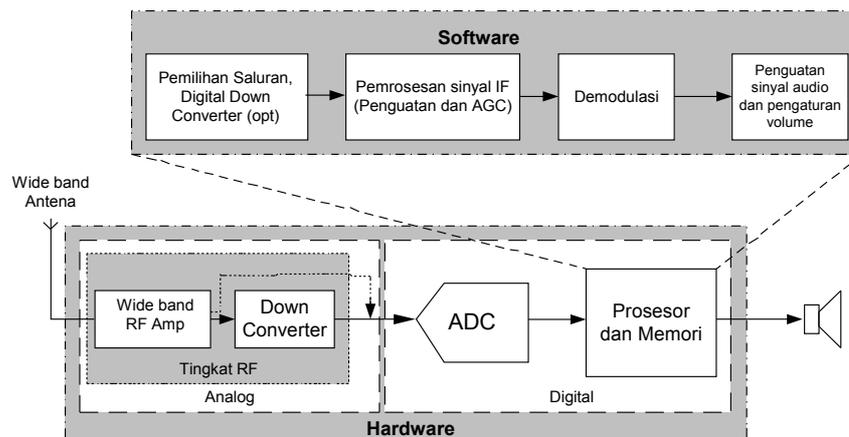
Arsitektur SDR yang lebih realistis menempatkan wideband ADC/DAC setelah Down Converter/Up Converter, sehingga konversi analog ke digital atau sebaliknya dilakukan pada sinyal *Intermediate Frequency* (IF) yang memiliki frekuensi lebih rendah dibanding sinyal RF. Arsitektur tersebut saat ini banyak dikembangkan dan dalam proses penelitian untuk implementasinya. Berdasarkan arsitektur pada Gambar-3, maka arsitektur radio

penerima konvensional dari Gambar-1, dapat dikembangkan menjadi radio penerima SDR seperti yang ditunjukkan pada Gambar-4 dengan melakukan pemisahan (partisi) antara perangkat keras dan perangkat lunaknya.

### 3. ISU DESAIN ARSITEKTUR SDR

Semenjak diluncurkannya istilah SDR pada tahun 1991 oleh Mitola, paradigma perancangan sistem radio juga mulai berubah. Dalam pendekatan konvensional, sistem radio multi standar diimplementasikan dalam bentuk kumpulan *transceiver* atau bisa disebut dengan barisan radio dengan standar berbeda-beda yang disatukan menjadi sebuah sistem radio multi standar. Pendekatan konvensional ini terbukti tidak fleksibel, karena masih melibatkan beberapa perangkat keras untuk tiap-tiap standar komunikasi dan tidak mampu mengantisipasi perubahan standar yang akan terjadi. Selain itu, pendekatan konvensional ini menjadi tidak layak dan mahal karena tingkat kompleksitas perangkat keras yang luar biasa besar [4]. Dengan adanya teknologi SDR, desain radio beralih pada pendekatan yang lebih efisien yaitu sistem radio yang dapat dikonfigurasi ulang dengan menyediakan sebuah platform yang mampu menerima perkembangan (perubahan) standar komunikasi. Fleksibilitas dapat diperoleh dari implementasi perangkat lunak yang dijalankan pada perangkat keras SDR.

Pada prinsipnya, desain arsitektur SDR meliputi desain perangkat keras dan perangkat lunak. Mitola (2000) menyatakan ada 7 (tujuh) tahapan yang harus dilakukan dalam membangun SDR, yaitu: (1) perancangan antena jalur lebar (*wide band antenna*); (2) rancangan frekuensi RF disesuaikan dengan kemampuan ADC-DAC sehingga diperoleh unjuk kerja yang baik; (3) pemilihan ADC/DAC untuk sinyal jalur lebar; (4) membangun interkoneksi jalur lebar untuk menghubungkan ADC/DAC dengan prosesor; (5) membangun prosesor untuk pengolahan sinyal digital, termasuk proses modulasi dan demodulasi; (6) merancang perangkat lunak, diutamakan berbasis obyek termasuk *message passing* dan *encapsulation*; (7) merancang aplikasi.



Gambar-4 Radio penerima SDR

Beberapa pendekatan yang diusulkan dalam merancang arsitektur SDR yaitu: (1) Analisis Arsitektur, dimana arsitektur SDR dapat dianalisis dari Fungsi (*Function*), Komponen (*Component*), dan Aturan Perancangan (*Design Rule*) [3]; (2) Berorientasi Obyek, dengan menggunakan Unified Modeling Language (UML) untuk memodelkan SDR berdasarkan *Use case view*, *Logical View*, *Component view*, dan *Deployment View*[3][6][7]; dan (3) Arsitektur terlapis (*Layered Architecture*), dimana fungsionalitas radio dibagi menjadi tiga lapisan yaitu: *Soft radio interface (SRI) layer*, *Configuration layer*, serta *Processing layer* [8].

SDR memiliki keuntungan karena sifat fleksibilitas (*flexibility*), lengkap dan dapat dikonfigurasi ulang secara mudah (*complete and easy reconfigurability*), serta kemampuan dapat diskala (*scalability*)[1]. Sedangkan Cristensen (2004) menyatakan bahwa kunci utama sebuah platform SDR embedded adalah sifat fleksibilitas (*flexibility*), dapat diperluas (*expandability*), dapat diskala (*scalability*), dapat dikonfigurasi ulang (*reconfigurability*), dan dapat diprogram ulang (*reprogrammability*). Sifat-sifat ini harus muncul dalam produk akhir dalam desain sebuah SDR[5].

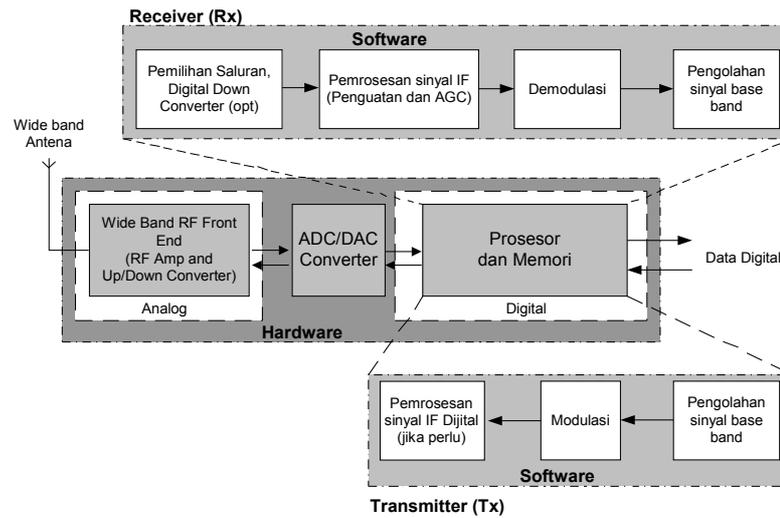
Berdasarkan uraian di atas, maka isu-isu penting dalam desain SDR adalah kemampuan ADC/DAC dan kapasitas atau kecepatan komputasi prosesor untuk pengolahan sinyal digital. Kedua isu ini akan menentukan unjuk kerja SDR. Unjuk kerja SDR dapat diukur berdasarkan konsumsi daya, biaya relatif, kemampuan komputasi, atau menggunakan profile *Quality of Service (QoS)*, yaitu *bit rate*, *bit error rate (BER)*, *packet loss rate (PLR)*, *delay spread*, dan *Grade of Service (GoS)*. Kapasitas komputasi prosesor biasanya dinyatakan dalam *millions operations per second (MOPS)* dan dihitung berdasarkan kebutuhan komputasi untuk pengolahan sinyal digital. Sebagai gambaran, ukuran unjuk kerja menggunakan QoS misalnya:  $BER < 10^{-3}$ ,  $PLR < 10^{-2}$ ,  $delay\ spread < 100\ ms$  dan  $GoS > 95\%$  [3]. Angka-angka yang digunakan sebagai patokan tersebut disesuaikan dengan rancangan aplikasi sistem SDR yang akan dikembangkan.

#### 4. PERSPEKTIF MATEMATIS PLATFORM SDR

Untuk menghasilkan sebuah platform SDR yang dapat dikonfigurasi ulang (*reconfigurable platform SDR*) dan memiliki sifat-sifat seperti tuntutan isu-isu desain SDR sangatlah tidak mudah. Agar sistem benar-benar menjadi fleksibel, maka pendekatan SDR adalah mengimplementasikan perangkat keras yang dapat diprogram ulang sebagai platformnya dan hal ini mendorong kita untuk menempatkan komponen ADC/DAC dalam membangun sebuah SDR. Tantangan utama adalah proses dijitasi, kapasitas dan kecepatan komputasi prosesor, serta sifat skalabilitas perangkat lunak.

Skalabilitas perangkat lunak dapat diperoleh dengan mencermati fungsi-fungsi SDR serta tingkat fleksibilitas yang diinginkan untuk menentukan kompleksitas komputasi, serta kemungkinan untuk pengembangan berikutnya. Semakin tinggi kompleksitas algoritma perangkat lunak yang akan dijalankan tentunya akan semakin besar kapasitas komputasi prosesor yang diperlukan. Lebih jauh lagi, untuk mendukung sifat dapat dikonfigurasi ulang dan dapat diprogram ulang memerlukan usaha yang keras dalam merancang sebuah platform SDR. Tuntutan kecepatan komputasi untuk mengejar pemrosesan sinyal digital dengan laju bit yang tinggi juga merupakan tantangan dalam merancang platform SDR.

Sebagai gambaran dalam desain platform SDR, dapat dilihat fungsi sederhana sebuah SDR seperti ditunjukkan Gambar-5. Sebuah platform SDR akan menjalankan fungsi pemancar dan penerima mulai dari ujung antena sampai dengan pemrosesan data *baseband* (informasi). Sisi pemancar (Tx) akan menjalankan fungsi : (1) pemrosesan sinyal digital baseband; (2) modulasi; (3) pengolahan sinyal digital IF; dan (4) pengiriman sinyal RF ke udara. Sedangkan sisi penerima (Rx) akan menjalankan fungsi: (1) pengolahan sinyal RF, (2) proses pemilihan saluran (*channelization*); (3) pengolahan sinyal digital IF; (4) demodulasi; dan (5) pemrosesan sinyal digital baseband. Nampak secara sekilas bahwa tingkat kompleksitas komputasi sistem penerima lebih tinggi dibanding sistem pemancar.



Gambar-5. Arsitektur perangkat keras dan perangkat lunak sebuah SDR

Letak ADC/DAC akan menentukan titik dimana fungsi-fungsi dapat dijalankan oleh perangkat lunak atau sering disebut dengan titik akses dijital (*digital access point*). Gambar di atas menunjukkan bahwa proses yang ditangani oleh perangkat lunak dimulai dari pengolahan sinyal IF. Tantangan utama dalam memilih letak ADC adalah masalah keterbatasan frekuensi sampling dan faktor ekonomis. Jika frekuensi sampling Nyquist sulit dilakukan karena dijitalisasi sinyal IF masih dipandang terlalu berat, maka alternatifnya adalah menerapkan *under sampling* terhadap sinyal IF sehingga menurunkan kompleksitas komputasi dari prosesor.

Formula untuk menentukan frekuensi under sampling adalah [9] :

$$\frac{2f_c + B}{m + 1} \leq f_s \leq \frac{2f_c - B}{m}; m \in \mathbb{N}$$

dimana  $f_c$  frekuensi center,  $B$  bandwidth,  $f_s$  frekuensi sampling. Misal sistem SDR dari Gambar-5, menggunakan sinyal IF 20 MHz dengan bandwidth 5 MHz dapat disampling dengan frekuensi 22,5 MHz untuk  $m=1$ , atau 17,5 MHz untuk  $m=2$ , atau 11,6 MHz untuk  $m=3$ .

Perhatian berikutnya adalah pada kompleksitas komputasi yang akan menentukan jenis dan ukuran prosesor yang diperlukan. Prosesor akan menjalankan perangkat lunak pengolahan sinyal dijital baik untuk penerima ataupun pemancar. Selain mencermati fungsi-fungsi yang dikerjakan oleh perangkat lunak dalam menentukan kompleksitas komputasi, juga dicermati faktor skalabilitas untukantisipasi perkembangan selanjutnya. Semua ini akan menentukan ukuran prosesor yang biasanya dinyatakan dengan *Million Operations per Second* (MOPS) atau *Million Instructions per Second* (MIPS).

Kebutuhan komputasi paling besar adalah pada sisi penerima khususnya pada bagian filter dalam pemilihan saluran, yaitu 100 s.d. 200 operasi/sampel [4][3][10]. Sehingga dengan sampling 22,5 MHz akan

menghasilkan 22,5 MSPS, sehingga membutuhkan 2.250 s.d. 4.500 MIPS. Kebutuhan komputasi lainnya adalah proses modulasi dan demodulasi, sebagai gambaran untuk modulasi dan demodulasi FM membutuhkan kurang lebih 100 s.d. 150 MIPS[11]. Berdasarkan angka-angka tersebut, kebutuhan komputasi untuk sistem penerima dan pemancar membutuhkan (diambil angka yang terbesar) kurang lebih 4.650 MIPS, yang sebagian besar diperlukan dalam proses radio penerima. Untuk mengantisipasi kebutuhan komputasi kendali dan pengembangan selanjutnya biasanya orang menempatkan 50% dari kapasitas komputasi maksimum dari sebuah prosesor. Dengan demikian, 4.650 MIPS membutuhkan prosesor yang memiliki kapasitas komputasi kurang lebih 10.000 MIPS.

Untuk meningkatkan unjuk kerja, para pakar mengimplementasikan komputasi pemilihan saluran pada sistem penerima ke dalam chip FPGA, sedangkan proses modulasi, demodulasi, dan pengolahan sinyal dijital baseband dilakukan pada prosesor yang lain. Kebutuhan komputasi ini akan lebih besar lagi jika kita memperhatikan perangkat lunak aplikasi yang sesungguhnya misalnya integrasi dengan standar komunikasi VoIP dan protokol TCP/IP untuk membangun sebuah VoIP Gateway berbasis SDR.

Berdasarkan gambaran perhitungan kompleksitas komputasi di atas, maka pengembangan aplikasi SDR untuk komunikasi nirkabel bergerak atau seluler generasi ke-3 (3G) dan NGN akan membutuhkan platform dengan kapasitas komputasi yang sangat besar terutama pada bagian penerima. Mitola (2000) memberikan gambaran kebutuhan komputasi basestation. Formula yang digunakan untuk menentukan kebutuhan komputasi adalah sebagai berikut:

$$D = D_{IF} + N * (D_{bb} + D_{bs} + D_s) + D_o;$$

$$D_{if} = W_a * (G_1 + G_2) * 2.5; \text{ dan } D_{bb} = W_c * (G_m + G_d);$$

$$\text{serta } D_{bs} = R_b * G_3 * (1/r)$$

dimana:

- $D$  = kebutuhan komputasi (MOPS)
- $D_{if}$  = kebutuhan pemrosesan IF
- $D_{bb}$  = kebutuhan prosesan sinyal baseband
- $D_{bs}$  = kebutuhan pemrosesan bitstream
- $D_s$  = kebutuhan pemrosesan sumber
- $D_o$  = pemrosesan proses *overhead management*
- $W_a$  = bandwidth akses layanan
- $G_1$  = kompleksitas filter isolasi layanan
- $G_2$  = kompleksitas isolasi saluran pelanggan
- $G_3$  = kompleksitas proses bitstream tiap saluran
- $N$  = jumlah pelanggan
- $W_c$  = bandwidth tiap saluran
- $G_m$  = kompleksitas modulasi dan filter
- $G_d$  = kompleksitas proses demodulasi
- $R_b$  = laju bit dari bitstream
- $r$  = laju kode

Segmen	Parameter	Illustrative Value	Demand
IF	$W_a$	25 MHz	$W_a * G_1 * 2.5 = 6.25 \text{ GOPS}^*$
	$G_1$	100 OPS/Hz	$D_{if} = W_a * (G_1 + G_2) * 2.5 = 12.5 \text{ GOPS}^*$
	$G_2$	100 OPS/Hz	
	$N$	30/cell site	
	$W_c$	30 kHz	
	$G_m$	20 OPS/Hz	$W_c * G_m = 0.6 \text{ MOPS}$
Baseband	$G_d$	50 OPS/Hz	$D_{bb} = W_c * (G_m + G_d) = 2.1 \text{ MOPS}$
	$R$	1 b/b	
	$R_b$	64 kbps	
Bit Stream	$G_3$	1/8 FLOPS/bps	$D_{bs} = R_b * G_3 * (1/r) = 0.32 \text{ MOPS}$
Source	$D_s$	1.6 MIPS/user	$N * G_4 = 4.02 \text{ MIPS per user}$
			$N * (W_c * (G_m + G_d) + R_b * G_3 / r + G_4)$ = 120.6 MOPS per cell site
	$D_o$	2 MOPS	
Aggregate	$D$		<b>122.6 MOPS per cell site</b>

\*) Typically performed in digital hardware in contemporary implementations

**Tabel-1.** Kebutuhan komputasi basestation tiap sel

Berdasarkan formula tersebut, kebutuhan komputasi tiap sel dapat ditentukan sebesar 12,5 GOPS pada bagian IF dan 122,6 MOPS untuk pengolahan sinyal baseband. Gambaran kebutuhan komputasi ini dapat dilihat pada Tabel-1.

**5. KESIMPULAN**

Berdasarkan uraian diatas dapat disimpulkan sebagai berikut:

- Tantangan utama dalam mengembangkan platform SDR yang fleksibel dan dapat dikonfigurasi ulang adalah komponen dijitasi (ADC/DAC) serta kompleksitas komputasi.
- Kebutuhan komputasi dapat dihitung dari kompleksitas komputasi perangkat lunak, serta pertimbangan skalabilitas perangkat lunak untuk mengantisipasi perkembangan standar komunikasi selanjutnya.
- Pendekatan yang digunakan dalam desain arsitektur lebih mengarah kepada perancangan berbasis obyek untuk menjamin tingkat fleksibilitas dan rekonfigurabilitas sebuah platform SDR.

**6. DAFTAR PUSTAKA**

[1] Reed, Jeffrey H, "Software Radio: A Modern Approach to Radio Engineering", New Jersey, Prentice Hall, 2002.  
 [2] Steinheider, J., "Software-defined Radio Comes of Age", *Mobile Radio Technology*, Feb 1<sup>st</sup>, [Online] <http://www.vanu.com/>

[resources/intro/software-defined\\_radio\\_comes\\_of\\_age.html](resources/intro/software-defined_radio_comes_of_age.html), 2003.  
 [3] Mitola, Joseph III, "Software Radio Architecture. Object-Oriented Approaches to Wireless Systems Engineering", Canada, John Eiley & Sons, Inc, 2000.  
 [4] Salkintzis, A., K., Nie, H., Mathiooulos, P. T., "ADC and DSP Challenges in the Development of Software Radio Base Stations", IEEE Personal Communications. August 1999.  
 [5] Christensen, Flemming, "A scalable Software-Defined Radio Development System". [On-line] [http://www.xilinx.com/publications/xcellonline/xcell\\_51/xc\\_pdf/xc\\_es-sundance51.pdf](http://www.xilinx.com/publications/xcellonline/xcell_51/xc_pdf/xc_es-sundance51.pdf), 2004.  
 [6] Barbeau, M., Bordeleau, F., Smith, J., "An Introduction to the UML of Software Radio", Proceedings on Conference in Telecommunication (ICT'2002), Beijing, China, une 2002.  
 [7] Zapata, J., Bordeleau, F., Smith, J., "UML Model Mapping in the Context of Software Radio", UML 2003, Oct 2003, San Fransisco, CA.  
 [8] Srikanteswara, S., Palat, R. C., Reed, J. H., Athanas, P., "An Overview of Configurable Computing Machines for Software Radio Handsets", IEEE Communications Magazine. July 2003.  
 [9] Angoletta, M. E., "From Analog to Digital Domain", [Online] [http://humanresources.web.cern.ch/humanresources/external/training/special/DISP2003/DISP-2003\\_L01A\\_20Feb03.pdf](http://humanresources.web.cern.ch/humanresources/external/training/special/DISP2003/DISP-2003_L01A_20Feb03.pdf).  
 [10] Dick, C.H., "Design and Implementation of High-Performance FPGA Signal Processing Datapath for Software Define Radios", [Online] [http://www.eetasia.com/ARTICLES/2001AUG/2001AUG09\\_ICD\\_AMD\\_RFD\\_TAC01.PDF](http://www.eetasia.com/ARTICLES/2001AUG/2001AUG09_ICD_AMD_RFD_TAC01.PDF), 2001.  
 [11] Huie, J., D'Antonio, P., Pelt, R., "Synthesizing FPGA Cores for Software-Defined Radio", [Online] <http://www.altera.com/cp/fpga-cores-for-sdr.pdf>. 2005.