

KAPITA SELEKTA

Middleware : SensorWare for Internet of Things



Oleh :

Chusniah (09011281320026)

JURUSAN SISTEM KOMPUTER

UNIVERSITAS SRIWIJAYA

TAHUN 2016

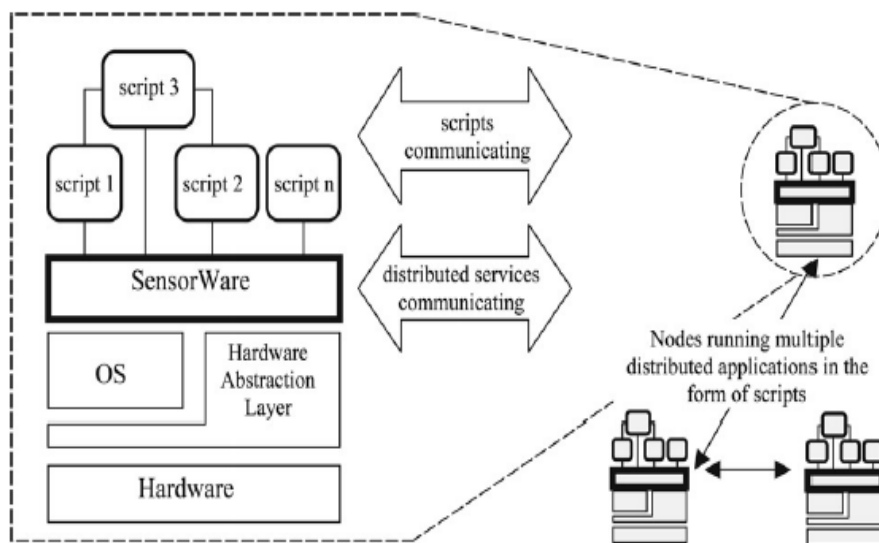
Middleware : SensorWare for Internet of Things

Konsep *Internet of Things* (IoT) menggambarkan kondisi masa depan khususnya di bidang digital dan benda fisik seperti smartphone, TV, mobil dapat dihubungkan dengan sarana informasi yang sesuai dengan teknologi komunikasi, untuk memungkinkan berbagi aplikasi dan jasa. Karakteristik IoT meliputi jaringan berskala ultra, perangkat dan jaringan *heterogeneous*, dan sejumlah peristiwa (*event*), serta berbagai tantangan yang ditimbulkan sehingga memicu perkembangan dibidang ini. Secara umum, middleware dapat meringankan proses pengembangan dengan mengintegrasikan komputasi heterogen dan perangkat komunikasi, dan interoperabilitas mendukung dalam aplikasi dan layanan yang beragam. Baru-baru ini, ada telah ada sejumlah arsitektural yang direkomendasikan untuk IOT *middleware*, salah satunya yaitu *SensorWare*.

Nirkabel ad hoc jaringan sensor sudah banyak dirancang secara statis dengan arsitektur untuk menjalankan tugas-tugas tertentu, sehingga kemampuan operasi dan interaksi tidak fleksibel. Untuk mengatasi masalah pemrograman bagian algoritmik ekspresif, kekompakkan kode saat transfer, efisiensi kode saat dieksekusi, dan kemudahan program, dipilihlah satu abstraksi untuk sensor simpul yaitu *SensorWare*.

SensorWare didefinisikan sebagai pendukung skrip ringan control mobile yang memungkinkan efisiensi penggunaan komputasi, perhitungan, komunikasi dan sumber daya penginderaan di node sensor untuk secara efisien dimanfaatkan dalam mode aplikasi-spesifik, melalui penggunaan layanan abstraksi. Layanan ini berbagi sumber daya dari sebuah node di antara banyak aplikasi dan Pengguna secara bersamaan salah satunya menggunakan WASN. Salah satu fitur dari *SensorWare* adalah memungkinkan instansiasi dinamis layanan baru (kode sebagai skrip kontrol) sebagai abstraksi dari simpul run time. Sebuah script bisa meniru secara mandiri atau bermigrasi data ke node lain, yang memengaruhi perilaku mereka secara langsung. Replikasi atau migrasi dari scrip tersebut disebut “populasi”. Pengguna “*inject*” *query*/program ke jaringan, dan *query* secara otomatis didistribusikan melalui algoritma hanya ke nodes yang terpengaruhi.

Untuk kebanyakan aplikais nodes ini hanya dapat diketahui pada saat *run-time*, sehingga penyebaran algoritma harus di eksekusi. Dengan fitur serbaguna dan kode *autonomously mobile*, programer dapat memilih pelaksanaan yang paling sesuai dengan kebutuhan aplikasi. Misalnya, programmer dapat merancang sebuah *selfsufficiant*, skrip *self-deployed*, yang hanya didasarkan pada layanan asli atau costum layanan *predeployed* serta skrip yang menggunakan layanan ini.

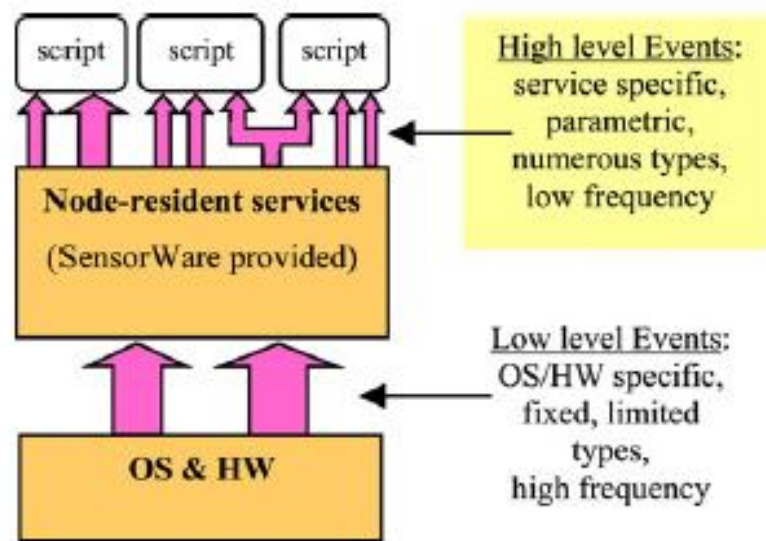


Gambar 1 Arsitektur Sensor Node Secara Umum

Pada gambar 1 diatas menunjukkan bahwa *SensorWare* terletak didalam arsitektur sensor node dan lapisan palih bawah atau terendah yaitu *hardware* (drive perangkat). Sedangkan untuk sistem operasi terletak pada lapisan atas dari *hardware*. OS menyediakan layanan serta fungsi standar yang dibutuhkan oleh lapisan atas. Misalnya *SensorWare* menggunakan fungsi-fungsi dan layanan untuk menyediakan lingkungan run-time untuk skrip kontrol. Skrip kontrol bergantung pada lapisan *SensorWare* sementara populasi bergantung pada jaringan. Skrip kontrol menggunakan layanan asli dimana *SensorWare* menyediakan layanan skrip untuk membangun apliaksi terdistribusi. Script akan berkomunikasi dengan skrip dari jenis yang sama (misalnya, memiliki kode yang sama) di node terpencil.

Dua bagian dari *SensorWare*, yaitu bahasa dan lingkungan yang mendukung kondisi *run-time*.

Sebuah bahasa skript butuh perintah-perintah untuk bertindak sebagai dasar blok atau algoritma skript. Perintah ini pada dasarnya sebagai antarmuka untuk layanan abstraksi yang ditawarkan oleh *SensorWare*. Contoh sederhana seperti layanan *timer*, penginderaan akuisi data, mobilitas skript, penemuan lokasi protokol. Selanjutnya, bahasa skript membutuhkan konstruksi agar blok terbentuk di skript kontrol. Beberapa contoh meliputi :



Gambar 2 Proses Script

konstruksi pengontrol aliran, seperti loop dan pernyataan kondisional, konstruksi untuk penanganan variabel dan konstruksi untuk mengevaluasi keadaan. Semua ini disebut dengan “*Glue Core*” dari bahasa. Sebagai *glue core* yang dapat menggunakan inti dari salah satu bahasa scripting yang tersedia. Bahasa scripting yang cocok untuk *SensorWare* adalah Tcl yang menawarkan modularitas besar dan portabilitas. Tcl ini digunakan sebagai *glue core* dalam bahasa *SensorWare*. Perintah-perintah dasar didefinisikan sebagai perintah Tcl baru menggunakan metode standar yang disediakan oleh Tcl. *SensorWare* bukan bahasa pemrograman tingkat node.

SensorWare sebagai penyedia layanan dengan mendaftarkan nama perangkat, mendefinisikan fungsi untuk empat dasar perintah perangkat, dan mendefinisikan beberapa tugas perangkat. *SensorWare* menggunakan skript untuk

menentukan dan mendaftarkan perangkat baru secara dinamis. Perangkat skrip harus mendefinisikan empat prosedur khusus untuk menerapkan empat perangkat dasar(*query*, *act*, *interest*, *dispose*) dan mendaftarkan nama perangkat menggunakan perangkat perintah khusus. Skript utama mengimplementasikan tugas perangkat. *SensorWare* dapat mengubah abstraksi dari lingkungan run-time yang ditawarkan untuk skrip berikutnya. *SensorWare* berada dibawah keluarga *sensor frameworks* aktif, jaringan lain yang sama seperti abstraksi *scripting*.

Listing 2

The script that floods an area of radius *r* around a target point

```
#code_id 64 radius_flood2
# to be used with gpsr test

parameter target_loc_x target_loc_y radius\
           populated_script user_x user_y

proc dist {x1 y1 x2 y2} {
    set diff_x [expr "$x1 - $x2"]
    set diff_y [expr "$y1 - $y2"]
    return [expr "($diff_x * $diff_x) + ($diff_y * $diff_y)"]
}

set my_loc [query location]
set my_loc_x [lindex $my_loc 0]
set my_loc_y [lindex $my_loc 1]
set dist_squared [dist $my_loc_x $my_loc_y\
                     $target_loc_x $target_loc_y]

# *****
# We check if we are inside the area that must be flooded.
# if yes, we replicate in ALL neighbors and spawn the
# populated_script
# *****

if {($radius <0) || ([expr $dist_squared-$radius*$radius] <= 0)}
{
    debug "flooding into neighborhood"
    replicate neighbor 0
    carry $user_x $user_y
    spawn self $populated_script
}

```

Gambar 3 contoh skript *SensorWare*

Bayangkan bahwa perangkat GPSR tidak termasuk dalam paket layanan asli yang ditawarkan oleh *SensorWare*. Misalnya, mungkin benar jika kita memutuskan untuk mengurangi jejak dari *SensorWare* dengan mengurangi beberapa perangkat diimplementasikan. Atau hanya membayangkan bahwa programmer, menemukan pelayanan GPSR tidak memadai (karena hanya rute ke titik), ingin untuk mengimplementasikan aplikasi dari awal.

Referensi :

Boulis, Athanassios. 2007. "SensorWare: Programming Sensor Networks Beyond Code Update and Querying". Elsevier, United State.