**Kelompok 4:**

**Guntur Wisnu Saputra\_11211042**

**Muhammad Insan Kamil\_11211058**

**Muhammad Ricky Zakaria\_11211062**

**Ramadhan Djibran Sanjaya\_11211070**

**Rangga Hermawan\_11211071**

**Rendy Pernanda\_11211074**

**Insertion Red Black Tree**

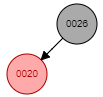
**(26 20 83 80 54 6 84 70)**

**Insert (26):**



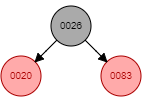
Karena Tree kosong, maka node(26) yang di-insert dijadikan sebagai root dengan warna hitam.

**Insert (20):**



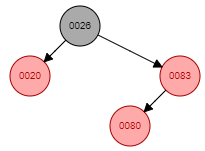
Node(20) di-insert disebelah kiri root karena key node(20) lebih kecil dari key root(26), kemudian warna node(20) diatur menjadi merah karena salah satu aturan Red Black Tree yaitu setiap node yang baru di-insert pasti merah.

**Insert (83):**

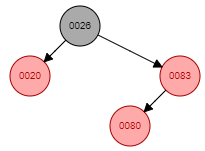
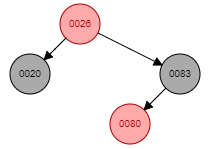


Node(83) di-insert disebelah kanan root karena key node(83) lebih besar dari key root(26), kemudian warna node(83) diatur menjadi merah karena salah satu aturan Red Black Tree yaitu setiap node yang baru di-insert pasti merah.

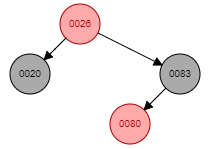
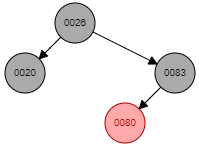
**Insert (80):**



1. Key node(80) dibandingkan dengan key root(26), yang dimana key node(80) lebih besar dari key root(26).
2. Karena lebih besar, maka key node(80) dibandingkan lagi dengan key dari node yang ada disebelah kanan root(node(83)), yang dimana key node(80) lebih kecil dari key node(83), sehingga node(80) di-insert disebelah kiri node(83). kemudian warna node(80) diatur menjadi merah karena salah satu aturan Red Black Tree yaitu setiap node yang baru di-insert pasti merah.

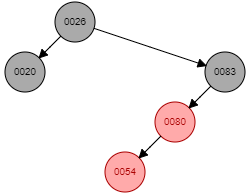
 

Karena parent dan uncle dari node(80) yang baru diinsert warnanya merah, maka parent dan uncle dari node(80) diganti warnanya menjadi hitam dan grandparent dari node(80) diganti warnanya jadi merah.

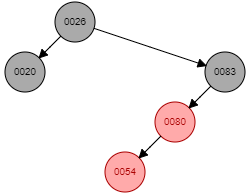
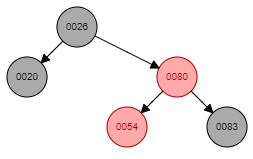
 

Berdasarkan aturan Red Black Tree, root harus selalu berwarna hitam, sehingga root(26) diubah warnanya menjadi hitam.

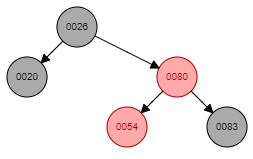
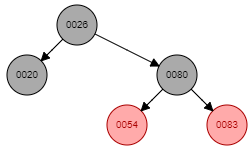
**Insert (54):**



1. Key node(54) dibandingkan dengan key root(26), yang dimana key node(54) lebih besar dari key root(26).
2. Karena lebih besar, maka key node(54) dibandingkan lagi dengan key dari node yang ada disebelah kanan root(node(83)), yang dimana key node(54) lebih kecil dari key node(83).
3. Karena lebih kecil, maka key node(54) dibandingkan lagi dengan key node(80), yang dimana key node(54) lebih kecil dari key node(80),sehingga node(54) di-insert disebelah kiri node(80). kemudian warna node(54) diatur menjadi merah karena salah satu aturan Red Black Tree yaitu setiap node yang baru di-insert pasti merah

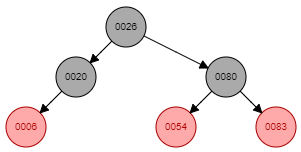
 

Batas toleransi balancing red black tree yaitu kedalaman jalur terpanjang maksimal dua kali kedalaman jalur terpendek. Karena kedalaman jalur terpanjang = 3(83 → 80 → 54) dan kedalaman jalur terpendek = 1 (83), kondisi tersebut telah melewati batas toleransi balancing red black tree atau bisa dikatakan red black tree dalam keadaan imbalance sehingga node(83) dirotasi kekanan agar Red Black Tree menjadi balance lagi.

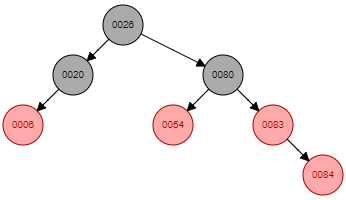
Berdasarkan aturan Red Black Tree, tidak boleh ada dua node merah yang berdekatan, sehingga node(80) diubah warnanya menjadi hitam dan node(83) diubah warnanya menjadi merah.

**Insert (6):**

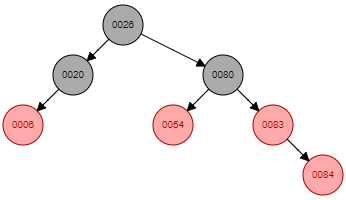
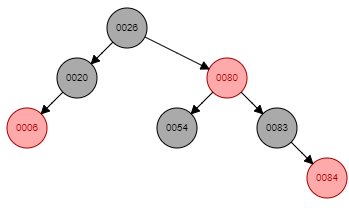


1. Key node(6) dibandingkan dengan key root(26), yang dimana key node(6) lebih kecil dari key root(26).
2. Karena lebih kecil, maka key node(6) dibandingkan lagi dengan key dari node yang ada disebelah kiri root(node(20)), yang dimana key node(6) lebih kecil dari key node(20), sehingga node(6) di-insert disebelah kiri node(20). kemudian warna node(6) diatur menjadi merah karena salah satu aturan Red Black Tree yaitu setiap node yang baru di-insert pasti merah.

**Insert (84):**

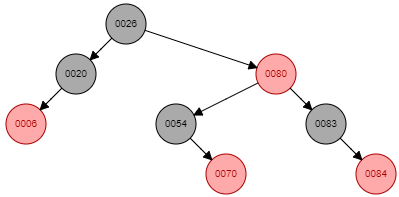
****

1. Key node(84) dibandingkan dengan key root(26), yang dimana key node(84) lebih besar dari key root(26).
2. Karena lebih besar, maka key node(84) dibandingkan lagi dengan key dari node yang ada disebelah kanan root(node(80)), yang dimana key node(84) lebih besar dari key node(80).
3. Karena lebib besar, maka key node(84) dibandingkan lagi dengan key node(83), yang dimana key node(84) lebih besar dari key node(83),sehingga node(84) di-insert disebelah kanan node(83). kemudian warna node(84) diatur menjadi merah karena salah satu aturan Red Black Tree yaitu setiap node yang baru di-insert pasti merah.

**** 

Karena parent dan uncle dari node(84) yang baru diinsert warnanya merah, maka parent dan uncle dari node(84) diganti warnanya menjadi hitam dan grandparent dari node(84) diganti warnanya jadi merah.

**Insert (70):**

****

1. Key node(70) dibandingkan dengan key root(26), yang dimana key node(70) lebih besar dari key root(26).
2. Karena lebih besar, maka key node(70) dibandingkan lagi dengan key dari node yang ada disebelah kanan root(node(80)), yang dimana key node(70) lebih kecil dari key node(80).
3. Karena lebih kecil,maka key node(70) dibandingkan lagi dengan key node(54), yang dimana key node(70) lebih besar dari key node(54),sehingga node(70) di-insert disebelah kanan node(54). kemudian warna node(70) diatur menjadi merah karena salah satu aturan Red Black Tree yaitu setiap node yang baru di-insert pasti merah.